# USING SERIOUS GAMES FOR IMPROVING MEDICAL EDUCATION: AN APPLICATION TO CYTOPATHOLOGY

DAN CRISTIAN ROTARU

MASTERS IN COMPUTER SCIENCE, FACULTY OF INFORMATICS,
COMPLUTENSE UNIVERSITY OF MADRID


Final Masters Project in Computer Science

20 / 06 / 2016

**Master Thesis Supervisors**

Baltasar Fernández Manjón, Complutense University of Madrid
Rosemary H. Tambouret, Massachusetts General Hospital, Harvard University
Avni Khatri, Massachusetts General Hospital

Final Grade: 9

# Autorización de Difusión

DAN CRISTIAN ROTARU

01 / 06 / 2016

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Using Serious Games for Medical Education: An Application to Cytopathology", realizado durante el curso académico 2015-2016 bajo la dirección de Baltasar Fernández-Manjón y con la colaboración externa de dirección de Rosemary H. Tambouret (MGH, Harvard University) y Avni Khatri (MGH), en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

# Resumen en castellano

El objetivo principal del proyecto es desarrollar una plataforma compuesta por aplicaciones educativas gamificadas para el entrenamiento de personal médico en países de recursos limitados en citopatología mediante dispositivos Android de bajo presupuesto. Antes de desplegar la plataforma in países con recursos limitados, va a ser probada en un curso de Introducción a Citopatología de la Escuela Médica de Harvard. El proyecto final debe funcionar tanto en *PCs* como en dispositivos Android de bajo coste (p.e. 50 dólares americanos, Amazon Kindle Fire 7 pulgadas) y no puede depender de una conexión a internet continua. Se han analizado algunas aplicaciones con propósito de juego y simulaciones gamificadas para tener una base de conocimiento común entre expertos médicos y desarrolladores. También se han estudiado juegos y aplicaciones cuyo objetivo es hacer uso de imágenes médicas para entrenamiento de personal médico o están enfocadas al diagnóstico mediante colaboración por parte de personal no-médico. Esto nos ha permitido identificar las mejores mecánicas de juego para nuestro caso de uso. A continuación, se han comparado diferentes herramientas de edición y motores de juegos desde el punto de vista del rendimiento ofrecido, las plataformas soportadas, su documentación y licencia. Todo ello nos ha permitido elegir la tecnología de desarrollo (libGDX). Finalmente, diseñamos e implementamos un sistema integrado de aplicaciones (editor de contenido y generador de juegos). El sistema está enfocado a reducir la dependencia entre el personal experto y los desarrolladores para crear y mantener contenido educativo. Se trata de una arquitectura formada por un servicio RESTful, y un editor asociado, orientado a la gestión de contenido educativo orientado para citopatología y dos clientes para diferentes plataformas (PC y Android) que consumen dicho servicio. Finalmente, se presentan las conclusiones y el trabajo futuro del proyecto.

# Palabras clave

Video juegos educativos
Juegos serios
Entrenamiento médico
Diseño de juegos
Gamificación
Citopatología
Exámenes médicos

# Resumen en inglés

The main goal of the project is to develop a platform composed of gamified educational applications to train cytologists in resource-limited areas using low-cost Android tablets. Before deploying the platform in those areas, it's going to be tested at an Introduction to Cytopathology course at Harvard Medical School. The project has some challenging technical requirement as to be deployable both in PCs and in low-cost Android tablets (e.g., USD 50 Kindle Fire 7 inches tablet) and not depend on a stable internet connection. We started reviewing some of the existing games or gamified e-learning modules to create a shared understanding between medical experts and developers about possible game mechanics and to identify which of those approaches can be suitable for our case. We have also studied a number of games and applications with different approaches and goals that use the analysis of medical image, either to train medical personnel or are intended for crowdsourced collaboration in diagnosis done by non-medical experts. This initial study allowed us to identify key gameplay mechanics that can be applied to our use case. Afterwards, we compared different authoring tools and game development frameworks from the point of view of the target platform, performance, documentation, license, learning assessment support and game mechanics. With this study we have been able to select the development technology of our project (libGDX). Finally, we design and implement an integrated system easy to use by the medical personnel (content editor and game generator). The system allows domain experts to manage the content without depending on programmers. The solution's architecture is composed of a RESTful service, and editor, aimed at managing cytopathology oriented content and two clients for different platforms (PC and Android) that use this service. Finally the conclusions and future work are presented.

# Keywords

Educational video games
Serious games
Medical training
Game design
Gamification
Cytopathology
Medical screening

# Table of Contents

# Figures

# Tables

# Agradecimientos / Acknoledgements

I would like to thank my advisors Baltasar Fernández-Manjón, Rosemary H. Tambouret and Avni Khatri, for their support and effort put into this work, by providing guidance and continuous constructive feedback.

I would also like to thank the whole e-UCM Research Group for their continuous support and motivation. Their experience in the field of e-Learning and Cytopathology was essential for completing this work.

# 1 Introduction

## 1.1 Digital Games

Digital games have increasingly gained relevance in society to the point of becoming one of the most popular entertainment forms. The total revenue of the digital games sector, in the US market alone, has grown from USD 2.6 billion in 1996 to over USD 15.4 billion in 2013 (ESA, 2015). The digital games sector has suffered a moderate slowdown in the last years (ESA, 2015) due to the economic crisis, but is still able to generate massive revenues. For instance, in September 2013 Rockstar released *Grand Theft Auto V*, which generated more than USD 1 billion in retail sales during its first three days on sale, being the fastest entertainment property to reach this milestone, including digital games and feature films ("Forbes - 'Grand Theft Auto V' Crosses $1B In Sales, Biggest Entertainment Launch In History," n.d., "IGN - GTA 5 SALES HIT $1 BILLION IN THREE DAYS," n.d.). The digital games sector has been compared with more established arts such as painting, music, or movies. The distinction between digital games and fine arts is slowly disappearing, as evidenced by growing numbers of movies and books based on games such as *Uncharted*, *Angry Birds* or *Assassin's Creed*.

The games culture has increasingly grown in the past years to the point of including a big part of the world's population. This is, in part, due to the proliferation of mobile devices (i.e. smartphones and tablets) that greatly increase the number of gamers (casual gamers) and the possibility to play games in different environments, previously impossible using desktop and laptop computers (NMC, 2014). Now, the average age of today gamer's is 35 and an 74% of gamers are over 18 years old (ESA & ESA, 2015). There are numerous examples of games and game-like approaches successfully used in education. Those games are called "serious games" as their main goal is not only entertainment.

## 1.2 Serious Games

Digital games, when used to further goals such as health or education rather than simply entertain, are usually called serious games (De Freitas, 2006) (SGs). Continuous education, at all ages, is becoming one of the great priorities for our society, and increasing student motivation, always an important problem in education, is now more relevant than ever. Since large and diverse portions of the population already play games regularly as part of their routine, serious games are a powerful tool to increase student engagement and motivation, presenting players with challenging and immersive environments, and providing the ability to fail without consequences. This safe, failure-tolerant environment allows players to experience situations from different

points of view, or experiment freely and exercise their curiosity. By providing constant feedback to the player, games can be used for procedural learning (Jarvis & Freitas, 2009). And by motivating and challenging the players to explore and overcome specific problems, serious games are gaining interest in teaching and training students. Serious games have been applied to health (Akl, Kairouz, & Sackett, 2013; Arnab, Dunwell, & Debattista, 2013; Brox, Fernandez-Luque, & Tøllefsen, 2011; Jr et al., 2007), marketing (Pempek & Calvert, 2009), research (Cooper et al., 2010), and multiple other disciplines, where they have proven to increase students motivation in the learning process (Dickey, 2005; Kirriemuir & McFarlane, 2004; Michael & Chen, 2005) stimulating their focus and concentration (De Freitas, 2006; Dede, 2009; Gee, 2003). Studies have shown SGs increasing academic performance (Connolly, Boyle, Macarthur, Hainey, & Boyle, 2012; Hwang & Wu, 2012; Perrotta, Featherstone, Aston, & Houghton, 2013) and providing an optimized learning process (Chen, 2007; Mayo, 2009), as part of a growing set of experiments that prove the effectiveness of correctly designed and implemented serious games (Annetta, Minogue, Holmes, & Cheng, 2009; Barzilai & Blau, 2014; Cheng, Su, Huang, & Chen, 2013; Papastergiou, 2009; Tüzün, Yilmaz-Soylu, Karakuş, Inal, & Kizilkaya, 2009).

### *1.2.1 Serious Games for Health*

Serious Games for health are a sub-group of the serious games and have been increasingly used in the past years. There are examples in very different areas, such as games targeted at HIV prevention education, cancer diagnosis, dental pain, etc. (Lyons, 2014). For instance, in Stanford University School of Medicine SICKO ("SICKO Serious Game," n.d.), a web-based game in which players must take care of three virtual patients while at the same time they make critical decisions in the operating room. Serious games are not only intended for medical personnel as playing serious games can also help the patients with different types of problems. For instance, veterans are using games to treat their posttraumatic stress disorder and burn patients are using games to relieve their pain (Wire, 2014).

### *1.2.2 Limitations of Serious Games for Education Training*

To promote the use of games in formal education settings, educators need additional support. According to Figure 1. Factors that influence adoption of serious games, from http://gamesandlearning.org, beyond positive reviews by other teachers, games that include assessment, tracking and other classroom management features are strongly preferred. Evidence of educational impact is also important. Both factors can be addressed by using established game learning analytics techniques. However, these techniques need to be applied early and often, ideally starting during the design stages of the SG and continuing through the final implementation. Additionally, once a game has been considered an educational content, it should remain

available and ready to use regardless of technological changes or platform evolution. The decision to benefit from serious games or forego their usage completely hinges on the total cost of ownership (TCO) of developing, fine-tuning, deploying and maintaining such a game, which are distinct from those of commercial entertainment games.

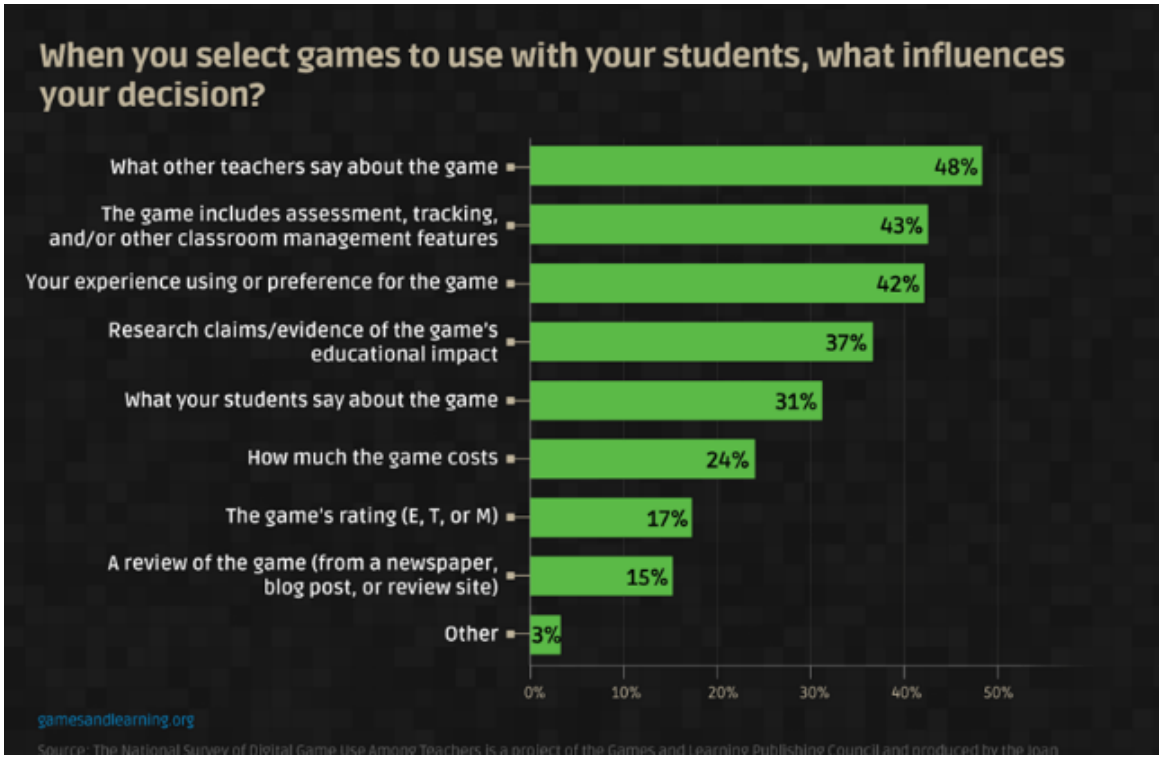



Figure 1. Factors that influence adoption of serious games, from http://gamesandlearning.org

## 1.3 An Introduction to Cytopathology

Cytopathology is a branch of pathology that studies and diagnoses diseases on the cellular level. At Harvard Medical School, professors that are teaching the Introduction to Cytopathology course have identified the possibility to take advantage of newer teaching methodologies in order to engage their students with more feature-rich content. The main goal of the project is to develop a platform for gamified educational applications that can be deployed to train cytologist in resource-limited areas of the world. Before being deployed, the applications are going to be tested in an Introduction to Cytopathology course at Harvard Medical School. The platform is to be designed so that the applications which result from it can eventually be used to train cytologists in low resource countries using low-cost Android tablets.

"Uterine Cervical Cytology" has been selected as the pilot module and will be built based on pre-existing slide content. Educational content will include the morphology of the spectrum of normal and abnormal cells and the guidelines for cervical cancer/precancer screening by cytology and HPV (human papillomavirus) testing and follow-up of abnormal results.

The educational approach is focused on training the medical personnel in medical imaging. The ability to morphologically identify normal and abnormal cells and to locate rare abnormal cells among many normal-appearing cells are the most difficult for students (or residents) to learn. These learning objectives had to be presented in a gamified way, offering exciting, innovative, and effective methods for increasing the knowledge of the learner. Knowing which story and game characteristics (e.g. game mechanics) appeal to specific types of people could help tailor game design and behaviour change procedures to maximize effectiveness.

Implementing the gaming platform will be done taking in consideration different key project requirements such as the target resolution for Android tablets and PCs, code maintainability, applications design, etc. Teaching aims include cell identification, differential diagnosis - will be created using the platform and tested with clinicians for further refinement - and formative evaluation.

## 1.4 Goal

In our project we want to create an educational gaming platform that can be used as content for an Introduction to Cytopathology course. But, the use of serious games to train medical personnel in cytopathology is still a relatively uncharted field. Our project has a limited budget and some challenging technical requirements as the game should be also aimed to train cytologists in resource-limited areas of the world. That means for instance that the tablet version of the game should run in low-cost Android tablets (e.g., USD 50 Kindle Fire 7 tablet) and be fully functional without requiring continuous internet connection. We decided to implement an integrated system composed of an education application (i.e. game) and an editor for

the application. The system must be easy to use by the medical personnel and must allow the content management without the need of a developer.

For this work we have defined the following goals:

- Goal 1 - Propose an architecture that will allow domain experts in the cytopathology field to define and manage gamified applications that are tested with students before being deployed in resource-limited areas to train cytologists.
- Goal 2 - Develop an educational application that can be used to train cytologist in resource-limited areas or the world.
- Goal 3 - The content of the application must be managed by the medical personnel without the dependency of the author of this work or any other developer.
- Goal 4 - Test the proposed workflow with students of an Introduction to Cytopathology course at Harvard Medical School.

Furthermore, implementing the gaming platform had to be done taking in consideration different key project technical requirements such as the target resolution for Android tablets and PCs, code maintainability, architecture design, etc. This requires to make an analysis of the process of serious games design and development for medical training and to analyze existing games aimed at medical training in the field of cytopathology and make use of medical imaging. We begin by doing and study the state of the art of the game development tools in order to identify the most suitable development technology for our project.

## 1.5 Structure of this Document

The following parts of this document are structured as follows:

- Chapter 3 – Describe the literature review about serious games for health to identify what game characteristics appeal to different kind of players. Furthermore, it also include a more specific analysis of serious games for medical training that was needed to obtain specific game mechanics that could be useful for our case study.
- Chapter 4 – In this chapter, we analyze the factors that impact the total cost of ownership of serious games used in classroom settings related to the development tools and frameworks used in the process. This study helped us identify the technology used to develop the game engine. Some difficulties between cytopathology domain experts and game developers are discussed and an approach to this problem is presented as a possible solution.

- Chapter 5 – The design on the integrated system is presented. A system that is usable by the medical personnel and facilitates the content management. Is composed of (1) a cytopathology application (client), (2) a RESTful backend service and (3) a cytopathology web editor.
- Chapter 6 – The integrated system is described from the point of view of the final user. How a player uses the cytopathology application (1), how a developer can use the RESTful backend service (2) and how the medical personnel can manage the content of the system using the web editor (3).
- Chapter 7 – Conclusions from this thesis and potential future work are presented.

## 1.6 Conclusions

Professors at the Harvard Medical School and Massachusetts General Hospital medical personnel want an educational application to train cytologists in resource-limited areas of the world. Before deploying the application to those areas, they want to test it with real students at an Introduction to Cytopathology course at Harvard Medical School. The course's content is composed of analysis and diagnosis of abnormal cells based on medical images.

The initial constraints identified are very challenging. We must develop an application that is easy to use by the medical personnel. The medical personnel shall be able to manage the content of the application without the dependency of a developer, so an associated editor is required. Furthermore, the educational application must run on low-cost Android devices and cannot depend on a stable internet connection because it's going to be deployed on resource-limited areas of the world using these devices and the internet connection it's not necessary stable in these areas. At the same time, it must also run on desktop computers because it's going to be tested at an Introduction to Cytopathology course at Harvard medical School. In order to remove the Operative System dependency we have chosen as target platform for the desktop computers the HTML5 web browsers. The initial budget of the project is very limited, enough to buy a low cost Android tablet, the USD 50 Amazon Kindle Fire 7 inches, the device used for testing purposes of the educational application.

Once we had identified our project goal and the initial project constraints, the next step is to study similar applications in order to identify the best gameplay mechanics suitable for out project. In the following chapter we explain in detail this study and how we identified the gameplay mechanics of the final developed system.

# 2 Introducción

## 2.1 Juegos Digitales

Los juegos digitales han adquirido cada vez más importancia en la sociedad hasta el punto de convertirse en una de las formas más populares de entretenimiento. Los ingresos totales del sector de los juegos digitales, en el mercado estadounidense, ha crecido de 2.6 mil millones de dólares americanos en 1996 a más de 15.4 mil millones de dólares americanos en 2013 (ESA, 2015). El sector de los juegos digitales ha sufrido una desaceleración moderada en los últimos años (ESA, 2015) debido a la crisis económica, pero todavía es capaz de generar ingresos masivos. Por ejemplo, en septiembre de 2013 Rockstar lanzó Grand Theft Auto V, que generó más de mil millones de dólares americanos en ventas al por menor durante sus primeros tres días a la venta, siendo la propiedad de entretenimiento más rápida para llegar a este hito, incluyendo juegos digitales y películas ("Forbes - 'Grand Theft Auto V' Crosses $1B In Sales, Biggest Entertainment Launch In History," n.d., "IGN - GTA 5 SALES HIT $1 BILLION IN THREE DAYS," n.d.). El sector de los juegos digitales ha sido comparado con las artes, como la pintura, la música o las películas. La distinción entre los juegos digitales y las bellas artes está desapareciendo lentamente, como lo demuestra el número de películas y libros basados en juegos como Uncharted, Angry Birds o Assasin's Creed.

La cultura de los juegos ha crecido cada vez más en los últimos años hasta el punto de incluir una gran parte de la población mundial. Esto es, en parte, debido a la proliferación de dispositivos móviles (es decir, los teléfonos inteligentes y tabletas) que aumentan en gran medida el número de jugadores (los jugadores ocasionales) y la posibilidad de jugar juegos en diferentes entornos, más allá de los ordenadores de sobremesa y portátiles (NMC, 2014). Ahora, la edad media de jugador de hoy es 35 y un 74% de los jugadores son mayores de 18 años de edad (ESA & ESA, 2015). Hay numerosos ejemplos de juegos y enfoques gamificados utilizados con éxito en la educación. Esos juegos son llamados "juegos serios" ya que su objetivo principal no sólo es entretenimiento.

## 2.2 Juegos Serios

Los juegos digitales, cuando se utilizan para otros objetivos, como la salud o la educación en lugar de simplemente entretener, son llamados juegos serios (De Freitas, 2006). La educación continua, en todas las edades, se está convirtiendo en una de las grandes prioridades de nuestra sociedad, y el aumento de la motivación del estudiante, siempre es un problema importante en la educación, es ahora más importante que nunca. Gran cantidad de personas juegan regularmente como parte de su rutina. Los juegos serios son una

herramienta poderosa para aumentar la participación de los estudiantes y la motivación. Por ejemplo enfrentando a los jugadores con entornos desafiantes y adictivos, pudiendo fallar continuamente para aprender. Este ambiente seguro, tolerante a fallos permite a los jugadores experimentar situaciones desde diferentes puntos de vista, o experimentar libremente y ejercer su curiosidad. Al dar *feedback* constante para el jugador, los juegos pueden ser utilizados para el aprendizaje procedimental (Jarvis & Freitas, 2009). Motivando y desafiando a los jugadores a explorar y superar los problemas específicos, los juegos serios están ganando interés en la enseñanza y formación. Los juegos serios se han aplicado a la salud (Akl et al., 2013; Arnab et al., 2013; Brox et al., 2011; Jr et al., 2007), marketing (Pempek & Calvert, 2009), la investigación (Cooper et al., 2010), y varias otras disciplinas, en el que han demostrado aumentar la motivación de los estudiantes en el proceso de aprendizaje (Dickey, 2005; Kirriemuir & McFarlane, 2004; Michael & Chen, 2005) estimulando su concentración (De Freitas, 2006; Dede, 2009; Gee, 2003). Los estudios han demostrado que los juegos serios aumentan de rendimiento académico (Connolly, Boyle, Macarthur, et al., 2012; Hwang & Wu, 2012; Perrotta et al., 2013) y que proporcionan un proceso de aprendizaje optimizado (Chen, 2007; Mayo, 2009), como parte de un conjunto cada vez mayor de experimentos que demuestran la eficacia de los juegos serios (Annetta et al., 2009; Barzilai & Blau, 2014; Cheng et al., 2013; Papastergiou, 2009; Tüzün et al., 2009).

### *2.2.1 Juegos Serios para la Salud*

Los juegos serios para la salud son un subconjunto de los juegos jerios y se han utilizado cada vez más en los últimos años. Existen ejemplos en muy diferentes áreas como por ejemplo juegos dirigidos a la prevención del VIH, el diagnóstico de cáncer, dolor dental, etc. (Lyons, 2014). Por ejemplo, en la Escuela Universitaria de Medicina de Stanford, SICKO es ("SICKO Serious Game," n.d.), un juego basado en la web en el que los jugadores deben tener cuidado de tres pacientes virtuales , mientras que toman decisiones críticas en la sala de operaciones . Los juegos serios no sólo están destinados para el personal médico, también puede ayudar a los pacientes con diferentes tipos de problemas. Por ejemplo, los veteranos están utilizando los juegos para tratar el trastorno de estrés postraumático y los pacientes que han sufrido quemaduras usan los juegos serios para aliviar el dolor (Wire, 2014).

### *2.2.2 Limitaciones de los Juegos Serios para la Educación*

Para promover el uso de juegos en la educación formal, los educadores necesitan apoyo adicional. De acuerdo con la Figure 1, los juegos que incluyan la evaluación, seguimiento y otras funciones de gestión del aula son preferidos por los profesores. La evidencia de impacto educativo también es importante. Ambos factores pueden abordarse mediante el uso de una infraestructura de analíticas para el aprendizaje. Sin embargo, estas

técnicas necesitan ser aplicadas temprano y durante las etapas de diseño del juego serio. Además, una vez que un juego se ha considerado un contenido educativo, debe estar disponible y listo para usar, independientemente de los cambios tecnológicos o la evolución de la plataforma. La decisión de beneficiarse de los juegos serios o renunciar a su uso influye por completo sobre el coste total de propiedad (CTP) de desarrollo, puesta a punto, el despliegue y el mantenimiento de un juego de este tipo, que difiere con el CTP de juegos de entretenimiento comerciales.

## 2.3 Introducción a la Citopatología

La citopatología es una rama de la patología que estudia y diagnostica las enfermedades a nivel celular. En la escuela médica de Harvard, profesores que enseñan el curso de Introducción a Citopatología han identificado la posibilidad de aprovechar las nuevas metodologías de enseñanza con el fin de motivar a sus estudiantes. El objetivo principal de nuestro proyecto es desarrollar una plataforma para aplicaciones educativas gamificadas que son parte de un curso de Introducción a Citopatología. La plataforma debe ser diseñada de manera que las aplicaciones que se derivan de ella, finalmente, se pueden utilizar para entrenar a citotecnólogos en países poco desarrollados usando tabletas Android de bajo coste.

"Citología uterina cervical" es el módulo usado para construir contenido basado en diapositivas preexistente. El contenido educativo incluirá la morfología de las células normales y anormales, las directrices para la detección del cáncer/precáncer de cuello uterino por citotecnólogos, la prueba VPH (prueba de Papanicolau) y el seguimiento de los resultados anormales.

El enfoque educativo se centra en la formación del personal médico en el tratamiento de imágenes médicas. La capacidad de identificar morfológicamente células normales, anormales y localizar las células anormales raras entre muchas células de apariencia normal son los más difíciles de aprender para los estudiantes (o residentes). Estos objetivos de aprendizaje deben ser presentados de manera gamificada, ofreciendo métodos interesantes, innovadores y eficaces para incrementar los conocimientos del alumno. Conociendo las características de juego (por ejemplo, la mecánica de juego) se puede diseñar el juego para maximizar el aprendizaje de los jugadores.

La implementación de la plataforma de juegos se hará tomando en consideración los diferentes requisitos clave del proyecto, tales como la resolución de pantalla para las tabletas Android y PC, el mantenimiento del código, etc.

## 2.4 Objetivo

En nuestro proyecto queremos crear una plataforma de juegos educativos que se pueda utilizar en un curso de Introducción a Citopatología. Sin embargo, el uso de juegos serios para entrenar a personal médico en citopatología es todavía un campo relativamente inexplorado. Nuestro proyecto tiene un presupuesto limitado y algunos requisitos técnicos desafiantes. Por ejemplo, entrenar a citotecnólogos en países poco desarrollados del mundo. Esto significa por ejemplo que la versión del juego para dispositivos móviles debería funcionar en tabletas Android de bajo coste (por ejemplo, 50 dólares americanos, Kindle Fire 7 pulgadas) y será completamente funcional sin necesidad de conexión continua a Internet. Decidimos implementar un sistema integrado compuesto por una aplicación educativa y un editor para la aplicación. El sistema debe ser fácil de manejar por el personal médico, le cual debe poder gestionar el contenido de la aplicación sin la necesidad de un desarrollador.

Para este trabajo se han definido los siguientes objetivos:

- Objetivo 1 - Proponer una arquitectura que permita a los expertos de dominio en el campo de la citopatología definir y gestionar aplicaciones gamificadas que se prueban con los estudiantes.
- Objetivo 2 - Desarrollar una aplicación gamificada que pueda ser usada para entrenar citólogos en áreas de recursos limitados del planeta.
- Objetivo 3 - El contenido de la aplicación educativa debe ser fácil de manejar por el personal médico sin depender el autor de este trabajo o cualquier otro desarrollador.
- Objetivo 4 - Realizar pruebas de proyecto propuesto con los estudiantes en un curso de Introducción a Citopatología en la Escuela de Medicina de Harvard.

Por otra parte, la implementación de la plataforma de juegos tenía que hacerse tomando en consideración los diferentes requisitos técnicos clave del proyecto como la resolución de pantalla para las tabletas Android y PC, mantenimiento del código de diseño, etc. Esto requiere hacer un análisis del diseño y desarrollo de juegos serios para la formación médica a partir de un análisis de los juegos existentes dirigidas a la formación médica en el campo de la citopatología basados en imágenes médicas. Para ello se ha estudiado el estado del arte de las herramientas de desarrollo de juegos con el fin de identificar tecnología de desarrollo más adecuada para nuestro proyecto.

## 2.5 Estructura de éste Documento

Las siguientes partes de este documento se estructuran de la siguiente manera:

- Capítulo 3 – Se presenta el estudio y análisis de los juegos serios para la salud para identificar qué características de juego apelan a diferentes tipos de jugadores. Por otra parte, se identifica que era necesario un análisis más específico de los juegos serios para la formación médica para obtener las mecánicas de juego específicos que podrían ser útiles para nuestro caso.
- Capítulo 4 - Análisis de las herramientas de desarrollo de juegos. Este estudio nos ayudó a identificar la tecnología utilizada para desarrollar el motor del juego (libGDX).
- Capítulo 5 – Se presenta el diseño de un sistema integrado de aplicaciones educativas cuyo objetivo es ser fácil de utilizar por el personal médico para no depender del desarrollador a la para gestionar el contenido del sistema. El sistema está dividido en tres partes (1) una aplicación para citopatología (cliente), (2) un servicio RESTful y (3) un editor web para la creación de contenido. Se explican las tecnologías utilizadas y las metodologías de desarrollo empleadas.
- Capítulo 6 – Se describe el sistema desde el punto de vista del usuario final. Cómo un jugador hace uso de la aplicación de citopatología (1), como un desarrollador puede hacer uso del servicio RESTful (2) y como el personal médico puede gestionar el contenido del sistema mediante el editor web (3).
- Capítulo 7 – Se presentan las conclusiones de esta tesis y el potencial trabajo futuro.

## 2.6 Conclusiones de capítulo

Profesores de la Escuela Médica de Harvard y personal médico del Hospital General de Massachusetts quieren una aplicación educativa para entrenar a citólogos en aras del planeta con recursos limitados. Antes de desplegar la aplicación en dichos áreas, la aplicación educativa va a ser probada con estudiantes reales en un curso de Introducción a Citopatología que se va a impartir en la Escuela Médica de Harvard. El contenido del curso está basado en el análisis y el diagnóstico de células problemáticas en imágenes médicas.

Partimos con unas restricciones iniciales bastante desafiantes. Debemos desarrollar una aplicación que sea fácil de utilizar por el personal médico. El personal médico debe de ser capaz de gestionar el contenido de la aplicación sin depender del autor de este trabajo ni de ningún otro desarrollador, por lo que se requiere un editor asociado. Además la aplicación educativa debe de funcionar en dispositivos Android de bajo coste y no puede depender de una conexión estable a internet para funcionar, ya que va a ser desplegada en áreas de recursos limitados mediante dicho dispositivos, donde la conexión a internet no suele ser estable. Al mismo tiempo debe de funcionar también en ordenadores ya que va a ser probada en un curso de Introducción a Citopatología en la Escuela Médica de Harvard con estudiantes reales. Para eliminar la dependencia de un

Sistema Operativo hemos decidido elegir como plataforma objetivo los navegadores web HTML5. El presupuesto inicial para este proyecto es muy limitado, el suficiente como para comprar una tableta Android de bajo presupuesto, el Amazon Kindle Fire de 7 pulgadas que cuesta 50 dólares americanos, el dispositivo utilizado para probar la aplicación educativa.

Una vez que hemos identificado los objetivos de nuestro proyecto y las restricciones iniciales, el siguiente paso lógico es estudiar aplicaciones educativas con objetivos similares para identificar las menores mecánicas de juego que puedan aplicarse a nuestro proyecto. En el siguiente capítulo explicamos con más detalle este estudio y cómo hemos llegado a identificar las mecánicas de juego del sistema final que hemos desarrollado.

# 3 Related Work and Applications

## 3.1 Systematic Review of Serious Games in the Medical Domain

Trying to identify best practices and design methodologies for serious games in *ehealth* (term for healthcare supported by electronic processes and communication) and more specifically in games for training medical personnel we queried several databases and performed a review of the results (in a similar way as it was done in (Connolly, Boyle, MacArthur, Hainey, & Boyle, 2012) about the empirical evidence on serious games). The query process was done on January, 1st of 2016.

We have searched different databases relevant to the topics of this study: ACM (Association for Computing Machinery), BioMed Central, PubMed, ERIC (Education Resources Information Center), and IngentaConnect.

We have used different search terms topics in conjunction. We have included terms for games such as "computer games", "video games", "serious games" and "simulation games" to help with the definition of digital games.

*("video games" OR "serious games" OR "simulation games")*

Terms to help narrow the medicine and health fields (especially the field of cytopathology and the study of infectious cells) have been used in conjunction with the digital games terms.

*AND ("cytopathology" OR "health" OR "infectious cell" OR "cancer")*

Finally, there have been chosen a set of terms to focus on the particular subject of game design and development of methodologies.

*AND ("design" OR "methodology" OR "survey" OR "best practices")*

Table 1. Search results for the electronic databases consulted

| Database | Search results | Number of papers meeting selection criteria |
|---|---|---|

| | | |
|---|---|---|
| ACM | 76 | 6 |
| BioMed central | 448 | 5 |
| ERIC | 35 | 1 |
| IngentaConnect | 13 | 1 |
| PubMed | 283 | 7 |
| **Total** | **1435** | **20** |

The query resulted in over 1435 studies (see Table 1) which were filtered according to the following criteria. Initially only studies published after 2005 that include evidence of the results were considered because the purpose of this study is to analyze the acquisition of knowledge while playing videogames.

We have also aimed to identify patterns in the development and use of videogames to teach health related topics to people, either patients or medical personnel. Therefore studies without empirical results but that identify patterns or best practices for videogames development in the health sector were also considered. Since the focus of this study was on digital games, we have excluded studies of non-digital games. After applying the described filters, 20 studies have been analyzed and reviewed.

### *3.1.1 Classification and Taxonomy*

As mentioned before, the focus of this study is aimed only at serious games whose main objective is learning and/or behavior change while keeping the user engaged (and hopefully having fun).

There are different domains which the outcomes and impacts of serious games can be classified. Wouters et. al (Wouters, van der Spek, & van Oostendorp, 2009) proposed four types of learning outcomes that games might display: cognitive, motor skills, affective and communicative. The studies included in the analysis were classified (1) determining the health-related domain of the study and (2) applying the taxonomy described by Wouters et al. to the studies.

## 3.2 Results Classified by Purpose of the Investigation

Following tables 2 to 6 presents the studies classified according to their main characteristics such as the skills promoted (cognitive, motor, affective-communication) or the kind of best practice proposed or the survey done. For each case, some basic information (authors, effects and control group) is presented describing the domain of application and a very brief summary of the results.

Table 2. Cognitive skills

| Study | Domain | Summary/Results |
|---|---|---|
| **Author/s:** (Good et al., 2014)<br>**Control Group:** General<br>**Effects:** Positive | Breast cancer | Crowdsourcing the game "The Cure" as a means to capture expert knowledge from its players showed a significant valid knowledge gain. The aggregation of the collected data helped identify the relation between genes and concepts such as cancer, disease progression, and recurrence. |
| **Author/s:** (Reichlin et al., 2011)<br>**Control Group:** Adults<br>**Effects:** Positive | Prostate cancer | "Time After Time" is the serious game used to enhance patient knowledge about prostate cancer and to increase their confidence in decision making. Patients' acceptance of the serious game was positive however they expressed concerns about some usability problems. |
| **Author/s:** (Vourvopolous, Lucía Faria, Ponnam, & Bermudel i Badia, 2014)<br>**Control Group:** General<br>**Effects:** Positive | Cognitive rehabilitation | In this paper the design and implementation "RehabCity" (online serious game designed for the rehabilitation of cognitive deficits) is presented. The validation concluded that the game is a valid tool for patients with cognitive deficits derived from a brain lesion. |
| **Author/s:**(Sajjad, Abdullah, Sharif, & Mohsin, 2014)<br>**Control Group:** Young<br>**Effects:** Positive | Brain tumor | A therapeutic 3D game with cognitive behavioral effects on the **children** suffering from brain tumor has been tested in three hospitals. The results showed that the game has been effective in recovering from psychological illness related to brain tumor. |
| **Author/s:** (Muehrer, Jenson, Friedberg, & Husain, 2012)<br>**Control Group:** Young<br>**Effects:** Positive | Cytology | The post quizzes of 161 participants showed significant improvement after playing the game for approximately one-hour sessions. There are other challenges such as a reliable internet connection. |

Table 3. Motor skills

| Study | Domain | Summary/Results |
|---|---|---|

| | | |
|---|---|---|
| **Author/s:** (Robert Kleinert et al., 2015)<br>**Control Group:** General<br>**Effects:** Positive | Training clinical knowledge | The development of an immersive patient simulator, and its validation with 25 students resulted in positive effect on reproduction of trained content and a high level of acceptance. By navigating the virtual world, students interact with other patients and face the consequences of different decisions. |
| **Author/s:** (Boulos & Yang, 2013)<br>**Control Group:** General<br>**Effects:** Indecisive | Physical activity | Exergames (games that require physical activity to play) are continuously gaining acceptance. The study concludes that further research is needed to quantify the exact benefits of GPS exergames and apps. |
| **Author/s:** (Kleinert et al., 2015)<br>**Control Group:** General<br>**Effects:** Positive | Training clinical knowledge | The study of all immersive patient simulators concluded that out of the 9 available IPSs, only one showed positive learning outcome and content validity. The use of IPSs could enhance (content quality and validity) the learning of procedural knowledge. |
| **Author/s:** (Corredor, Gaydos, & Squire, 2013)<br>**Control Group:** Young adults<br>**Effects:** Positive | Biology | Students played a game with the objective of learning biology. Students that played the game described a higher number of temporal-dependent interactions as measured by the coding of verbal protocols and drawings that students who used diagrams and texts to learn the same topic. |

Table 4. Affective and communicating skills

| Study | Domain | Summary/Results |
|---|---|---|
| **Author/s:** (Caldwell & Christiansen, 2013)<br>**Control Group:** General<br>**Effects:** Indecisive | Physical activity | A Patient Empowerment (PE) videogame was used for improving cancer patients' motivation for physical exercise. This video game concept can be applied to neurological, metabolic or cardiovascular diseases too. |

| | | |
|---|---|---|
| **Author/s:** (Kazemi, Cochran, Kelly, Cornelius, & Belk, 2013)<br>**Control Group:** Young adults<br>**Effects:** Positive | Drinking habits | Mobile Technologies (mHealth) show potential at informing of alcohol-intervention programs and increase their accessibility. The students showed a positive response that would be valuable to college students. |
| **Author/s:** (Beale, Marin-Bowling, & Guthrie, 2006)<br>**Control Group:** General<br>**Effects:** Positive | Self-care and health-related problems | Patients indicated a high degree of acceptability (4.1 out of 5) and credibility (3.7 out of 5). The game 'Re-Mission' would be a useful addition to the psycho-educational resources available to treatment teams. |
| **Author/s:** (Li, Chung, & Ho, 2011)<br>**Control Group:** Children<br>**Effects:** Positive | Cancer | The effectiveness of therapeutic play (virtual reality computer games) in minimizing anxiety in children hospitalized with cancer is supported by the empirical evidence of this study. |
| **Author/s:** (Fujiki, Kazakos, Puri, Buddharaju, & Pavlidis, 2008)<br>**Control Group:** General<br>**Effects:** Positive | Physical activity | An experiment demonstrates that players are engaged in NEAT-o-Games (obesity prevention and intervention) and become more physically active while having fun. |

Table 5. Design and best practices

| Study | Domain | Summary/Results |
|---|---|---|
| **Author/s:** (Preston, 2013)<br>**Control Group:** General<br>**Effects:** Indecisive | General health | Describes a "rapid and scalable" process for leveraging the creativity of the developers to design games with the potential for behavioral change intent among players. The results of these games on its players are not fully explained. |
| **Author/s:** (Ushaw, Davison, Eyre, & Morgan, 2015)<br>**Control Group:** General | General health | This study summarizes the best practices of designing engaging serious games that do not distract from the main health benefit. |

| | | |
|---|---|---|
| **Effects:** Positive | | It is based on the creation of a number of successful titles for over twenty years of work. |
| **Author/s:** (Barjis, Samarrai, & Smith, 2009)<br>**Control Group:** Adults<br>**Effects:** Positive | Cytology | The study describes the design, modeling and simulation of a 3D virtual Cell as a Game and the design decisions taken to enhance the learning process of its players. |
| **Author/s:** (Thompson et al., 2010)<br>**Control Group:** Children<br>**Effects:** Indecisive | Behavioral change | This paper describes how behavioral scientists and video game designers can team up in order to develop a highly effective serious game that promotes behavior change and also entertains its players. Exposes the best practices and design concepts involved in the game creation process. The validation results and the outcome evaluation of the game presented (DIAB) were not clarified in the study. |

Table 6. Surveys

| Study | Domain | Summary/Results |
|---|---|---|
| **Author/s:** (Kron, Gjerde, Sen, & Fetters, 2010)<br>**Control Group:** Young adults<br>**Effects:** Positive | Medicine in general | A survey with 217 medical students has been performed in this study to know whether they approve the usage of new media teaching methods in medicine. Students showed a very high level of acceptance (even those that do not play videogames). |
| **Author/s:** (Scanlon, O'Shea, O'Caoimh, & Suzanne Timmons, 2015)<br>**Control Group:** Older adults<br>**Effects:** Indecisive | Acquisition of Knowledge | Older adults have poor skills using technology.Usage of technology in the care, rehabilitation, and monitoring of older adults is low. Further research is required to determine how people's attitudes toward technology usage influence its utility in clinical practice. |

This review resulted in a large set of serious games considered but even in that situation we were not able to find specific guidelines directly applicable for developing a serious game that both rely on medical imaging and could train medical personnel with our technical requirements. For instance, most of the games from the previous study were designed to use an internet connection (not suitable for our case study) and to be executed on PCs where the size or the game and its assets didn't suppose a technical problem. But still this analysis gave us some general recommendations and guidelines about the design, development, and validation process of a serious game for health. However, we need to review more specific game applications for medical training that rely on the use of medical images.

## 3.3 Games and Game-Like Applications for Medical Training Using Medical Imaging

The studies from this chapter are extracted from the paper Design and development of a serious game for medical training in cytopathology (Rotaru & Rosemary, 2016), more information can be found in the actual paper or in the Appendix A - ACM DIGITAL HEALTH 2016, where the full paper is included.

As previously stated, our goal is to develop a serious game for cytopathology training of personnel in very different environments. The game should run on both PC and low-cost Android tablets (e.g., USD 50 Kindle Fire 7 inches). The tablet game should be fully functional without an internet connection. This means that all the assets of the game must be packed with the game and cannot be downloaded dynamically, which may imply size-related concerns for the Android version of the game.

The games and applications reviewed, in this second analysis, have the purpose of training personnel (expert or non-expert) in medical related domains relying on medical imaging.

### *3.3.1 MalariaSpot - Cell Map Explorer*

MalariaSpot is a game-like application oriented for crowdsourced collaboration in Malaria diagnosis done by non-medical experts (Luengo-Oroz, Arranz, & Frean, 2012). It is available online for free

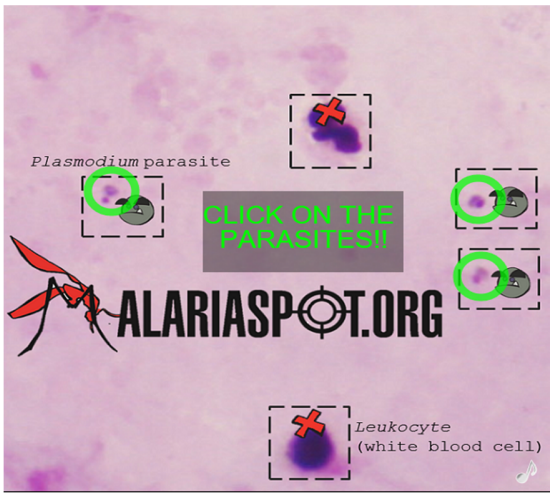



Figure 2. Screenshot of the game MalariaSpot briefly showing some gameplay instructions

("MalariaSpot - Crouwsourced serious game fighting malaria," n.d.).

The objective of this game mechanic is to tag as many parasites as possible in an image for a period of time (e.g., one minute). As shown in Figure 2. Screenshot of the game MalariaSpot briefly showing some

gameplay instructions There is an initial introduction screen as mini-tutorial explaining what is a parasite and what it is not and how to interact with the background image.

During the game, if the player finds all the parasites of an image in the allocated time, a new image will be loaded up. Each image should be considered as a level of the game, therefore, a player can analyse several images (levels) in a single game.

- There are several game mechanics to reinforce the player engagement.
- The players receive continuous feedback. For instance, each click is represented with an icon that indicates a correct or incorrect selection, shown in Figure 3. MalariaSpot gameplay where three parasites have been correctly tagged. Furthermore, if a player misidentifies a target and clicks in the wrong one (e.g. on a leukocyte) there is a penalization reducing the remaining time available and the final score of the level.
- There is one image per level. Only one part of that image is shown at the screen a time and the user can move around it looking for all the parasites.
- The difficulty is increased with each level by increasing the score and time penalty for wrong targets identified.
- The players' score is tracked and included in a table of high scores (leaderboard) for that day and for the week.
- Sensors from the mobile device could be integrated with the mechanics of the game. For instance, the gyroscope can be used for the image exploration.
- The targets could change to identify other type of events on the map, not just parasites.

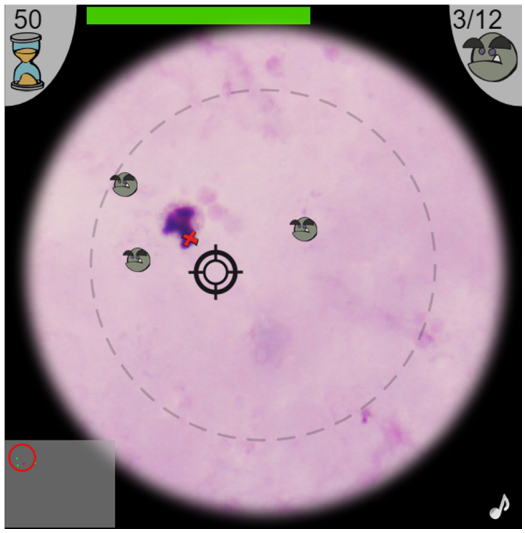



Figure 3. MalariaSpot gameplay where three parasites have been correctly tagged

In our case, the image size could suppose a technical problem for the device's video memory, especially if the image is very large. A possible solution can be to slice the image in multiple tiles that are used to render the entire map. The tiles can be loaded and unloaded from the graphic memory dynamically as they are shown in the screen or hidden.

### *3.3.2 Virtual Microscopy Adaptive Tutorials - Quiz Challenge*

The cytopathology virtual microscopy adaptive tutorials (VMATs) uses an adaptive eLearning platform that includes a whole slide images for pathology education as a content for effective training of both students and pathology specialists (Simone, Pryor, Belinson, Salisbury, & Velan, 2015). VMATs have different question formats (e.g., multiple-choice, drop-down lists, drag and drop type questions, fill-in-the-blank) to enhance interactivity and motivation. VMATs are designed to "adapt" to the user's decision making and aid with possible misconceptions through feedback. Even if it is not a game there are several mechanics that provide a game-like behavior (at least they could be considered as a gamified interaction).

As we can see in Figure 4. Example of a quiz of a VMAT case study, a quiz challenge in this type of gameplay is composed of a text-based question on one part of the screen and a related image (slide) on the other side of the screen. The mechanic is pretty simple yet is integrated with different gamification methods.

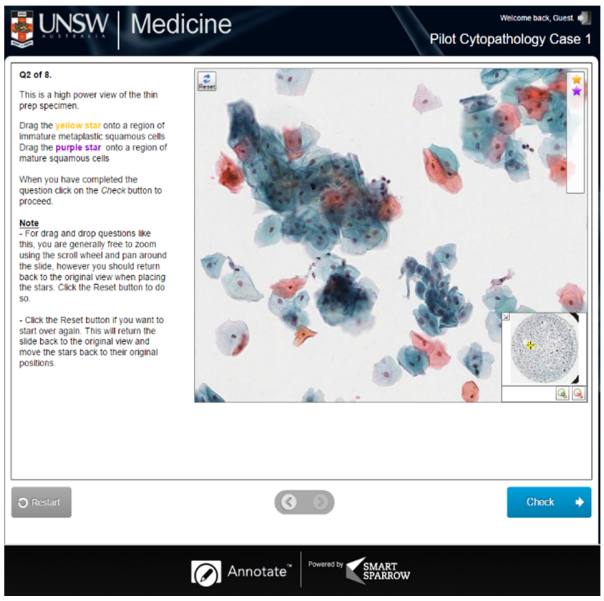



Figure 4. Example of a quiz of a VMAT case study

- To increase the engagement the question format can change (e.g., multiple-choice, drop-down lists, drag and drop type questions, fill-in-the-blank).
- Immediate feedback can be provided following the user's/learner's submission of responses. The feedback may contain useful educational information about the quiz or a specific area on the slide. The area can be highlighted using different colours or shapes on the feedback screen.
- The game randomly select a subset of questions each time the game is played, ensuring its replayability value.
- The players are challenged to answer as many quizzes as possible (and accumulate points for correct answers) in a one-minute round.

VMATs approach can be reused in our project as it is oriented to cytopathology and it takes as a content starting point a set of teaching slides. But there is a fundamental difference as VMATs is only for PC and it relays in a continuous internet connection as analytics is continuously collected from these interactions for adaptive purposes and to provide evidence about the effectiveness of the quizzes. The size of the slide image can also present technical problems for the device's hardware, if it is too large. A possible solution is to reduce the size of the slides to an acceptable level.

### *3.3.3 BioGames - Positive vs. Negative Game*

BioGames is a training game that helps identify malaria infected cells (Mavandadi et al., 2012). This game challenges the player to identify infected and uninfected cells. Figure 5 shows that for each level a set of cells are displayed to the player. The player must identify the infected cells before the time runs out (but the timer is optional). The player has to label the available cells either as "positive", "questionable" or "negative". The player receives a score depending on how well he or she performed identifying infected cells. There is also a progress bar as a visual feedback that indicates the advance made in the game. The players' top scores are integrated with a leaderboard system to promote competition and to improve engagement.

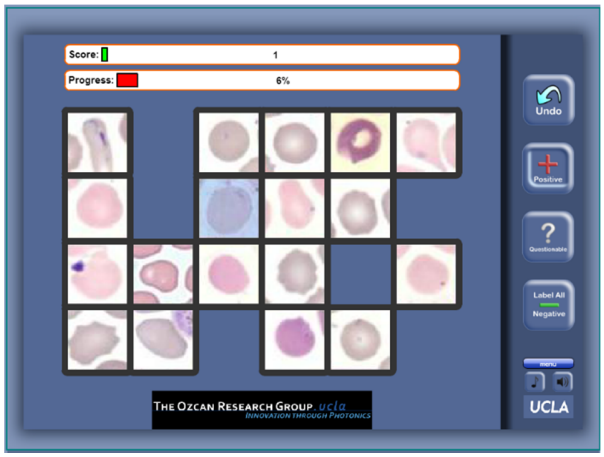



Figure 5. BioGames in-game example

An introductory tutorial guides the player with the instructions, displayed in Figure 6. This type of game mechanics can be easily integrated with an analytics tracking system to collect useful data. To reuse a similar approach in our game a technical issue that must be considered is the size of the final game. Since all the images must be packed inside the game (cannot be downloaded dynamically), the number of cell images available to the player might increase the size of the game considerably. If there are too many images available the size of the final game can grow considerably. This might become a problem for Android devices.

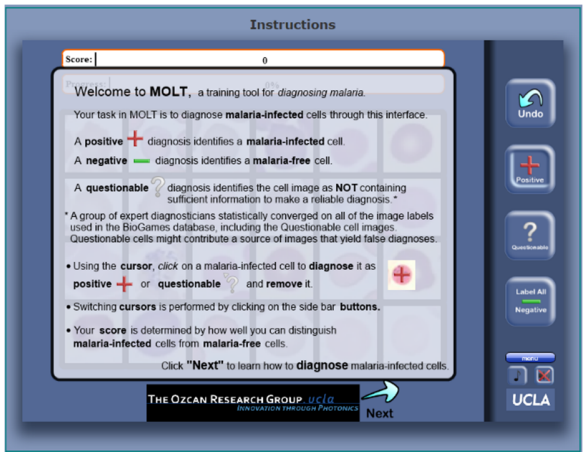



Figure 6. Brief tutorial showing game instructions

### *3.3.4 Cell Slider*

*Cell Slider* is a game developed in collaboration between Cancer Research UK and citizen science experts Zooniverse. The players are asked to classify and identify cells on archived cell samples. Players must examine tumor tissue samples and identify cancerous cells. In order to ease the identification process, players are asked simple questions about what they can see in the image of a tissue sample. To increase the validity of the answers the images are reviewed by several people. The results obtained are used to help with the breast cancer research (considering different types of breast cancer and how well they respond to different forms of treatment). This analysis greatly reduces the amount of information that scientists have to analyze.

As we can observe the Figure 7. *Cell Slider* identification of cancerous cells7, *Cell Slider* combines game mechanics from BioGames, by asking its players to identify specific tissue samples with a higher focus on each

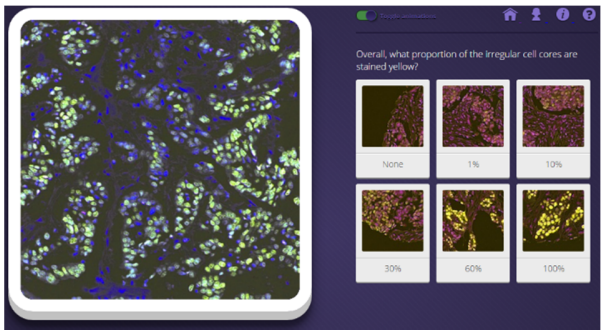



Figure 7. *Cell Slider* identification of cancerous cells

image individually, and VMATs, by displaying a user interface similar to a quiz challenge composed of a text-based question and a related image.

## 3.4 Conclusions

In this chapter we identify the best gameplay mechanics that are suitable for training medical personnel. We started doing a literature review and classifying over 20 studies on Serious Games for different health-related domains (physical exercise, psychology, different infectious diseases, cytopathology, cancer, etc). According to the preliminary results of this study, there is still a lot of ground to be explored in order to reach a mature methodology for the development of these types of games.

We were neither able to find general guidelines about how to develop serious games in ehealth neither specific guidelines directly applicable for developing a serious game that rely on medical imaging. In the best-case scenario the reviewed studies only provided guidelines for developing games aimed at their specific health domain. Only in specific cases there is described how this could be applied in a different domain. For instance, Craig Caldwell et al. (2013) (Caldwell & Christiansen, 2013) briefly explains how his Patient Empowerment (PE) videogame could be applied not only for cancer diseases but also for other diseases such as neurological, metabolic or cardiovascular diseases.

In order to identify the best practices for designing a game to train pathology personnel, we reviewed several games and game-like applications designed to train in the analysis of medical images (e.g. MalariaSpot, BioGames, Cell Slider and VMATs).

After analyzing these games we've realized that there are different approaches and game mechanics that can be used in a game or game-like application to train medical personnel in medical imaging and more specifically for cytopathology:

- Introductory story to set up the context and motivate the player.
- Tutorial explaining how to use the application or game.
- Succession of levels that get incrementally more difficult.
- Giving continuous feedback to the player through the gameplay experience.
- Different gamification features such as progress and scores tracking, leaderboards system, and timers.

We have concluded that one possible design could be presenting the player with a short story to set up the context, followed by a concise tutorial explaining the game mechanics and ending with the player having to overcome different challenges (i.e. levels) that can be measured to assess the learner's progression. A challenge, in its simplest form, could be a multiple choice question about a concept, an image or a specific region of an

image where the user should identify if there is any anomaly or special circumstance. There could be a set of challenges, questions or puzzles from which a randomized sample is taken every time the learner starts playing a session. The challenges may also vary in difficulty as the learner advances and could have a gamification metric associated (allowing the creation of rankings between players) increasing the content diversity for each gameplay session. This design presents the content as a progression of events - initial story, basic concepts description, progressive challenges - that can be easily understood by the player. The initial story is a key element in the game's design, it provides a supporting narrative meant to address the learner's motivation and interest.

This initial design should be complemented including the consideration of the hardware and software restrictions of our project as the game should run both on PC and on low-cost tablets and be used to train cytologists in low resource countries where the availability of a stable internet connection is low.

Some of the analyzed games and game-like applications capture user interaction data with different purposes (e.g. adaptation, leaderboard) and we consider that this is the way to go even if in our case the game deployed in tablets cannot rely on a continuous internet connection. This will allow to include learning analytics techniques to better analyze the interaction for different purposes such as evaluating player learning and improving the actual game.

To conclude, we believe that the review done and the conclusions obtained about designing serious games for medical training in cytopathology provide a solid ground for developing our prototype. Next steps in the project is the research related with the technologies available for the creation of the initial prototype and the design and implementation of a solution.

# 4 Development Tools

On the previous chapter we had identified the gameplay mechanics that we wanted to implement in order to train cytologists in resource-limited areas. The next step is to research which is the most suitable development technology for our project. The studies from this chapter are extracted from the paper Tools and Approaches for Simplifying Serious Games Development in Educational Settings (Calvo, Rotaru, Freire, & Fernandez-manjon, 2016). The paper in included in the Appendix B - IEEE EDUCON.

There are a many of different tools available for digital game development, depending on factors such as game genre, purpose, platform, team size or available budget. These tools have in common the little support provided for any activity related to the design of the game. This limitation is especially relevant at the very beginning of the process, when the team is focused on capturing and evolving the concept of the digital game into a stable design ready for implementation.

We have analyzed the development tools and game frameworks available in the game development industry and classified them in four classes. This section provides a concise overview of the state of the art of in-game development tools, structured in four subsections.

### *4.1.1 AAA*

In the authoring tools industry, *AAA* – pronounced "triple A" – is a classification term used for authoring tools with the highest development budgets and levels of promotion or the highest ratings by reviewers. An authoring tool considered to be AAA is therefore expected to be high quality, and focused on the development of high quality games – a.k.a. AAA games. To achieve this goal, these tools provide optimized features needed in a game (artificial intelligence, physics engine, animations, level design, etc.) as well as support for the roles that take part in the development (programmers, graphic artists, testers and designers). Their user interface is intended for experts, focused on implementation and advanced features provided by the tool to create the game, and leaving aside the features that would allow for rapid prototyping of game ideas.

Among other features, these environments provide an engine that abstracts platform-dependent features from the game code. AAA authoring tools are designed by experienced programmers in game development, and manage different features such as assets, input/output, network connections, rendering pipeline, etc.

Regarding content creation, level designers can create new content for the game without modifying any code from the engine itself. These editors also provide authoring tools for animations and assets targeted at artists with little or no programming knowledge. *CryEngine* and *Unreal* Engine are two outstanding examples.

Developed by *Crytek* and *Epic Games* respectively, they were initially used to create two successful commercial entertainment games (*Unreal Tournament* and *FarCry*), and have been actively improved ever since. Both have been used in dozens of other commercial games across different platforms (consoles, PC and mobile devices) and feature multiple licensing tiers, which can represent a substantial portion of the game development budget. For example, the highest tier for Unreal Engine 4 requires 5% of the total game profits.

### 4.1.2 All in one

The AAA tools previously exposed are very powerful but at the same time their cost is too high for most developers. The team that created CryEngine numbers nearly 300 game professionals over the world. There are other semi-professional tools created for more reduced teams with lower budget available.

One of the most popular is Unity, an all-in-one (AiO) tool equipped with all the necessary modules to configure all the aspects of a game, including the ability to export for different platforms (consoles, PC, Mac, Web and mobile devices). Currently, it is also being used for mobile development by professional teams. Nintendo's Wii U video game console uses Unity as the default software development kit (SDK) including a free copy with each Wii U developer license.

AiO tools lack the native support for content creation which is done with other tools such as Maya and *3D Studio*.

### 4.1.3 Frameworks

The main focus of these tools is to provide a lower level technology, closer to the programmer. In this section we can find engines, frameworks and libraries focused on digital games, which can be commercial or free, and are designed so that the programmer makes use of its desired development environment (Eclipse, NetBeans, IntelliJ, Visual Studio, etc.). Examples are *libGDX*, *Ogre3D* or *Allegro*.

In general these tools require advanced game programming knowledge and can support different platforms.

### 4.1.4 Specialized tools

These tools have less functionality and potential but they simplify the game creation process in order to bring them to a more casual public. They are mostly used in the education, aimed at professors and students allowing them to make simple games with a high educational value. One of the first examples is GameMaker, developed by Mark Overmars. Scratch("Scratch - Imagine, Program, Share," n.d.) is another popular tool, which tries to teach programming to the kids while creating digital games(Resnick et al., 2009). *eAdventure* ("e-

Adventure e-Learning Games," n.d.) is a tool that started with the aim to provide teachers with the ability to create games devised for education. *eAdventure* focuses on the development of two different types of 2D *point-and-click* games: *first person*, the scene is seen through the eyes of the main player, and *third person*, where the player is represented as an avatar in the scene.

## 4.2 Comparison between the Different Kind of Tools

The use of the different types of tools has effects on the monetary costs, time, and the end result. It is important to have a deeper understanding about these tools and their applicability. Below we expound some properties to consider.

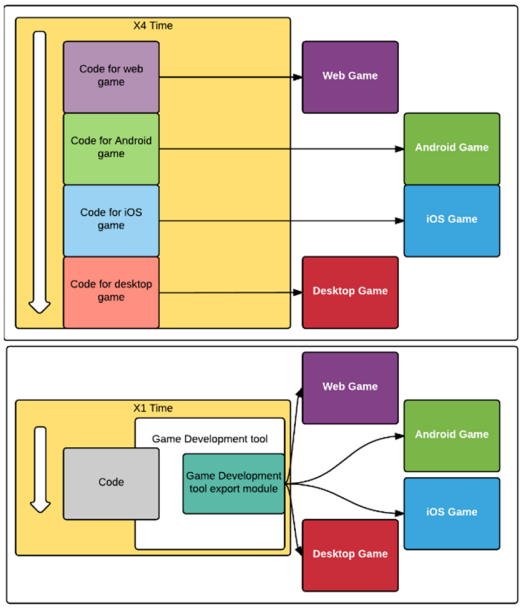



Figure 8. Multi-platform game development time costs comparison between the traditional method (top) and making use of a tool/framework (bottom)

### *4.2.1 Platforms*

One of the first things that should be considered in a game development is the set of target platforms that it should be runnable in. In serious games, the most common environment is the desktop, where the three main Operating Systems are Windows, Mac and Linux. With the expansion of mobile devices, it is increasingly frequent to find games developed for mobile platforms, such as Android, iOS or Windows Phone. Some tools do not have the capacity to export the game for all the available platforms, the game's target platforms can

suppose the difference between using one tool or another. In Figure 8 we can see the time cost required for developing a multiplatform game development framework or tool.

AAA tools are specialized in *AAA* games and deploy to the highest amount of platforms. Their engines can create games with the latest graphics quality, physics and technologies. On the other side, frameworks and specialized tools have more platform deployment restrictions. Generally they give support for one or two platforms (i.e. Windows and Android). libGDX is a framework that supports the development for several platforms such as: Windows, Mac, Linux, Android, iOS and HTML5.

Unity has the industry-leading multi-platform support. Unity can export to a very wide range of devices across different platforms. IOS, Android, Windows Phone, BlackBerry and Tizen are the main mobile devices platforms and are supported by Unity. For the desktop environment Unity can export to Windows, Windows Store Apps, Mac, Linux and Steam OS. Unity also targets the web through its Unity Web Player plugin or directly exporting to WebGL. The past and current generation of game consoles have sold an important amount of units, as we can see in Figure 9.

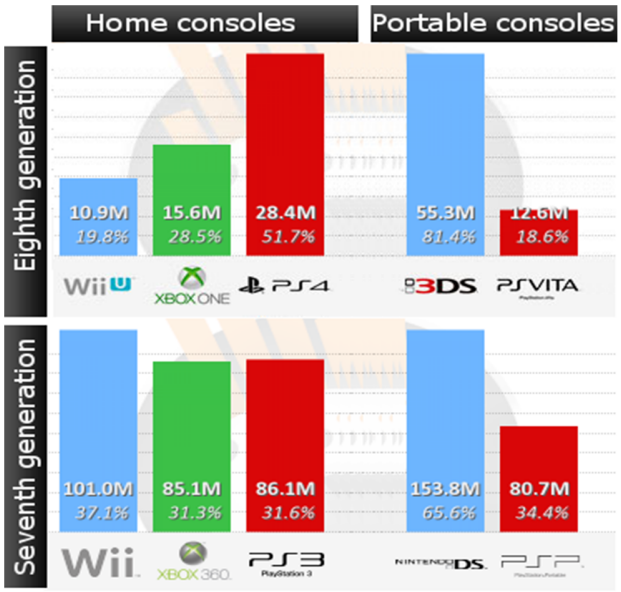



Figure 9. Consoles sales as of 14th November 2015, from http://www.vgchartz.com/

Unity allows the developers to target PlayStation 4, PlayStation 3, Xbox One, XBox 360, PlayStation Mobile, PlayStation Vita and Wii U free of charge. Publishing to a console requires an additional approval process that varies from platform holder to platform holder. Unity also provides support for Virtual Reality and Augmented Reality platforms such as Oculus Rift, Gear VR and PlayStation VR. Microsoft HoloLens

support has been announced - April 29, 2015 - and is still being developed as of November 2015. Other platforms supported by Unity are Android TV, Samsung Smart TV and Windows Universal Platform.

The difficulty to publish games on different platforms must be considered by the developers when choosing a desired engine or framework. For instance libGDX does a good job to ease the deployment to different platforms though there are several conflicts that might need to be resolved before deploying to specific platforms. The libGDX code base is written in Java and it provides a great cross platform integration between desktop - Windows, Mac, Linux - and Android devices as long as the developer keeps the source code compatible with Java version 6 during compilation. For the HTML5 deployment, libGDX uses Google Web Toolkit to compile the source to highly optimized JavaScript code that runs across all browsers.

Unity offers a faster solution by providing a user interface for the deployment process. Unity's solution requires less platform-specific knowledge from the developers and can speed up the development process. There is typically some minimal effort required, such as integrating with each platform's store for in-app purchases.

### *4.2.2 Game mechanics and interactions*

Another thing to keep in mind is the kind of game and the mechanics that must be developed. Games may vary in features, mechanics, and user interaction. Some tools only support the development of a specific type of games, especially the specialized tools –eAdventure focuses on the *point'n click* game development– that generally only allow the development of one or two genres. RPG Maker is a specialized tool targeting RPG games. AAA and all in one tools allow the creation of a much broader spectrum of game genres, e.g. platforms, first or third person shooters, adventures, simulations, strategy games, etc.

Specialized tools provide easier ways to create simple games, faster and with lower costs because these games require fewer scripts and elements, and their graphic user interface editor is highly optimized for the creation of a specific game genre. An objection is the difficulty to innovate or create news features in this kind of games. An example is eAdventure, which allows the creation of *point and click* games in few hours if the graphics resources are already created, with all mechanics and characteristics that characterize the adventure graphics games. On the other hand, a developer cannot easily introduce other game mechanics with eAdventure, e.g. shooter mechanics.

### 4.2.3 Features

In order to choose the best game development framework or engine, the nature of the graphics - 2D or 3D - must be taken in consideration. Some tools only can only make two dimensional games. It is necessary to keep in mind that not all the tools have the same eases for development.

Unity supports both approaches but it was designed mainly for 3D game development while libGDX has been originally designed for 2D games, it also supports 3D. Unity has officially supported 2D development since the release of its 2D module, introduced in Unity 4.3 (November, 2013). The 2D dedicated engine in Unity was introduced because of the higher difficulty of 2D games development, an example is the use of sprites. Unity's 2D module introduced tools to work with sprites and a physics specialized for 2D mechanics, tiles editor, sprite animations, masks, improved packages manager and a better performance for the 2D engine.

Game development frameworks are different than game engines because they provide different features. Game development frameworks are a collection of lower level libraries, tools and other frameworks used to create games. Game engines encapsulate powerful logic designed to create games and provide more features. Game engines may have a graphic user interface editor while a framework is mainly code based. Unity has a visual editor that can improve the development speed. The developer, has to consider whether the tool is actively supported or there is a risk to become discontinued. The frameworks and engines have to continuously update their features with the new emerging technologies, improving the game's development workflow.

Also, engines as Unity and Unreal have been designed considering the profiles of game developers and also designers, providing an easy-to-use *drag and drop* editor user interface. They also provide *asset stores*, repositories of different kinds of assets –graphics, sounds, game objects, etc. – that can be obtained for free or purchased.

Other important feature is whether the testing can be easily done on the development platform. Engines and tools that export to mobile devices allow the developers to test the game in the platform the game is developed. Having to export the game for a different platform every time a new feature must be tested or to find and fix a bug can be tedious. It is important that the tool allows to easily play the game in the same platform the game has been developed. Engines like Unity or Construct2 have this property, but not all frameworks, for instance libGDX has it but not AndEngine or E3roid.

### 4.2.4 License and Cost

The license of a game development framework or engine may suppose the difference between a developer choosing to use them or not. In order to be competitive, Unreal Engine's license has been made free

with the condition to pay a 5% royalty on games and applications released by the developers - after the first USD 3,000 of revenue per product per quarter.

LibGDX is licensed under Apache 2.0 and maintained by the community. The code is open source, available at GitHub. Developers can use it free of charge, without strings attached in commercial and non-commercial projects.

Unity has a free version - Personal Edition - available for developers with lower revenue than USD 100,000 in the previous fiscal year. Unity's Professional Edition license has a minimum cost of USD 75/month and packs additional features not available in the Personal Edition: customizable splash screen, Unity Analytics Pro, prioritized bug handling, Game Performance Profiling, Unlimited Revenue and Funding, professional editor skin, etc. Unity offers support to big enterprises with its tailored Enterprise Solutions, a package with additional features such as source code licensing, on-demand support and volume discounts available on Unity licenses. Unity Education aims to support learners, educators and educational institutions through special offers on licenses, content, services and resources - e.g. Professional Skills Standard and Curricular Framework -, a grand program for secondary institutions, discounted tickets for students and faculty to Unity's Unite developer conferences, etc.

### *4.2.5 Documentation*

The framework or engine documentation reduces the developer's learning curve and is an important factor to improve reusability and maintainability. The documentation is composed of several kinds of documents and contents such as wikis, code documentation, manuals and tutorials, example code and demos, books and additional third-party and community support.

Wikis help developers understand the abstract - and most important - concepts. Manuals and tutorials teach how to use the tools providing 'best practices' methods, code examples and demos. An active community helps the developers with specific doubts and misunderstandings which can be used to further clarify the documentation.

The community size is closely related with the efficiency, effectiveness, and the tool's growth potential. It is important to consider, not only the documentation, but also the community behind the tool. The community's main goal is to help the developers - indie or professional teams - resolve their unexpected problems and doubts. In addition, one of the key components of software engineering is the reusability - a.k.a. 'don't reinvent the wheel' -, and in this case is important to consider the contributions of the other game developers that use the same tool. Unity has an official documentation composed of source-code documentation, feature manuals and tutorials exemplifying the creation of different types of games. Furthermore, Unity has a community with easy find tutorials and developers eager to help with solutions.

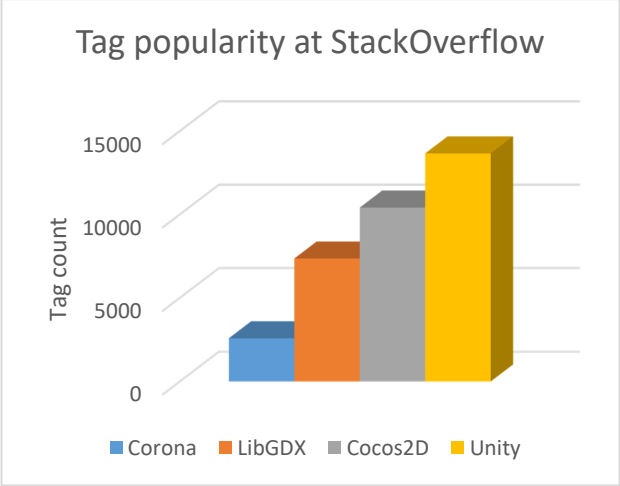



Figure 10. Tags popularity about some game development tools in StackOverflow community

As we can see in Figure 10 about *StackOverflow*, a developers community that covers a wide range of programming and statistical topics(Muenchen, Robert A., 2014), contains a large number of posts and associated tags, and the data are easy to obtain(Stanley & Byrne, 2011). As of November 2015 there are: +2500 tags about Corona framework, +7300 about libGDX, +10400 about Cocos2D, +13600 about Unity.

### *4.2.6 Performance*

Performance is another key factor considered by the developers when choosing a game development framework or engine. By using game development frameworks, developers can achieve better performance than a fully-featured game engine, since they don't provide so many canned game features. Some game development frameworks - libGDX - allow the developers to have access to low level programming instructions to squeeze all the performance of the underlying graphics technology - OpenGL or DirectX.

LibGDX focuses on avoiding garbage collection for Dalvik/JavaScript by careful API design and the use of custom collections. LibGDX provides a single code base for all the platforms which greatly increases its maintainability. libGDX recommends using the Facade programming pattern to write platform specific code and provides static methods to check the platform in which the code is being run. The code base is written in Java allowing the use of software patterns by the developers to improve its readability and maintainability. Unity heavily emphasizes on its GUI editor to *drag and drop* entities and components in the game scene with the possibility to add behaviors as C# or JavaScript code snippets. Unity also provides environmental variables to easily check the current platform, allowing the execution of platform specific code. The GUI editor allows fast changes in the objects' positions or appearances, and uses the same code snippets for all the targeted platforms.

### *4.2.7 Educational assessment*

The most amount of frameworks and engines don't consider the educational assessments, but the games creates for learning with this tools can use some services or tools to make educational assessments and measure the results to know the level of learning that it provide.

Most game development frameworks and engines don't provide direct support for educational assessment. libGDX has no direct support for any educational assessment mechanism. The developer must integrate any specific educational feature that he requires with the additional cost of time and resources. Developers using frameworks that lack the support for educational assessment mechanics should not develop the required features' infrastructure from scratch. As we can see in Figure 11, developing a learning analytics infrastructure from scratch is an effort consuming activity due to its complexity.

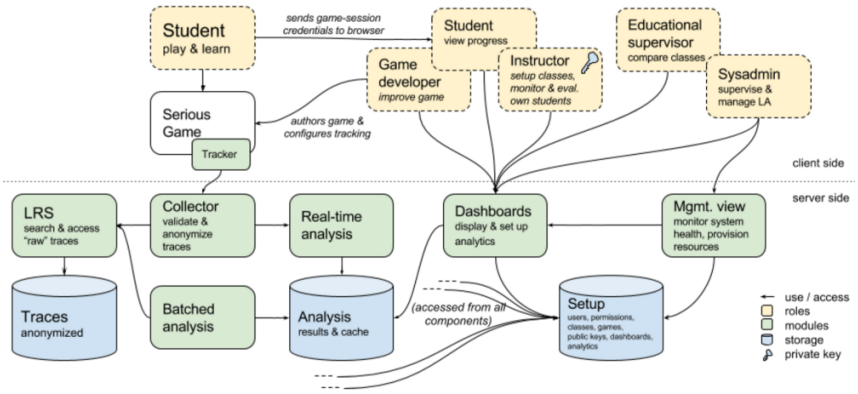



Figure 11. Learning Analytics architecture at RAGE project. http://rageproject.eu/

Instead it is recommended to integrate a service that provides a similar functionality out of the box. There are different analytics services available, that provide different features - platform, cost, specific reports, data storage policies, etc. - that should be considered before choosing one. Google Analytics is the de-facto industry standard for web and cloud-based analytics and dashboards. It provides a RESTful API that can be used through all platforms but lacks support for official Software Development Kits on many platforms (e.g. Sony's PlayStation 4 or Microsoft's Xbox One). Google Analytics offers specific data visualization tools and the ability to define custom dashboards. Game Analytics is an alternative to Google Analytics that adds an extra layer of functionalities and reports designed for games. Flurry Analytics offers similar features as Google Analytics with some differences - i.e. the data is not sampled even if the, Google Analytics' free license samples the dashboard data if it surpasses 500.000 visits. Although these analytics services, generally, weren't created specifically for learning assessments, they can be adapted for this propose.

Unity offers, in its Professional Edition license, the Unity Analytics Pro feature. Unity Analytics Pro gathers data from the game and transfers that data to a cloud-based data store. The gathered data is processed, analyzed and sent to the Unity Analytics Dashboard. Like Google Analytics, it is very business oriented but can be adapted to provide a layer of learning analytics functionalities on top of it.

eAdventure was designed to create games with educational purposes in mind. It allows the developer to integrate specific educational features - adaptation profiles and evaluation profiles - in the game. Each type of profile is composed by a set of rules where a condition must be satisfied in order to trigger a series of effects. Developers can define different ways to evaluate or adapt the game in order to aim the game to different kinds of contexts and public. Evaluation profiles can generate XML reports - targeted to a learning server - and HTML reports for a game instructor or player. Adaptation profiles are executed when a set of external properties are satisfied, at the start of each game scene. An adaptation profile execution may change the content of the scene in terms of available objects, puzzles difficulty, etc.

Realising an Applied Gaming Eco-system (RAGE) is a project developing an open source infrastructure to deploy learning analytics and other educational assessment features. RAGE aims to develop, transform and enrich advanced technologies from the leisure games industry into self-contained gaming assets that support game studios at developing serious games easier, faster and more cost-effectively. The project assets will be available along with a large volume of high-quality knowledge resources through a self-sustainable ecosystem, which is a social space that connects research institutions, gaming industries, intermediaries, education providers, policy makers and end-users. One of these assets is the Learning Analytics asset that allows teachers to monitor students and receive alerts and warning depending on the student's game state, progress or score. The learning analytics asset provides a tracker client available in different programming languages -

JavaScript, Java and C# - that can be integrated with other frameworks and game engines such as libGDX and Unity.

## 4.3 Conclusions

After identifying the gameplay mechanics that we want to implement in the previous chapter, we decided to identify the best development technology for our application. We did a study of the different options available in the market and discussed the two main candidates libGDX and Unity 3D.

The great economic importance and evolution in games has improved the tools used to author them, resulting in the growth of the game development community and the creation of a broad spectrum of authoring tools, some aimed at the creation of specific games – e.g. eAdventure, *point and click* games – others designed to create a variety of game genres, such as Unity.

While there are tools to create all kinds of games, there is still no single "best" way to develop an engaging and effective game. The developer must choose the best options depending on the game requirements, and considering different factors such as the target platforms, the desired interactions and mechanics, the game genre, learning objectives, the time constraints, resources and supported technologies. Additionally, an active and supportive community can help solving many of the project's problems.

A game development framework might be preferred in cases where the developers already know the required programming language and the game requirements are aligned with the framework's features. Frameworks are chosen over game engines by developers that want to have full control of all aspects of the game development and by small work teams. Developers with high time constraints may choose a specialized authoring tool in order to optimize the time spent developing the game. *All in one* authoring tools might be the best option for most of the developers since they balance the amount of provided features, learning curve and targeted game mechanics. Unity is the most influential *all in one* tool, excelling in many of its provided features, and is able to compete with *AAA* tools in terms of graphics excellence while, at the same time, being able to support the highest amount of platforms available in the market and compete with game development frameworks in terms of control over the game for the developers.

But from the educational point of view the most of tools lack of learning and assessments features to evaluate the players and their knowledge or skills learning. Only some concrete specialized tools have these features. And, even in those cases, sometimes people with low programming knowledge can have problems creating the assessments assets for their games, generating conflicts when a non-specialized person needs to create a game for educational usage.

Moreover, developing games with technologies that might get discontinued and deprecated is a risk that can be averted by carefully choosing a good development framework or engine. For example, *Science Pirates: The Curse of Captain Brownbeard* is case of an educational game that is now difficult to play because of an obsolete technology. The proposed solution is to invest in maintainable game development frameworks or engines that can deploy to multiple platforms from the same codebase, that are properly documented, have the support of active communities and provide features that can satisfy the game's requirements.

We had to choose between Unity 3D or libGDX. We compared both options considering different aspects such as the target platforms (both options provided support for HTML 5 web browsers and Android), documentation (both options have a complete documentation), license and costs (both are free and their license don't conflict with our project), community (both options have a very active community with continuous new games being developed) and performance (libGDX has an overall greater performance output than Unity 3D). Since we had to deploy out application on low-cost Android devices, the performance output of the development technology is of utmost importance. LibGDX is the framework selected. It provides the performance needed to run the application on low cost Android devices and supports the deployment to multiple platforms using the same codebase, including the required platforms of this project (Android and HTML 5). Moreover, the learning curve of this development framework was not an obstacle because the author of this work has a deep understanding of the framework.

# 5 Design and Implementation of an Integrated System

The previous research served as a base to identify the correct technologies needed to implement the project and satisfy its requirements. As we can observe in Figure 12, the integrated system consists of (1) a game-like application (client) that runs on low-cost Android devices and HTML 5 web browsers, (2) a backend service that exposes a RESTful API for the management of the data model used by the client and (3) a web-based editor interface for the creation of content for the clients without the requirement of a developer. The following sections will proceed to describe the game-like application (client) and the backend service with the editor web interface.

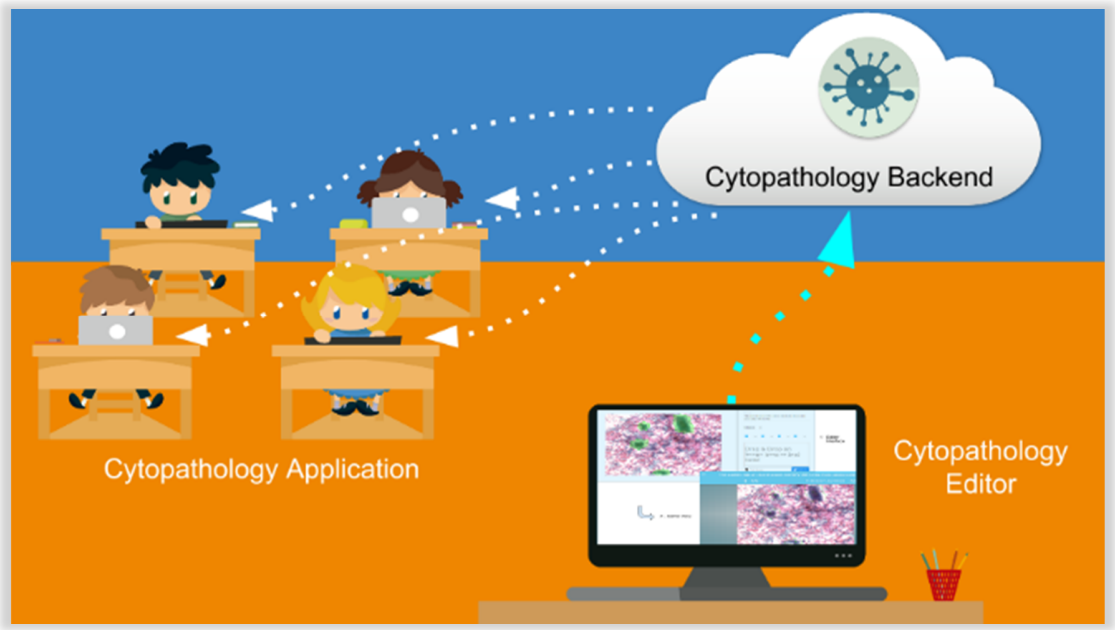



Figure 12. Development framework

## 5.1 Cytopathology Application

At this point we had already defined out development framework for the application (libGDX) and had already identified the best gameplay mechanics that we wanted to implement. An initial story to set up the context for the player, a tutorial explaining how to use the application, a level-based system that gets

incrementally more difficult, continuous feedback throughout the gameplay and different gamification features such as progress and score tracking, leaderboards, timers, etc.

The Cytopathology Application is created using the libGDX game development framework and Java programming language. It consumes the backend RESTful API to obtain the content of the system. The content of the system is composed of Challenges (levels) and Courses. A Course is a set of challenges with the same topic. Courses are used for grouping challenges and filtering through challenges.

Cytopathology Challenge is composed of different challenges that the student must solve to receive a grade based on the solution given. Each challenge can be seen as a small level in a videogame. A challenge-based game mechanic allows us to structure the displayed content easily in different types of gameplay mechanics described below. Furthermore it's easier and faster to understand by the students since all the videogames take advantage of the concept of a level that must be completed. Our design also groups challenges by courses. A course is a group of challenges that share the same topic. By grouping challenges, we were able to define exams for each course. An exam is gameplay mechanic where the student must answer different challenges in a row from the same course. A grade will be received when the student performs an exam. Each type of challenge has its own JSON format definition. There are 5 different types of challenges: Multiple Image Answer Question, Multiple Answer Question, Drag and Drop, Fill the Options and Highlight an Image Area. Figure 13 displays the JSON format of a simple Multiple Answer Question type of challenge.

```
{
    "class": "es.eucm.cytochallenge.model.TextChallenge",
    "imagePath": "superficial sq cell 60x.jpg",
    "difficulty": "EASY",
    "textControl": {
        "class":
"es.eucm.cytochallenge.model.control.MultipleAnswerControl",
        "text": "What type of cell is the pink cell?",
        "answers": [
            "Intermediate squamous cell",
            "Parabasal cell",
            "Squamous metaplastic cell",
            "Superficial squamous cell"
        ],
        "correctAnswer": 3
    }
}
```

Figure 13. JSON definition of a Multiple Answer Question challenge

Moreover, a challenge-based gameplay can be easily integrated with an analytics framework to track the progress of the students and improve the feedback the teachers receive through detailed dashboards. The

dashboards can display detailed information about the game such as each student and its progress, visualizations about each challenge (and course) and its completion rate and general information about the gameplay session.

For the lifecycle management of our application we use Gradle. Gradle is an open source build automation system that uses a Groovy-based domain-specific language (DSL) instead of the traditional XML used by Maven. Moreover, Gradleis a dependency management and build system.

A dependency management system is an easy way to pull in 3rd party libraries into our project, without having to store the libraries in your source tree. Instead, the dependency management system relies on a file in your source tree that specifies the names and versions of the libraries you need to be included in your application. Adding, removing and changing the version of a 3rd party library is as easy as changing a few lines in that configuration file. The dependency management system will pull in the libraries you specified from a central repository (in our case Maven Central) and store them in a directory outside of your project.

A build system helps with building and packaging the application, without being tied to a specific IDE. This is especially useful if we use a build or continuous integration server, where IDEs aren't readily available. Instead, the build server can call the build system, providing it with a build configuration so it knows how to build your application for different platforms. In case of Gradle, both dependency management and build system go hand in hand. Both are configured in the same set of files.

We have divided our application in multiple projects in order to export for different platforms while having the same codebase. The common codebase is available inside the Core project, while platform specific code is written inside the corresponding platform project (e.g. Html, Desktop for testing purposes and Android). See Figure 14 for the layout of the Cytopathology Application project.

```
settings.gradle           <- definition of sub-modules. By default core, desktop,
android, html
build.gradle              <- main Gradle build file, defines dependencies and plugins
gradlew                   <- script that will run Gradle on Unix systems
gradlew.bat               <- script that will run Gradle on Windows
gradle                    <- local gradle wrapper
local.properties          <- Intellij only file, defines android sdk location

core/
    build.gradle          <- Gradle build file for core project*
    src/                  <- Source folder for all the game's code

desktop/
    build.gradle          <- Gradle build file for desktop project*
    src/                  <- Source folder for the desktop project, contains Lwjgl
launcher class

android/
    build.gradle          <- Gradle build file for android project*
    AndroidManifest.xml   <- Android specific config
    assets/               <- contains for the graphics, audio, etc.  Shared with
other projects.
    res/                  <- contains icons for the app and other resources
    src/                  <- Source folder for the Android project, contains android
launcher class

html/
    build.gradle          <- Gradle build file for the html project*
    src/                  <- Source folder for the html project, contains launcher
and html definition
    webapp/               <- War template, on generation the contents are copied to
war. Contains startup url index page and web.xml
```

Figure 14. Layout of the Cytopathology Application project following Gradle build system

The Cytopathology Application source code is available at GitHub as a repository with 50 commits as of September 2016.

## 5.2 Backend

In order to design a standardized and extensible integrated system a RESTful application programming interface has been defined. The RESTful API aims at exposing a standardized interface for the management of the data of the system. The system data is composed of different types of Challenges and Courses. Moreover, in order to protect the resources of the system added functionality for the management of user accounts, authorization and authentication. This design allows the extension of the system by external developers easily by using the RESTful API without knowing the underneath implementation of the system. For instance, an external developer only needs to know the specification of the RESTful API to create a new client for another platform or with added functionalities.

In order to decide what data format to use (XML or JSON) we decided to study the current state of the art of both formats. The newest applications and technologies are using the JSON format instead of XML for the definition of their data model. Nowadays, XML is actively being replaced in favor of JSON as a format for data models. JSON (JavaScript object Notation) is slightly faster to write and more human-readable, compared to XML Furthermore, non-relational databases that are document-oriented (e.g. ElasticSearch or MongoDB) can be easily integrated with a JSON data-model because they also store the information following a JSON format. We have decided to use JSON as the format of our data model for the storage of Challenges and Courses. Moreover, the development framework of the Cytopathology Application client (libGDX) provides utility classes for the parsing of JSON documents which facilitates the communication with a RESTful API.

The next logical step was to decide what technology to use for the backend development. The author of this work already had knowledge of NodeJS, an open source cross-platform runtime environment that has been written in JavaScript, which makes it an exceptional choice for web applications and backend service development. Using JavaScript as a programming language for the backend makes it easy to send and synchronize the data between the backend and the frontend web editor. JavaScript integrates easily with a JSON data format. The Cytopathology Application client development framework (libGDX) also provides utilities for the management of JSON based data formats. After deciding that we wanted to use we studied the characteristics provided by the main JavaScript platform, NodeJS. NodeJS uses V8 engine by Google, which makes it fast, with an exceptional running speed. Furthermore, NodeJS encourages sharing through the Node Package Manager (NPM) by including thousands of packages which helps the development of effective

platforms and solutions. Since NodeJS allows the development of server-side (backend) and client-side (frontend) code, the frontend web editor is also created using NodeJS and AngularJS.

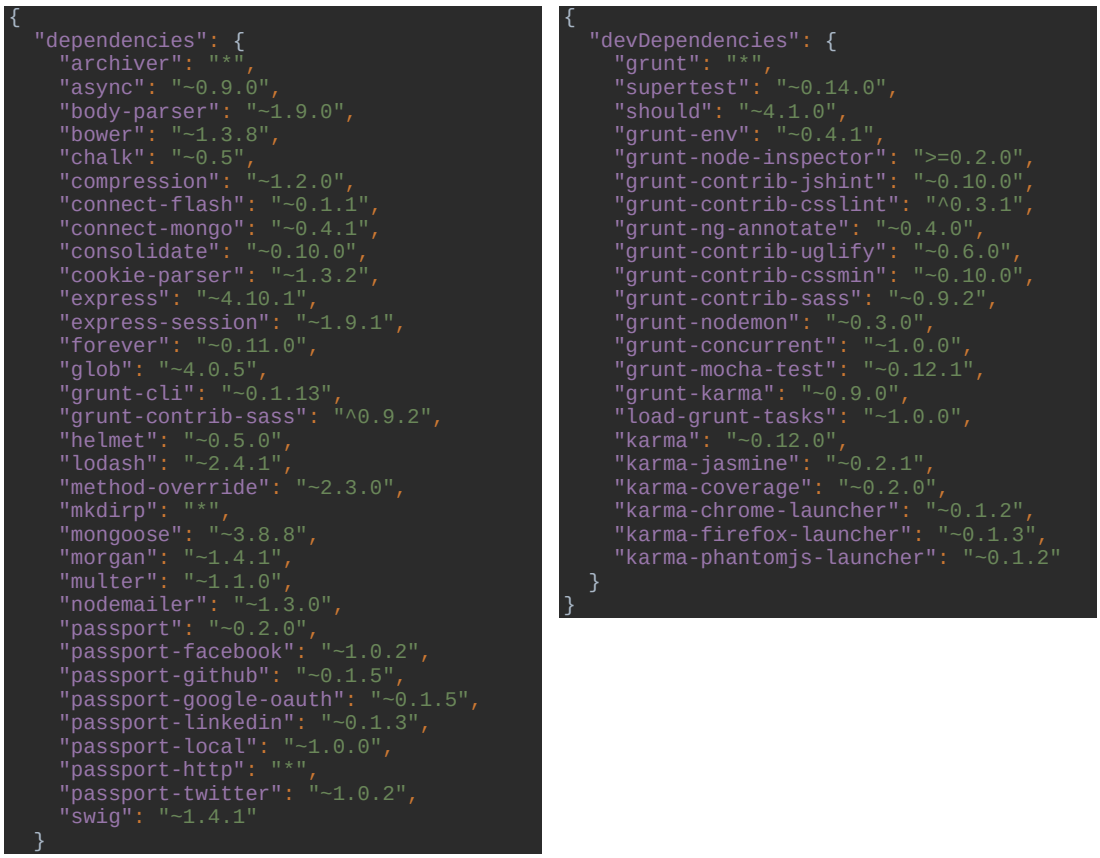
After deciding to use NodeJS, we had to select a database that stores JSON documents. We studied the differences between ElasticSearch and MongoDB and decided to use MongoDB. It is considered the world leading non-relational document oriented database and it integrates very easily with NodeJS through different packages that act as client-libraries to connect with the database, perform queries, insertions and update JSON documents easily. In our project we have used Mongoose. Mongoose provides an additional layer of features

```
{
    firstName: {
        type: String,
        trim: true,
        default: '',
        validate: [validateLocalStrategyProperty, 'Please fill in your first name']},
    lastName: {
        type: String,
        trim: true,
        default: '',
        validate: [validateLocalStrategyProperty, 'Please fill in your last name']},
    displayName: {type: String}
    email: {
        type: String,
        trim: true,
        default: '',
        validate: [validateLocalStrategyProperty, 'Please fill in your email'],
        match: [/.+\@.+\..+/, 'Please fill a valid email address']},
    username: {
        type: String,
        unique: 'Username already in use!',
        required: 'Please fill in a username',
        trim: true},
    password: {
        type: String,
        default: '',
        validate: [validateLocalStrategyPassword, 'Password should be longer']},
    salt: {type: String},
    provider: {
        type: String,
        required: 'Provider is required'},
    providerData: {},
    additionalProvidersData: {},
    role: {
        type: String,
        default: 'user'},
    locked: {
        type: Boolean,
        default: false},
    updated: {
        type: Date},
    created: {
        type: Date,
        default: Date.now},
    /* For reset password */
    resetPasswordToken: {type: String},
    resetPasswordExpires: {type: Date}
}
```

Figure 15. JavaScript definition of the UserSchema for Mongoose

on top of MongoDB. Mongoose requires the user to define a schema used to model the application data. Mongoose includes built-in typecasting, validation, query building, business logic hooks and more. In Figure 15 we can observe a simplified version of our Mongoose schema definition for the user accounts of our system.

To speed up the development and deployment process of the backend service and web editor we use GruntJS, a task management library to automate the launched tasks when starting up the server. In order to download the required Grunt dependencies we specify them in our *package.json* file. In the file *package.json* the dependencies that are going to be downloaded from the Node Package Manager (NPM) are defined. The Figure 16 example displays the dependencies of our system and an example of the *package.json* file. We use several grunt dependencies such as grunt-nodaemon (watches files in a directory in which nodaemon was started and if any files change, nodaemos will automatically restart the NodeJS server) or grunt-karma (sets up the configuration utilities to test the frontend web editor and run different types of tests).



```
{
  "dependencies": {
    "archiver": "*",
    "async": "~0.9.0",
    "body-parser": "~1.9.0",
    "bower": "~1.3.8",
    "chalk": "~0.5",
    "compression": "~1.2.0",
    "connect-flash": "~0.1.1",
    "connect-mongo": "~0.4.1",
    "consolidate": "~0.10.0",
    "cookie-parser": "~1.3.2",
    "express": "~4.10.1",
    "express-session": "~1.9.1",
    "forever": "~0.11.0",
    "glob": "~4.0.5",
    "grunt-cli": "~0.1.13",
    "grunt-contrib-sass": "^0.9.2",
    "helmet": "~0.5.0",
    "lodash": "~2.4.1",
    "method-override": "~2.3.0",
    "mkdirp": "*",
    "mongoose": "~3.8.8",
    "morgan": "~1.4.1",
    "multer": "~1.1.0",
    "nodemailer": "~1.3.0",
    "passport": "~0.2.0",
    "passport-facebook": "~1.0.2",
    "passport-github": "~0.1.5",
    "passport-google-oauth": "~0.1.5",
    "passport-linkedin": "~0.1.3",
    "passport-local": "~1.0.0",
    "passport-http": "*",
    "passport-twitter": "~1.0.2",
    "swig": "~1.4.1"
  }
```

```
{
  "devDependencies": {
    "grunt": "*",
    "supertest": "~0.14.0",
    "should": "~4.1.0",
    "grunt-env": "~0.4.1",
    "grunt-node-inspector": ">=0.2.0",
    "grunt-contrib-jshint": "~0.10.0",
    "grunt-contrib-csslint": "^0.3.1",
    "grunt-ng-annotate": "~0.4.0",
    "grunt-contrib-uglify": "~0.6.0",
    "grunt-contrib-cssmin": "~0.10.0",
    "grunt-contrib-sass": "~0.9.2",
    "grunt-nodemon": "~0.3.0",
    "grunt-concurrent": "~1.0.0",
    "grunt-mocha-test": "~0.12.1",
    "grunt-karma": "~0.9.0",
    "load-grunt-tasks": "~1.0.0",
    "karma": "~0.12.0",
    "karma-jasmine": "~0.2.1",
    "karma-coverage": "~0.2.0",
    "karma-chrome-launcher": "~0.1.2",
    "karma-firefox-launcher": "~0.1.3",
    "karma-phantomjs-launcher": "~0.1.2"
  }
}
```

Figure 16. Backend and frontend *package.json* file

To ensure the JavaScript code quality we use JSHint, a tool that helps to detect errors and potential problems in the JavaScript code. The utility is flexible and can be adjusted to particular coding guidelines that are defined in a configuration file (see Figure 17).

```
{
   "node": true, // Enable globals available when code is running inside of the NodeJS runtime
environment.
   "browser": true, // Standard browser globals e.g. `window`, `document`.
   "esnext": true, // Allow ES.next specific features such as `const` and `let`.
   "bitwise": false, // Prohibit bitwise operators (&, |, ^, etc.).
   "camelcase": false, // Permit only camelcase for `var` and `object indexes`.
   "curly": false, // Require {} for every new block or scope.
   "eqeqeq": true, // Require triple equals i.e. `===`.
   "immed": true, // Require immediate invocations to be wrapped in parens e.g. `( function(){}()
);`
   "latedef": true, // Prohibit variable use before definition.
   "newcap": true, // Require capitalization of all constructor functions e.g. `new F()`.
   "noarg": true, // Prohibit use of `arguments.caller` and `arguments.callee`.
   "quotmark": "single", // Define quotes to string values.
   "regexp": true, // Prohibit `.` and `[^...]` in regular expressions.
   "undef": true, // Require all non-global variables be declared before they are used.
   "unused": false, // Warn unused variables.
   "strict": true, // Require `use strict` pragma in every file.
   "trailing": true, // Prohibit trailing whitespaces.
   "smarttabs": false, // Suppresses warnings about mixed tabs and spaces
   "globals": { // Globals variables.
      "jasmine": true,
      "angular": true,
      "ApplicationConfiguration": true,
       "$": false
   },
   "predef": [ // Extra globals.
      "define",
      "require",
      "exports",
      "module",
      "describe",
      "before",
      "beforeEach",
      "after",
      "afterEach",
      "it",
      "inject",
      "expect"
   ],
   "indent": 4, // Specify indentation spacing
   "devel": true, // Allow development statements e.g. `console.log();`.
   "noempty": true // Prohibit use of empty blocks.
}
```

Figure 17. JSHint configuration file

A similar utility was used to ensure the code quality of the CSS files, CSS Lint. CSS Lint also points out problems with the CSS code. Does basic syntax checking as well as applying a set of rules to the code that check for problematic patterns or signs of inefficacy. The rules used can be seen in Figure 18 and they are completely configurable.

```
{
  "adjoining-classes": false,
  "box-model": false,
  "box-sizing": false,
  "floats": false,
  "font-sizes": false,
  "important": false,
  "known-properties": false,
  "overqualified-elements": false,
  "qualified-headings": false,
  "regex-selectors": false,
  "unique-headings": false,
  "universal-selector": false,
  "unqualified-attributes": false
}
```

Figure 18. CSS Lint configuration file

The RESTful APIs used by the system is built using ExpressJS, which aims to increase the code maintainability and flexibility. ExpressJS also serves as the main engine of the web editor, connecting the backend and frontend views.

GitHub is used for the development of the backend and frontend (see Figure 19). GitHub is a version control system that facilitates the management of code repositories and can be easily integrated with other services to facilitate the building and deployment of the system. As of September of 2016 the GitHub repository of the backend web service and the frontend web editor has 38 commits and will continue to be improved in the future for other iterations.

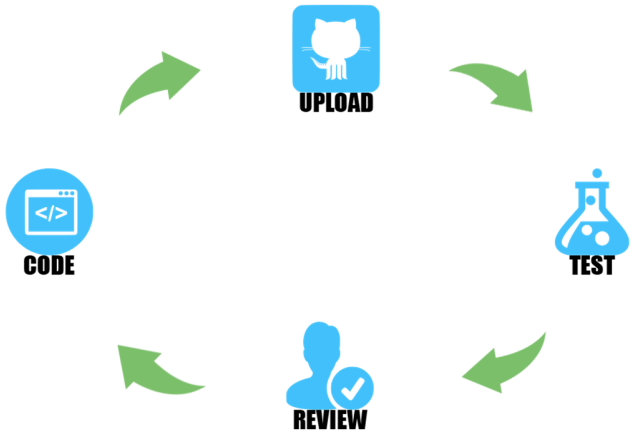



Figure 19. Development process

JSON is the format of the data model of the user accounts, challenges and courses. This data is consumed by the client application to build and display the gamified application. Since the format of the challenges and courses is JSON, it's easy to integrate a non-relational document-oriented database as a storage

backend. The JSON data is stored inside Mongo database. Mongo is a non-relational document oriented database designed to store JSON formatted documents. Mongo provides a series of functionalities such as scalability, stability and access control.

The web editor and backend service is packaged inside Docker images. This facilitates its integration and deployment in a server and reinforces the security of the data because the services are encapsulated inside Docker containers that are only accessible from within a defined virtual network. This solution facilitates the control of access points to the system increasing the security of the data. Docker is growing at a very accelerated pace and offers an automatic building, integration and management service connected directly to the GitHub repository, improving the speed of the process from the development of the system all the way to the deployment (see Figure 20). When changes are detected on the source code of the GitHub repository, either due to a commit or a pull request that has been merged, a new build process gets started on Docker to create a new version of the Docker image containing the changes.

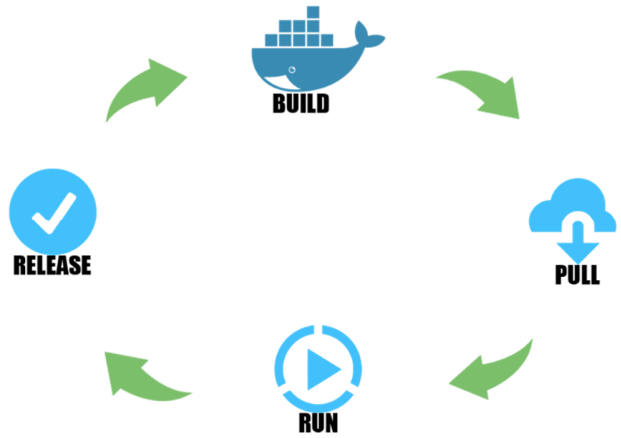



Figure 20. Web Editor build process

As a development IDE, we have used WebStorm by JetBrains. It provides intelligent coding assistance for JavaScript and compiled-to-JavaScript languages (NodeJS, HTML and CSS). It also provides a powerful code completion mechanism, navigation features, on-the-fly error detection, integration with JSHint and refactorings for all of these languages. It also provides advanced coding assistance for Angular (the technology used for the frontend web editor development) and it integrates with Karma test runner. WebStorm can be obtained for free by logging in with a student license using the Complutense University of Madrid email.

## 5.3 Frontend

The frontend web editor was designed to be easy to use by the medical personnel. In the frontend the medical personnel, domain expert or even a player can sign in with a user account and create content for the application creating different types of challenges for multiple courses.

The development technology is AngularJS. By using AngularJS we are required to implement a Model View Controller (MVC) design pattern. Furthermore, AngularJS is defined inside the HTML to determine the execution of the user interface. Data models for AngularJS are plain JavaScript objects and all properties found on the '$scope' object are automatically bound to the view by Angular. Meaning, AngularJS watches for changes to these properties and updates the view automatically.

To manage the frontend web editor dependencies we use Bower. Bower offers a generic solution to the problem of front-end package management, while exposing the package dependency model via an API that can be consumed by a more opinionated build stack. There are no system wide dependencies, no dependencies are shared between different apps, and the dependency tree is flat. Bower dependencies are defined inside the *bower.json* file (see Figure 21).

```
{
  "name": "Cytopathology Editor & Backend Service",
  "version": "0.0.1",
  "description": "Full-Stack JavaScript with MongoDB, Express,
AngularJS, and Node.js",
  "dependencies": {
    "angular": "~1.3",
    "angular-resource": "~1.3",
    "angular-mocks": "~1.3",
    "angular-cookies": "~1.3",
    "angular-animate": "~1.3",
    "angular-touch": "~1.3",
    "angular-sanitize": "~1.3",
    "angular-ui-utils": "~0.1.1",
    "angular-ui-router": "~0.2.11",
    "angular-material": "~0.10.0",
    "lf-ng-md-file-input": "*"
  },
  "resolutions": {
    "angular": "~1.3"
  }
}
```

Figure 21. *Bower.json* configuration file

The user interface is designed using a Google Material design style and guidelines. Material Design is composed of different layers and follows Google's conceptual design philosophy that outlines how apps should look and work on mobile devices. It breaks down everything (such as animation, style, layout, etc.) and gives

guidance on patterns, components and usability. According to Google: "We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design."

Material Design has borrowed plenty of design concepts from the flat aesthetic and other trendy techniques. In fact, some would argue that Material Design is a close cousin to Flat Design because many of the visual treatments are quite similar.

What separates layered interfaces from totally flat design is that effects are used to create more three-dimensional spaces and to mimic lighting. In essence, designers are bringing back some of the design tricks eliminated with flat design. The difference is that they're using these tricks to improve usability rather than simply as decorative accents.

Google provides a vast number of colour options. While many designers opted reds when it comes to designing flat, more layered interfaces seem to feature blues, deep purples and yellows. That's likely because each of these hues is easy to pair with contrasting white for the background or the text.

The color palette for the Cytopathology Web Editor is mainly blue and light-blue and the implementation of the client-side views logic is done using the Google AngularJS engine facilitating the integration of the model-view-controller design pattern.

## 5.4 Conclusions

In the previous chapter we had identified the development technology used for our educational application (libGDX), after doing an exhaustive study about different alternatives and studying in detail the two selected options Unity 3D and libGDX.

In this chapter we explain how we designed and implemented an integrated system that satisfies the initial project goals and constraints. We decided to split the integrated system in three parts, each one of them explained in detail in this chapter, in terms of design and implementation technologies, and from the point of the final user, in the next chapter. The three parts that compose the integrated system are:

1. An educational application, called Cytopathology Challenge, which runs on low-cost Android devices, HTML5 web browsers and doesn't require an internet connection to run. Developed using libGDX over Java programming language. The previously identified gameplay mechanics are implemented in the Cytopathology Application, sharing similar mechanics as the previously analyzed applications that made use of medical images (MalariaStop, Cell Slider, BioGames and VMATs). The application starts with an introductory story to set up the context for the player and the initial motivation and the player has to successfully complete Challenges

(levels) of different types. A Scoring system is associated to each Challenge to create leaderboards.

2. A backend service that exposes a RESTful API. In order to manage Challenges, Courses and user accounts (authorization and authentication to protect the resources of the system that are stores in a non-relational document-oriented Mongo database). The backend service is designed to facilitate the extension of the system through other clients and for other platforms in a standardized manner, communicating with a RESTful service. It's developed using NodeJS backend technology, Node Package Manager for the backend dependencies and Grunt automated task manager. Its code is available at GitHub and integrated with DockerHub, a continuous integration engine that builds the deployable service when changes are detected in the source code.
3. A Cytopathology Web Editor designed to be easy to use by the medical personnel and to be able to manage the system content without depending on the author of this work or any developer at all. It's also developed using NodeJS frontend technology, with AngularJS and following the Google material Design style guide. For the frontend dependencies management we use Bower.

After describing the reasons behind our integrated system design and the implementation technologies, the next chapter explains the system from the point of view of the final user. Explaining how a player uses the educational application (Cytopathology Challenge), how a developer can extend the system functionality consuming the backend service and how the medical personnel manages the content of the system through the Cytopathology Web Editor.

# 6 Description of the Integrated System

The solution proposed for our project consists of an integrated System designed to be easy to use by the medical personnel and to facilitate the content management of the system. The backend's RESTful API can be used by a developer to extend the functionality of the system with other clients and even to other platforms. The Cytopathology Application is designed to be played by final stakeholders interested in learning about cytopathology. Finally, the web editor can be used by the domain experts (medical personnel) to manage (create, delete or edit) the content of the system. The game-like application can be automatically produced from this content without requiring any other coding.

## 6.1 Cytopathology Challenge: Game-like Application

This section describes the game-like application that runs on low-cost Android devices and HTML 5 web browsers. The name of the application is *Cytopathology Challenge* as a way to introduce the learner with the gameplay dynamic, which consists of solving challenges. The default language is English with the possibility to translate the texts to other languages.

The application is composed of different screens available for the player to explore. In the next sections the most important screens from the application are described.

### *6.1.1 Laboratory*

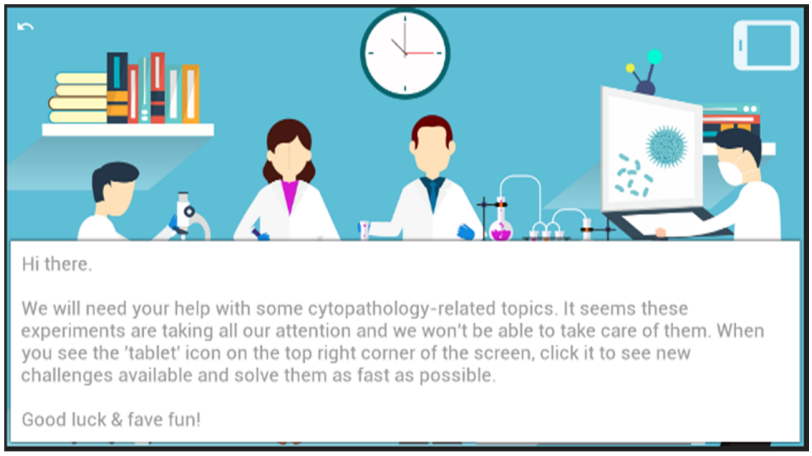



Figure 22. Laboratory screen displaying an introductory story

This is where an introductory story is displayed to the learner. This aims to introduce the player with the initial context of the application. It also has a motivational purpose, the current obtained score will be

displayed along with other gamification information (e.g. leaderboards). Pressing the *tablet* button (top-right corner) displays the *Courses List* screen.

### *6.1.2 Courses List*

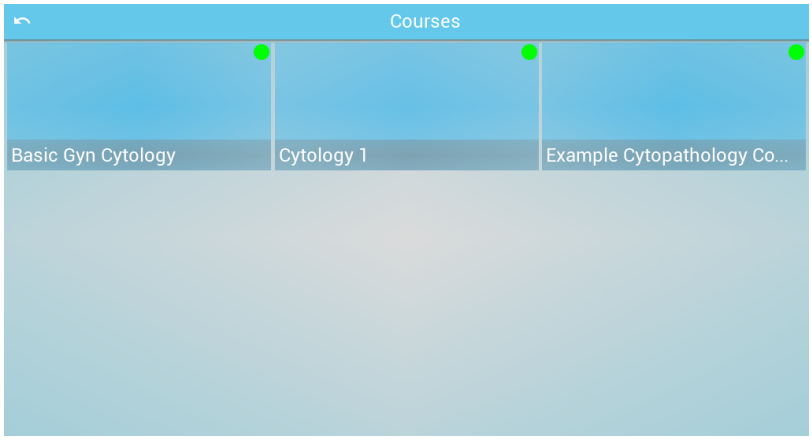



Figure 23. Display of the courses of the application

The Figure 23 displays all the courses available to the student. The courses are composed of a group of challenges that follow the same topic. A course is used to have exams about a topic. An exam is a mechanic where the student is prompted to solve multiple challenges from the same course as a sequence. After completing an exam the student receives a grade.

### 6.1.3 Challenges List

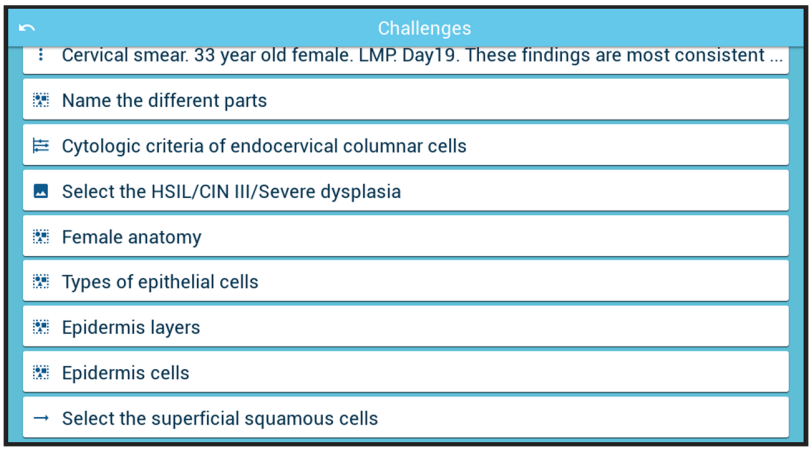



Figure 24. Multiple challenges that the learner must solve

The Figure 24 displays a list of challenges that should be solved by the learner. Once a challenge is solved a score is obtained depending on how well the challenge has been solved. Currently there are five different types of challenges described below.

#### 6.1.3.1 Multiple Answer Question


Figure 25. 'Multiple Answer Question' user interface

Figure 25 shows text question with multiple answers is displayed on the right and a related slide on the left. The slide can be explored using an interaction similar to *Google Maps* (drag, zoom in/out, etc) to give a better understanding of its content. The order of the multiple answers can be randomized to increase the replayability of the challenge.

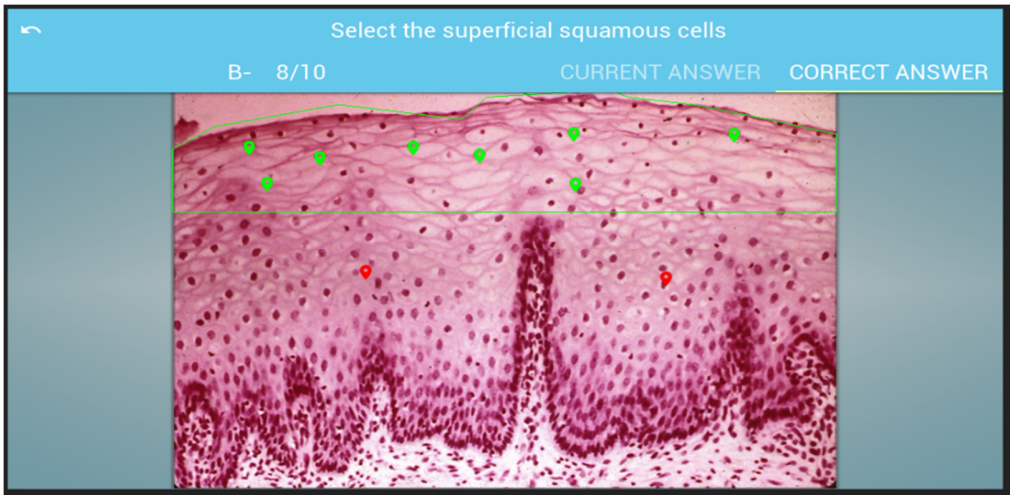



Figure 26. Result screen displaying the obtained score and the correct answer. In this challenge the player has to target parts of the slide that are contained inside different areas (polygons)

The *check* action (bottom-right corner) will display a result screen with the obtained score, the current answer and the correct answer (see Figure 26).The result screen gives feedback to the student about the current challenge. The result screen displays the obtained score, computed based on the current answer, following the academic grading in the United States. The student can also see the correct answer to the challenge as well as the current answer, and compare the differences between both. The result screen has been designed to help the student find the mistakes committed in the challenge.

#### *6.1.3.2 Multiple Answer Question with Images*

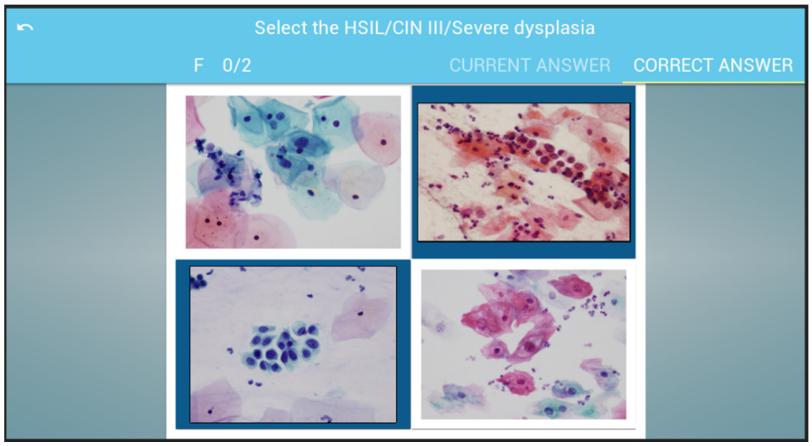



Figure 27. Result screen of a 'Multiple Answer Question with Images' challenge

Figure 27 shows a challenge dynamic similar to the previous one (*Multiple Answer Question*). Answer a question choosing from four different image-based choices instead of text options (labels).

### *6.1.3.3 Drag and Drop*

In Figure 28 the learner must drag objects from the right part of the screen and drop them on the correct blank option of the slide.

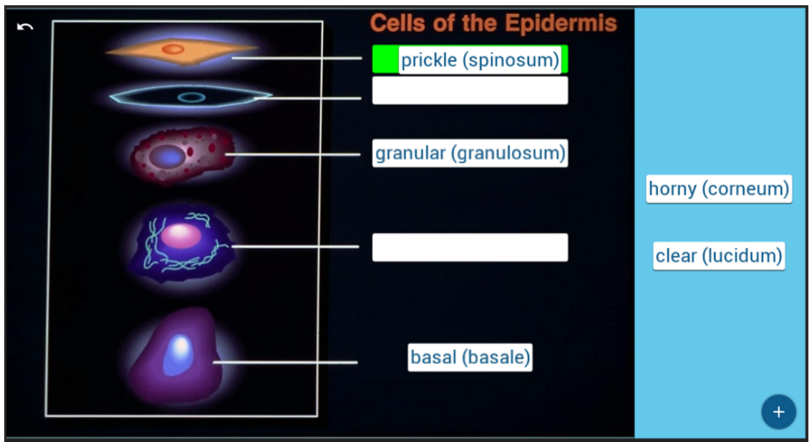



Figure 28. 'Drag and Drop' user interface

### *6.1.3.4 Fill the Options*

This challenge shown in Figure 29 displays a text area with different label-options. A label-option is composed of different tags from which the correct one must be chosen to complete the challenge.

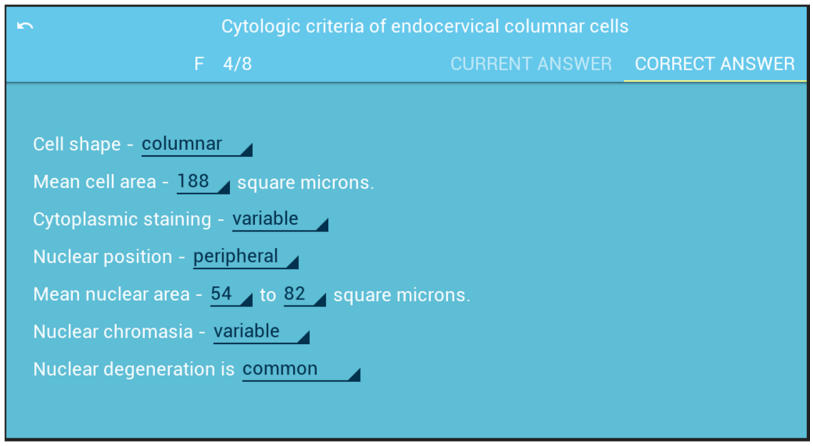



Figure 29. Result screen of a 'Fill the Options' challenge

### *6.1.3.5 Select an Area from the Slide*

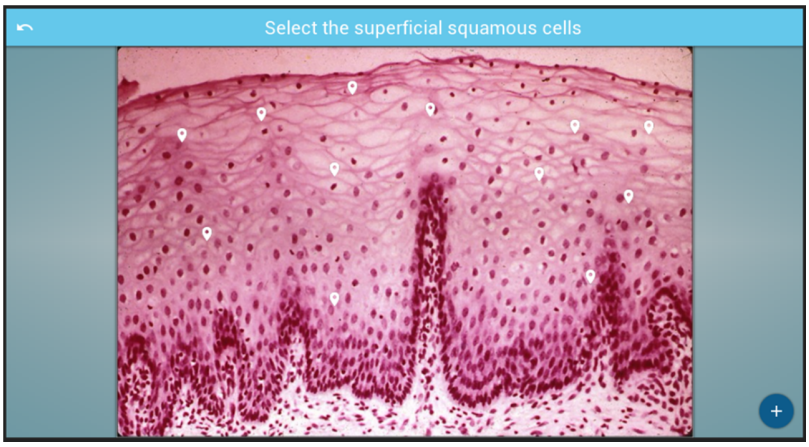



Figure 30. 'Select an Area from the Slide' user interface

This challenge asks the learner to mark a certain area from the slide (see Figure 30). With a click, a marker is placed. A mark can be removed by clicking it. The result is computed depending on the amount of markers placed correctly (within the correct predefined polygon area from the slide) and incorrectly.

Note that all the slides can be explored using an interaction similar to *Google Maps* (drag, zoom in/out, etc) to give a better understanding of its content. The *fit* button (bottom-left part of the slide) automatically fits the slide to the bounds of the outside container (see Figure 31).


Figure 31. Demonstration of exploring a slide (zoom in and drag movements)

### 6.1.4 Gamification

Other interesting features are the gamification layers of the application. There are some features identified when we analyzed similar applications in previous studies.

We decided to improve the feedback the players receive by displaying customizable information about each challenge solved by the student. The content creator (e.g. teacher) can set up the feedback information shown to the students. This process is achieved configuring a Challenge Explanation view, explained with more details in the following sub-chapter. Each challenge has an associated difficulty mechanism that is used to establish the initial time available for the student to complete the challenge (timer). Another gamification layer is to improve the difficulty mechanism so that some challenges are unlocked only when certain amount of challenges of lower difficulty are completed. Moreover, a timer based mechanism increases the engagement of the players. Each challenge the student has to solve must be solved before a timer finishes. The initial time of the timer changes depending on the difficulty of each challenge so that a challenge with higher difficulty has a lower amount of time available to be solved compared with another challenge of lower difficulty. An interesting gamification layer is the students ranking based on their overall score and the amount of challenges solved. Finally, an introductory tutorial guides the player with the game mechanics and sets up the initial story with the purpose to establish a base context for the student to start. Other applications such as *Angry Birds* have a rating system that gives the players stars (up to three) based on how well they have solved the challenge. We have decided to use a score-based system where each challenge receives a score between 0 and 100. The final challenge score can be considered as a grade. We displayed the grades to the student following the academic grading system in the United States. Another gamified application is *Duolingo* that rewards the player after each 'mini-challenge' and has a badge system, so that every time a player achieves a certain goal, he receives a reward competing against the friends.

## 6.2 Cytopathology Challenge Web Editor

The HTML 5 Cytopathology Challenge editor is a web platform designed to facilitate the creation of challenges that can be integrated within the game-like application. The user interface facilitates the creation and management of challenges in a collaborative manner. This web editor is composed of an HTML 5 user interface (frontend) that communicates with a service (backend) through a RESTful application programming interface.

### 6.2.1 Backend

The backend service exposes a RESTful API that stores the underlying data model and files (images) attached to each challenge. The data model is stored using an extensible JSON format that facilitates the

integration with the Android and HTML 5 game clients, and other possible clients in the future, improving the integration across different platforms (Android, Windows, Mac, Linux, HTML 5, etc.).

#### *6.2.1.1 Users Authentication and Authorization*

The first, and most basic service, is the user account management service. Exposes a series of operations that allows the creation, modification, deletion and administration of user accounts. Those services are used to sign-up new users, modify or delete user accounts, or event define a roles-based authorization to facilitate the administration of the system. Note that in the first versions of the framework, there is no authorization protection implemented for the resources of the system. This is, indeed, a design decision in order to develop a first prototype and receive feedback as soon as possible, without the need to invest time in administration procedures that might be tedious and non-relevant for the goal of the system, which is to facilitate the creation of cytopathology-related challenges and integrate them in a gamified multi-platform application. In the final version of the service the authorization has been fully integrated in order to protect the resources of the system. In order to access or modify the system's resources (accounts, courses and challenges) the proper authorization must be given based on the roles of the account that is trying to access the resource.

There is also support for single-sign-on functionalities. The sign up of user accounts through already created social-media accounts. This functionality is fully implemented, but it's blocked in the initial release of the platform. In order to sign in, a new account must be created from scratch in the initial release of the web editor since this release is intended for testing purposes. In the final version of the web editor the use of social-media accounts to sign in the system has been unlocked and is fully functioning. Table 7 displays the RESTful API for an external developer that wants to extend the system and create new user accounts.

Table 7. Most important user API requests useful for a developer

| Method | Route | Description |
|---|---|---|
| GET | /users/me | Returns the current user |
| PUT | /users | Update the profile |
| DELETE | /users/accounts | Remove OAuth provider |
| POST | /users/password | Changes the user's password |
| POST | /auth/forgot | Initiate the 'Forgot password' |
| GET | /auth/reset/:token | Validate the reset token |

| | | |
|---|---|---|
| POST | /auth/reset/:token | Reset the password |
| POST | /auth/signup | Signup an account |
| POST | /auth/signin | Sign in an account |
| GET | /auth/signout | Sign out a signed in account |
| GET | /auth/facebook | Authenticate using Facebook |
| GET | /auth/twitter | Authenticate using Twitter |
| GET | /auth/google | Authenticate using Google+ |
| GET | /auth/linkedin | Authenticate using LinkedIn |
| GET | /auth/github | Authenticate using GitHub |

#### *6.2.1.2 Challenges Management*

The main service is the challenges management service. Exposes a RESTful API for the creation, modification and deletion of challenges. Stores the underlying data model of each challenge using a JSON format inside a Mongo database as well as the images attached to each challenge. There are different types of challenges with different data model structures. The challenges share some commonalities in their structure such as their name, description, and the course they belong to.

Table 8 displays the most important API requests to manage challenges from an external client.

Table 8. Challenges API requests

| Method | Route | Description |
|---|---|---|
| POST | /upload/:challengeId | Upload a single image for a challenge |
| POST | /uploads/:challengeId | Upload four images for a challenge |
| POST | /hints/:challengeId | Upload images for a challenge hint (explanation) |
| GET | /challenges | List all the challenges |
| POST | /challenges | Create a new challenge |
| GET | /challenges/:challengeId | Return a single challenge |

| | | |
|---|---|---|
| PUT | /challenges/:challengeId | Modify the content of a given challenge |
| DELETE | /challenges/:challengeId | Delete a given challenge |

#### *6.2.1.3 Courses Management*

Each challenge belongs to a given course. A course is composed of a name and a group of challenges that defines its content. Courses are used to group challenges under a similar category and facilitate the navigation through the challenges database. A RESTful API is also exposed for the management (creation, modification and deletion) of courses and their challenges. Table 9 displays the courses management API.

Table 9. Courses API requests

| Method | Route | Description |
|---|---|---|
| GET | /courses | List all the courses |
| POST | /courses | Create a new course |
| GET | /courses/:courseId | Return a single course |
| PUT | /courses/:courseId | Modify the content of a given course |
| DELETE | /courses/:courseId | Delete a given course |

### *6.2.2 Web Editor User Guide*

The frontend is composed of a user interface designed to create and manage user accounts, challenges and courses. The user Interface is composed by the main toolbar, situated in the top part of the screen that exposes two different types of actions: managing user accounts and creating challenges.

### 6.2.2.1 *Sign in/out*

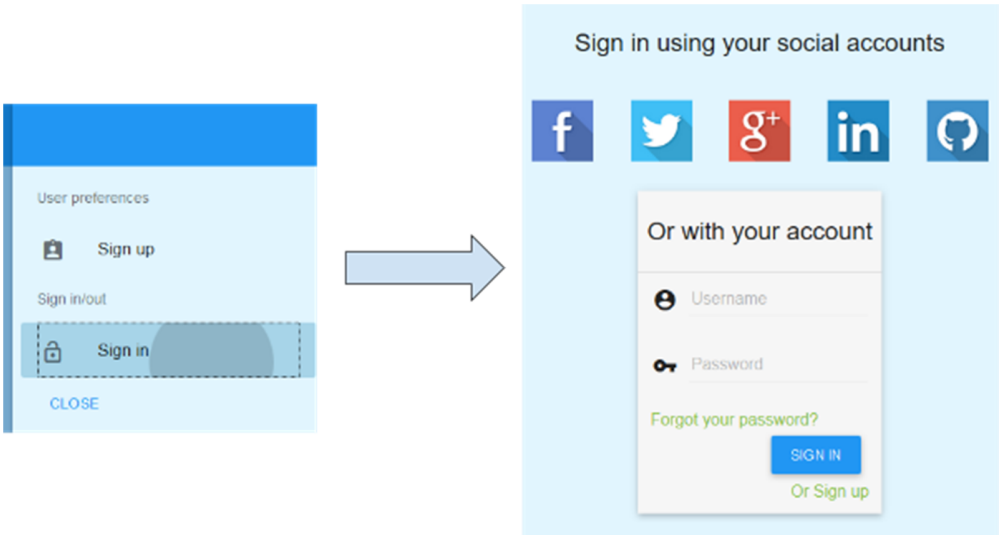



Figure 32. Sign in user interface

As seen in Figure 32, the right side of the toolbar (top right corner of the screen) displays the user account management interface. This is useful in order to sign in the system with an already created account, create a new account from scratch (sign up) or sign in the system using a social network account (Facebook, Google, Twitter, Linkedin, etc.).

### 6.2.2.2 *User Account Management*

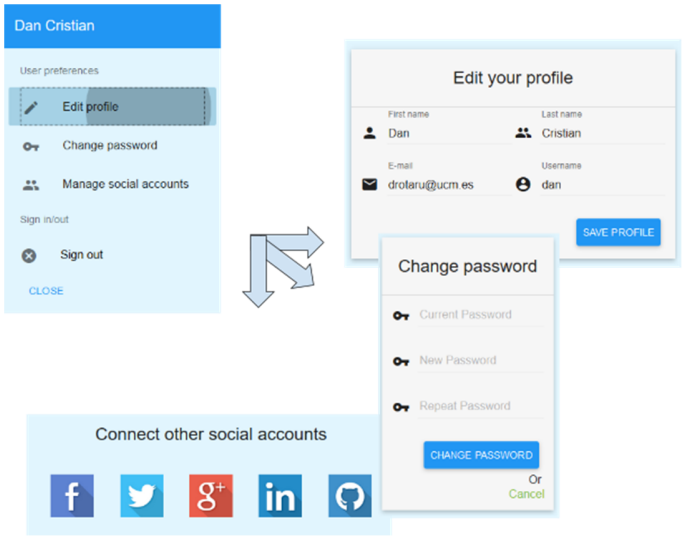



Figure 33. Account edition user interface

After creating a user account and singing in the system, a new set of operations are displayed on the top right corner of the interface: managing the user profile, changing the account password or connecting with other social networks (see Figure 33). Note that in the initial release of the web editor this functionality was blocked for testing purposes but in the final release this functionality is fully working.

#### *6.2.2.3 Course and Challenge Management*

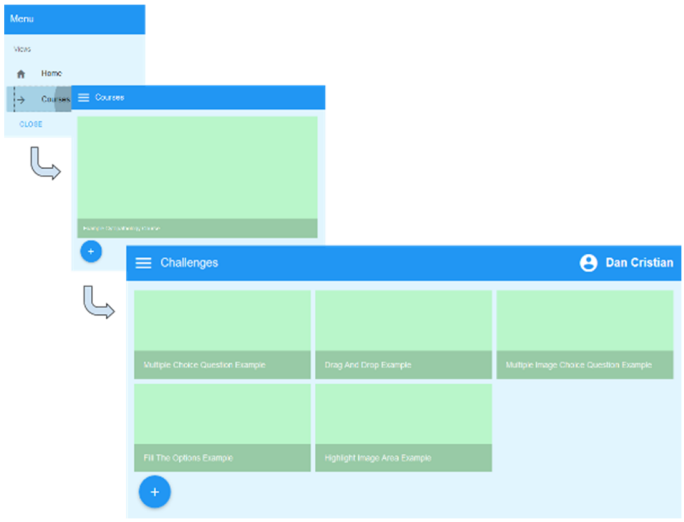



Figure 34. Challenges management user interface

The top left menu displays operations to navigate to the current courses, create new courses and to navigate the challenges of a specific course as well as to create new challenges for that specific course (see Figure 34).

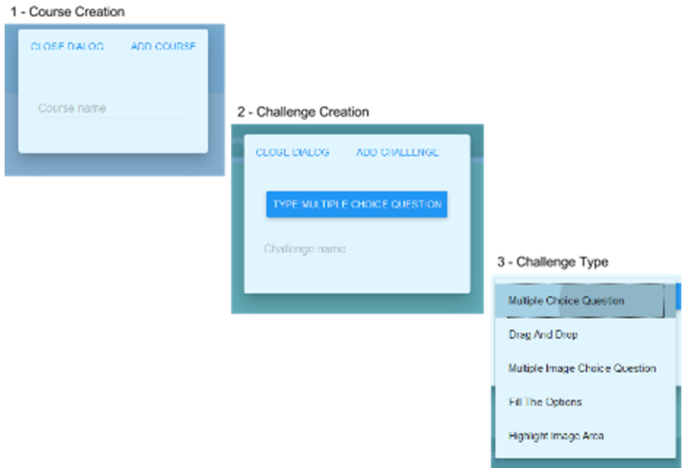



Figure 35. Challenges creation user interface

For the creation of a course the name is required. To create a challenge the name and its type (Multiple Choice Question, Drag and Drop, Multiple Image Choice Question, Fill the Options, Highlight Image Area) must be specified (see Figure 35). Note that in the initial release of the web editor, an example course is available as well as an example challenge for each type previously mentioned.

#### *6.2.2.4 Challenges edition*

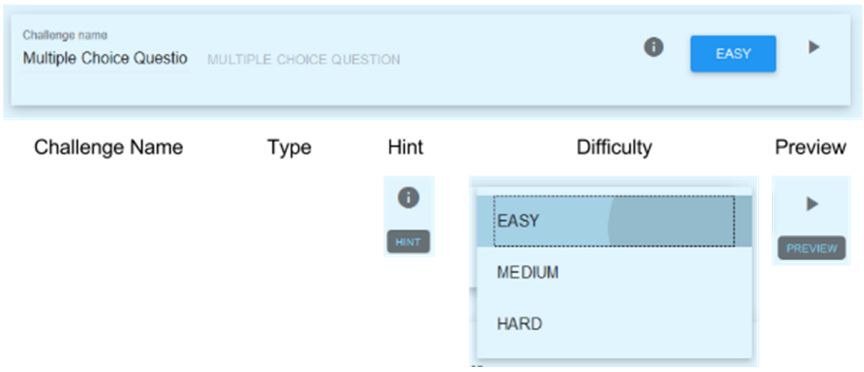



Figure 36. Common challenge attributes

The different challenge types share a common set of attributes: name, difficulty and type. The hint and preview are common operations for each challenge, as shown in Figure 36.

### *6.2.2.5 Challenge Explanation*

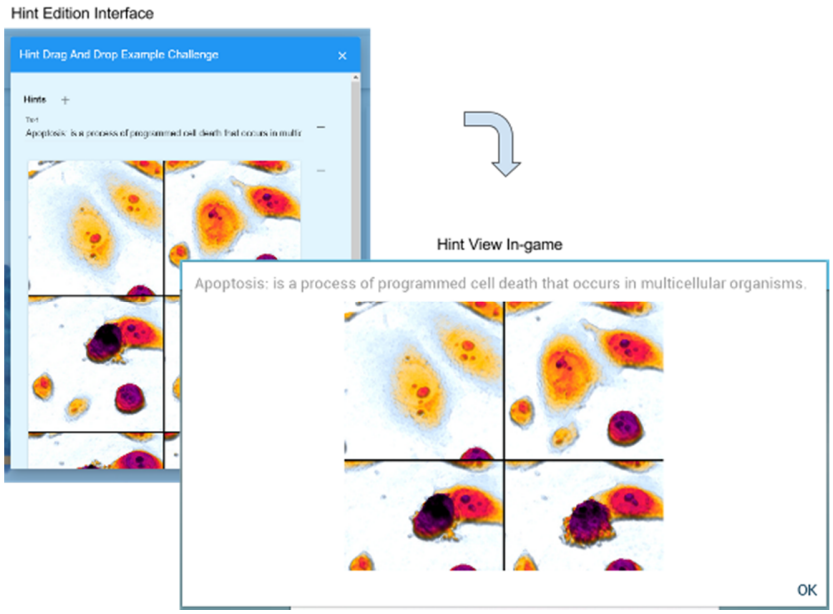



Figure 37. Hints user interface, web editor (left) and application (right)

Hints are displayed to the student once a challenge has been completed as a means to give feedback to the student and learn more information about the specific answer. This greatly increases the feedback a student receives after completing a challenge. Hints are composed of a list of either text paragraphs or images displayed from top to down in a vertical layout (see Figure 37).

### *6.2.2.6 Preview*


Figure 38. Preview a Drag and Drop challenge user interface

The preview functionality allows the creator to play the challenge from the point of view of a student and it also shows the results screen as well as the configured hints (see Figure 38). The preview functionality shows how a challenge is viewed exactly inside the game application. Facilitates the edition by giving an exact view of the challenge that is being created inside the game. This view displays to the teacher the student's experience when playing the challenge. There are five different types of challenges, with different mechanics and structures.

#### *6.2.2.7 Multiple Answer Question Challenge*

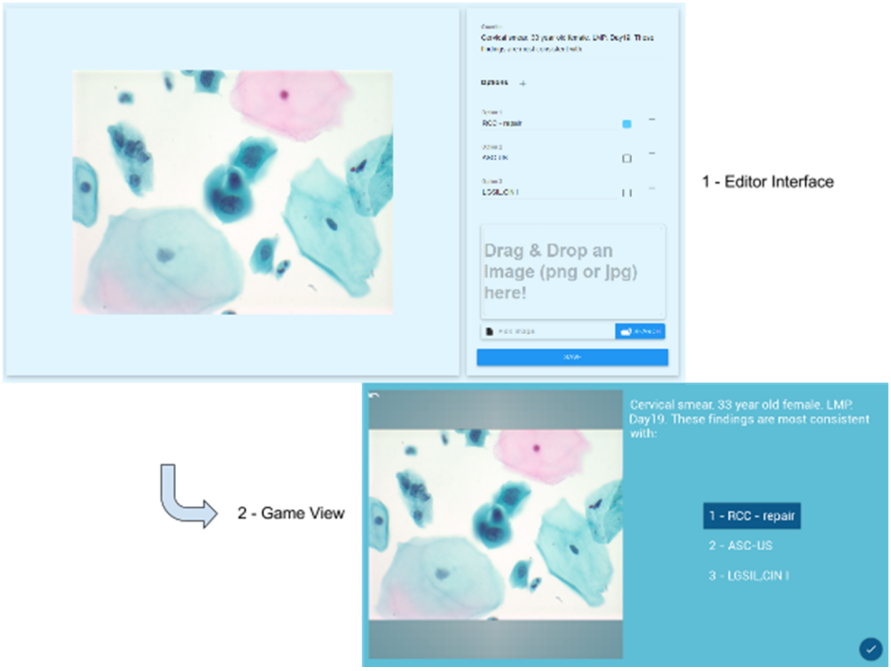



Figure 39. Multiple Choice Question user interface, web editor (left) and game-like application (right)

The Multiple Choice Question Challenge editor seen in Figure 39 displays the user interface to create this type of challenge: define a question with multiple text answers with only one possible correct answer and a related slide in the left side of the screen.

### 6.2.2.8 *Multiple Image Choice Question Challenge*

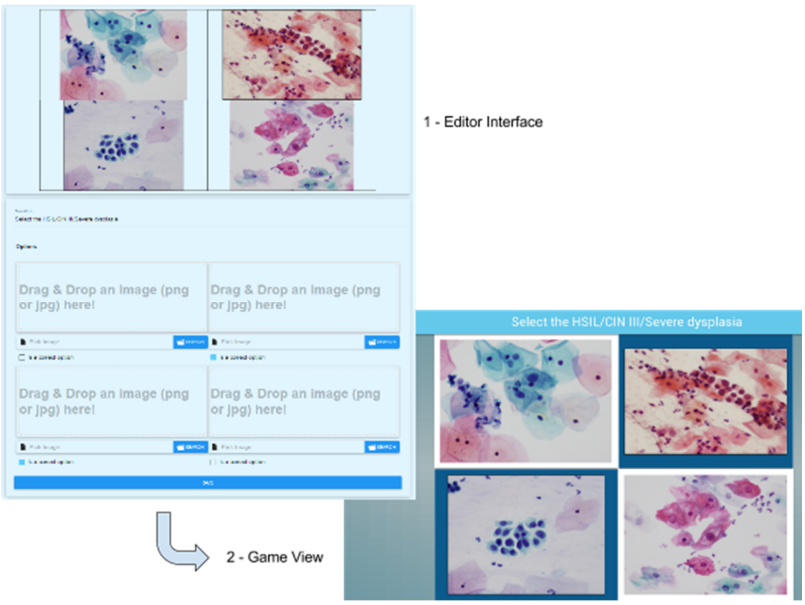



Figure 40. Multiple Image Choice Question user interface, web editor (left) and game-like application (right)

The Multiple Image Choice Question Challenge editor displays the user interface to define a question and four different image answers from which up to four answers can be correct (see Figure 40).

### 6.2.2.9 *Fill the Options Challenge*


Figure 41. Fill the Options user interface, web editor (left) and game-like application (right)

The Fill the Options Challenge editor is designed to create text statements that are mixed with multiple options labels from which the learner must choose the correct one (see Figure 41).

#### *6.2.2.10 Highlight Image Area Challenge*


Figure 42. Highlight Image Area user interface, web editor (left) and game-like application (right)

The Highlight Image Area Challenge (see Figure 42) consists of different areas of interest defined as polygons over a slide. The learner is asked to select them by clicking on the slide and leaving 'marks'.

#### *6.2.2.11 Drag and Drop Challenge*

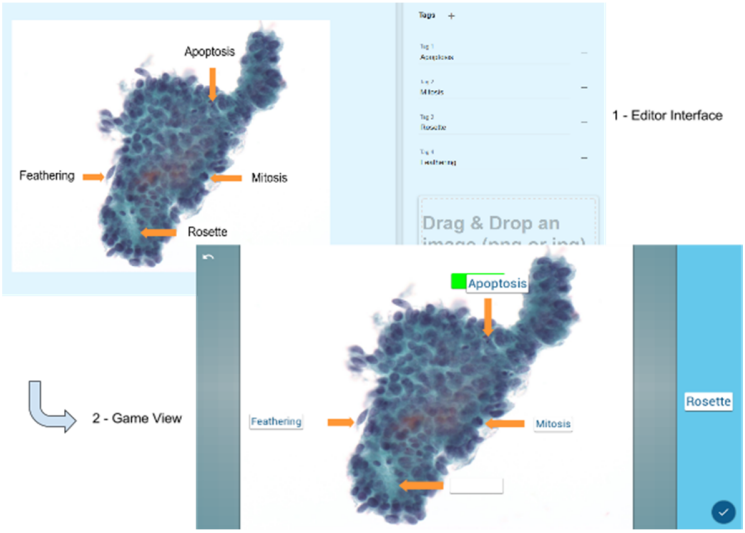



Figure 43. Drag and Drop user interface, web editor (left) and game-like application (right)

The Drag and Drop Challenge is composed of a set of labels that must be positioned in specific areas over a slide, as seen in Figure 43. It's designed to test the classification of specific parts of a slide. The learner must correctly sort the labels by placing them in their correct position, indicating that he has specific knowledge about the area of the slide pointed by the label.

## 6.3 Conclusions

In the previous chapter we presented the design and implementation aspects of the integrated system to be produced. In this chapter we describe the system from the point of view of students (players), medical personnel experts and developers.

We start describing the educational application (Cytopathology Challenge) and the different screens a player will encounter throughout the gameplay. We describe the menu screens, for instance Laboratory screen when the initial story is displayed to set up the context, or the Challenges List a screen where the player can select the different challenges of a given course and see additional information about each challenge (score or the amount of seconds it took to be completed). Afterwards, the different types of challenges, their gameplay mechanics and their user interface, are described. Challenge types such as Multiple Answer Question when the player must answer a question by choosing from a list of different answers or Drag and Drop where the player must drag concepts from a list of concepts and position them correctly in the image matching their meaning.

We describe the backend service RESTful API that manages the data-model of the whole system. This service that be extended even to other platforms or applications. This can be easily achieved, since the developer doesn't need to know how the system works underneath, only needs to know the specification of the RESTful API to create another client that communicates with the backend service in a standardized manner. We describe the three main data-models available in our system (1) a user account management service, (2) the challenges management service and (3) the courses management service.

Finally, the web editor user interface is described. Designed so that medical personnel can easily manage the content of the system without the dependency of a developer. The different parts of the user interface are described, from the process or signing up to the process of creating courses and challenges. The preview functionality is described, a view that allows the medical personnel to view the edited challenge from the point of view of the player, directly from the editor interface. The Challenge Explanation view is described, a view completely configurable to show the player information about a given challenge after completing it, giving valuable feedback about the challenge and the correct/incorrect answer.

At this point we have described from the point of view of the final user the integrated system that we designed and implemented as an educational platform to train cytologists in resource-limited areas and to be

tested at an Introduction to Cytopathology course at Harvard Medical School. In the next, and final chapter, we review how many of the initial goals we managed to satisfy and whether the initial project constraints have been successfully respected.

# 7 Conclusions and Future Work

## 7.1 Conclusions

Our initial goal was well defined, professors at Harvard medical School and medical personnel at Massachusetts general Hospital want an educational application to train cytologists in resource-limited areas of the world. Before deploying the educational application to resource-limited areas it is going to be tested with actual students at an introduction to Cytopathology course at Harvard medical School.

Our initial technical constraints were very challenging. The educational application had to run without a stable internet connection and on low cost Android devices, the devices used to deploy the system in resource-limited areas of the world where we cannot rely on a stable internet connection. At the same time, the educational application had to run on personal computers because it's going to be tested with actual students at Harvard Medical School.

We decided to conduct several studies before taking any design decision and starting the implementation. In order to identify the best design and development practices for serious games in the health domain, we have performed a systematic review of serious games in the medical domain. Furthermore, to identify the best gameplay mechanics suitable for our project, we have analyzed different games and game-like applications (MalariaSpot, Cell Slider, VMATs and BioGames) focused on medical training using medical imaging. After this first set of studies, we had identified the best gameplay mechanics for our project: having an introductory story to set up the context and motivation of the player, followed up by a tutorial explaining the educational application, then a set of levels for the player to complete are presented with continuous feedback about the content of the application so that the player knows how to improve its answers and, finally, some gamification features such as score tracking through leaderboards and timers for each level of the application.

After identifying what gameplay mechanics we wanted to implement, the next logical step was to decide which development technology is the most suitable for our project. We decided to analyze the most important development tools and game development frameworks available in the game industry sector, classifying them in four different classes, from more flexible and harder to learn (AAA tools) to more specialized, easier to learn but not so flexible in terms of what game we can develop with them (Specialized tools such as eAdventure). Finally ¡, we had to decide between two options, Unity 3D or libGDX, a Java game development framework. Both options were suitable for our use case in the majority of considerations, for instance both options support the target platforms of our project (HTML 5 and Android), have an active community of users publishing games, a well-documented source code, their license are open source, are free of charges and provide more

than enough features to develop our educational application. The big difference between them is that libGDX has an overall greater performance output, which, in our case is critical considering that our application must run on low-cost Android devices. For this reason we decided to use libGDX as our development technology, furthermore the author of this work had a lot of experience of the framework.

At this point we know (1) what gameplay mechanics we wanted to implement in our educational application and (2) which development technology we wanted to use. With this information, and taking in consideration the challenging constraints and the goals of the project we designed an integrated system composed of an educational application, a web editor for the content and a standardized RESTful service. Considering the initial goal of the project and the designed system, we can conclude that the following objectives have been successfully completed

- Goal 1 – We have designed an integrated system that complies with the initial constraints. The educational application, Cytopathology Challenge, runs on HTML 5 web browsers making it suitable for testing in personal computers, or even mobile devices using a web browser, with students at an Introduction to Cytopathology course at Harvard Medical School.
- Goal 2 –The educational application runs seamlessly on low-cost Android devices and does not require an internet connection to work which makes it deployable in resource-limited areas, where these devices are used and we cannot rely on a stable internet connection. Furthermore, the gameplay mechanics implemented have been proven to work on similar games or game-like applications based on medical imaging used to train medical personnel. More information about the gameplay mechanics implemented and how have been is available in Chapter 3 - Related Work and Applications.
- Goal 3 – We have developed a web editor designed to be easy to use by the medical personnel. The web editor can be used to manage (create, edit and delete) the content of the application. To be able to manage the content of the application we have defined two main data models used by the educational application, Courses and Challenges. A Course is a set of Challenges that share the same topic. Courses are used to group Challenges and to filter and query more easily challenges. A Challenge is a piece of gameplay designed with educational purposes and, at the same time, to be engaging enough for the players. Players must successfully complete Challenges to succeed in the educational application, and there are 5 different types of Challenges.
- Goal 4 –The integrated system is ready to be tested with students but the estimated date for the tests is set on February, 2017. Until then the integrated system will be under continuous

improvement revisions through external feedback with different users. We plan on iterating several times, in the meantime, doing tests with users and collecting feedback in order to improve the final user experience.

The project had a very limited budget, and some challenging initial constraints. After several studies we designed and implemented an integrating system that satisfies the initial goals, as we have previously concluded. The integrated system consists of

(1) a game-like application for HTML5 and low-cost Android devices,

(2) a RESTful backend service for the management of the content used by the application and

(3) a web-based editor designed to be easy to use, removing the need of developers, where domain experts can manage the content presented to their students.

## 7.2 Contributions

The main contributions of this project are:

- Literature review of serious games related with the e-health domain searching several databases (ACM, BioMed, PubMed, etc.) to classify relevant results similar to our use case in order to identify best practices that can be used in our project (study described in section Systematic Review of Serious Games in the Medical Domain).
- Analysis of games and game-like applications for training of medical personnel using medical imaging, such as MalariaSpot, Cell Slider, VMATs and BioGames (see section Games and Game-Like Applications for Medical Training Using Medical Imaging). This study has been published in a paper at the 6th International Conference on Digital Health, 2016 (Rotaru & Rosemary, 2016). The full contents of this paper are available at Appendix A - ACM DIGITAL HEALTH 2016.
- Comparison between the main game development tools and frameworks. In this study we have considered factors such as the target platforms, game mechanics and interactions, tool features, license, tool documentation, performance and education assessment support. The comparison is fully described in Chapter 4 - Development Tools. This study has been published in the IEEE Educon 2016 Educational Games Conference (Calvo et al., 2016). The full paper is available at Appendix B - IEEE EDUCON 2016.
- An integrated system that satisfies the goals and constraints of this project is designed, implemented and described from the point of view of the final user. This solution focuses on

the maintainability overcoming the domain expert's lack of programming and development skills..

## 7.3 Future Work

The integrated system is ready to be tested with students at the Introduction to Cytopathology course at Harvard Medical School (i.e. formative evaluation). The estimated date for this experiment is February 2017 and, in the meantime, we will continue iterating to improve the final user experience by doing tests with external users and collecting feedback that can be used to fix bugs and improve the system.

One of the desired features is the integration with an educational assessment infrastructure. A *Learning Analytics* infrastructure can be easily integrated with the game mechanics of the proposed solution. Each challenge has an attached score that can be used to evaluate the overall result of a learner when it's playing a session. A *Learning Analytics* infrastructure improves the overall value received by the domain experts interested in knowing how much their students are improving. The chosen development framework, libGDX, has no direct support for any educational assessment mechanism. A possible feature is the integration of a specific educational feature.

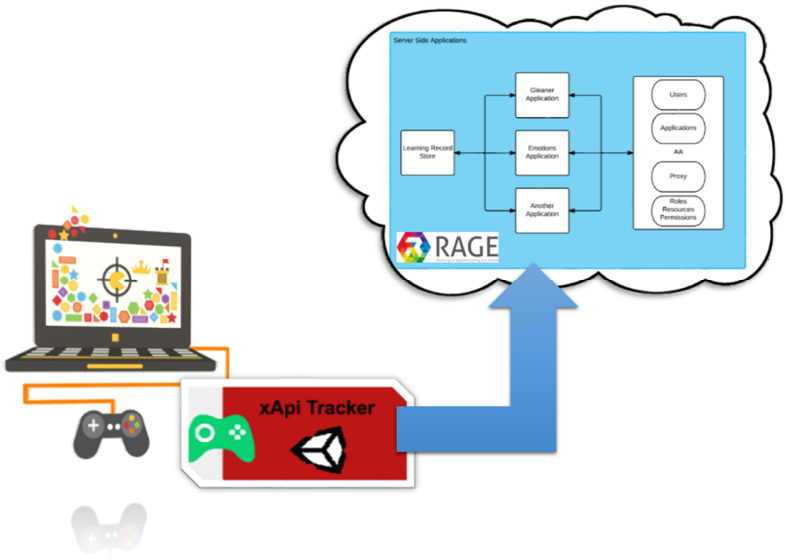



Figure 44. RAGE Analytics tracker connecting a game with the collection service

RAGE, Realising an Applied Gaming Eco-System, is a H2020 project that aims to develop a framework of technologies focused on greatly improving the serious game development and maintenance. The RAGE framework is composed of *assets*. An *asset* is a piece of software designed to satisfy a certain functionality

in a serious game. There are a lot of different RAGE *assets* for every functionality a serious game may require. The main purpose of an *asset* is to reduce the development and improve the maintainability and software reusability when developing a serious game

RAGE Analytics is an *asset* that provides tracking, analysis and dashboard creation features (see Figure 44). The RAGE Analytics project is developed by the e-UCM Research Group at the Complutense University of Madrid. One of its features is a client-side tracker *asset* (RAGE Analytics Tracker) available for different programming languages (java, JavaScript, C#) and for different development technologies (libGDX, Unity 3D and any C# video game). The Rage Analytics Tracker exposes an API for an easy integration with its *Learning Analytics* infrastructure. Since RAGE Analytics Tracker is available for libGDX, which is the technology used for the development of the game-like application of our project, we recommend using this solution in order to provide educational assessment to the stakeholders using the RAGE Analytics infrastructure.

The final aspect of the project will be a summative evaluation. Even if different formative evaluations have been carried out with experts during the development process to assess and improve the final version of the framework a more comprehensive evaluation with final users is intended. As previously described, once that the framework is stable and there is enough content an experiment with MGH residents and other medical personnel will be performed to evaluate the efficacy of the approach. Then this experiment will be extended to a different country with limited resources to check if we obtain similar results that in the previous evaluations. The results of this formal evaluation will be also contrasted with the data obtained from the learning analytics infrastructure.

# 8 Conclusiones y Trabajo Futuro

## 8.1 Conclusiones

Nuestro objetivo inicial estaba bien definido, profesores de la Escuela de Medicina de Harvard y el personal médico en el Hospital General de Massachusetts quieren una aplicación educativa para entrenar a citotecnólogos en áreas de recursos limitados del mundo. Antes de implementar la aplicación educativa en dichas zonas va a ser probada con estudiantes reales a un curso de Introducción a Citopatología de la Escuela Médica de Harvard.

Partimos con unos requisitos iniciales desafiantes, pero que conocemos desde el inicio del proyecto. La aplicación educativa tiene que funcionar sin una conexión a internet estable y en dispositivos Android de bajo coste. Éstos son los dispositivos utilizados para desplegar el sistema en áreas de recursos limitados del mundo, en las que, además, no podemos depender de una conexión a internet estable. Al mismo tiempo, la aplicación educativa tiene que funcionar en ordenadores, ya que va a ser probada en un curso de Introducción a Citopatología con estudiantes reales en la Escuela Médica de Harvard.

Al conocer nuestros objetivos y haber definido definido los requisitos iniciales, decidimos llevar a cabo varios estudios antes de tomar cualquier decisión de diseño o implementación de la aplicación. Con el fin de identificar las mejores prácticas de diseño y desarrollo de juegos serios en el campo de la salud, hemos realizado una revisión de literatura de los juegos serios en el ámbito médico. Por otra parte, para identificar las mejores mecánicas de juego para nuestro proyecto, hemos analizado diferentes juegos y aplicaciones similares (MalariaSpot, teléfono Slider, VMATs y BioGames) orientadas a la formación médica utilizando imágenes médicas. Después de esta primera serie de estudios, habíamos identificado los mejores mecánicas de juego para nuestro proyecto: tener una historia introductoria para establecer el contexto y la motivación del jugador, seguido por un tutorial que explica la aplicación educativa, a continuación, una serie de niveles para que el jugador complete y continuo feedback sobre el contenido de la aplicación para que el jugador pueda cómo mejorar sus respuestas y, por último, algunas de las características gamificación como seguimiento de puntuaciones a través de tablas de clasificación y temporizadores para cada nivel de la aplicación.

Después de identificar las mecánicas de juego que queremos implementar, el siguiente paso lógico es decidir qué tecnología de desarrollo es el más adecuada para nuestro proyecto. Decidimos analizar las herramientas de desarrollo más importantes en el sector de la industria de juegos, clasificándolos en cuatro clases diferentes. Desde las más flexibles y más difíciles de aprender (herramientas AAA) a más especializadas, más fáciles de aprender, pero no tan flexibles en términos de qué juegos podemos desarrollar con ellas (herramientas específicas tales como eAdventure). Por último, tuvimos que decidir entre dos candidatos Unity

3D o libGDX, una librería de desarrollo de juegos para el lenguaje Java. Ambos candidatos eran adecuados para nuestro caso proyecto en la mayoría de los aspectos, por ejemplo, los dos dan soporte para las plataformas de nuestro proyecto (HTML 5 y Android), tienen una activa comunidad de usuarios que publican juegos, un código fuente bien documentado, su licencia es de código abierto, gratuitas y proporcionan las características necesarias para desarrollar nuestra aplicación educativa. La gran diferencia entre ellos es que libGDX tiene un mayor rendimiento general, lo que, en nuestro caso es fundamental teniendo en cuenta que nuestra aplicación debe ejecutarse en dispositivos Android de bajo coste. Por esta razón hemos decidido utilizar libGDX como nuestra tecnología de desarrollo, además, el autor de este trabajo tenía experiencia usando la librería.

Una vez llegados a este punto sabemos (1) qué mecánicas de juego queremos implementar en nuestra aplicación educativa y (2) la tecnología de desarrollo se quiere utilizar. Con esta información, y teniendo en cuenta los requisitos y los objetivos del proyecto se diseñó un sistema integrado compuesto por una aplicación educativa, un editor web para el contenido y un servicio REST estandarizado. Teniendo en cuenta el objetivo inicial del proyecto y el sistema diseñado, se puede concluir que los siguientes objetivos se han completado con éxito

• Objetivo 1 - Hemos diseñado un sistema integrado que cumple con los requisitos iniciales. La aplicación educativa, Cytopathology Challenge, se ejecuta en los navegadores web HTML 5 para realizar pruebas en ordenadores, o incluso dispositivos móviles que utilizan un navegador web, con los estudiantes en un curso de Introducción a Citopatología en la Escuela Médica de Harvard.

• Objetivo 2 - La aplicación educativa se ejecuta sin problemas en los dispositivos Android de bajo coste y no requiere una conexión a internet para funcionar que hace que sea desplegable en áreas con recursos limitados, donde se utilizan estos dispositivos y no podemos depender de una conexión a internet estable. Por otra parte, las mecánicas de juego implementadas han sido probadas en los juegos basados en imágenes médicas. Más información acerca de las mecánicas de juego implementadas está disponible en el Capítulo 3 - Trabajo Relacionado y Aplicaciones.

• Objetivo 3 - Hemos desarrollado un editor web diseñado para ser fácil de utilizar por el personal médico. El editor web se puede utilizar para gestionar (crear, editar y eliminar) el contenido del sistema. Para ser capaz de gestionar el contenido del sistema se han definido dos modelos de datos principales utilizados por las aplicaciones educativas, Cursos y Retos. Un curso es un conjunto de retos que comparten el mismo tema. Los cursos se utilizan para agrupar desafíos y para filtrar y consultar más fácilmente desafíos. Un reto es un nivel del juego diseñada con fines educativos y, al mismo tiempo, para enganchar el interés de los jugadores. Los jugadores deben completar con éxito los desafíos para tener éxito en la aplicación educativa, y hay 5 tipos diferentes de desafíos.

• Objetivo 4 - El sistema está listo para ser probado con los estudiantes, pero la fecha estimada para las pruebas es en Febrero de 2017. Hasta entonces el sistema estará bajo mejoras continuas a través de la feedback externo con diferentes usuarios. Estamos pensando iterar varias veces, mientras hasta la fecha del evento, haciendo pruebas con los usuarios y abteniendo feedback con el fin de mejorar la experiencia del usuario final.

En resumen, el proyecto tenía un presupuesto muy limitado, y algunas limitaciones iniciales desafiantes. Después de varios estudios hemos diseñado e implementado un sistema que satisface los objetivos iniciales, como hemos concluido anteriormente. El sistema final consiste en

(1) una aplicación educativa para HTML5 y dispositivos Android de bajo coste,

(2) un servicio de backend RESTful para la gestión del contenido utilizado por la aplicación y

(3) un editor web diseñado para ser fácil de usar, eliminando la necesidad de los desarrolladores, donde los expertos de dominio pueden gestionar el contenido que se presenta a sus estudiantes.

## 8.2 Contribución

Las principales contribuciones de este proyecto son:

- Estudio de los juegos serios relacionados con el dominio de la salud buscando en diferentes bases de datos (ACM, BioMed, PubMed, etc.) clasificando los resultados obtenidos similares a nuestro caso de uso para identificar las mejores prácticas aplicables a nuestro proyecto (estudio descrito en la sección Systematic Review of Serious Games in the Medical Domain).
- Análisis de juegos y aplicaciones similares para la formación de personal médico utilizando imágenes médicas tales como MalariaSpot, Cell Slider, VMATs y BioGames (ver Games and Game-Like Applications for Medical Training Using Medical Imaging). Este estudio ha sido publicado en la Sexta Conferencia Internacional en Salud Digital (Rotaru & Rosemary, 2016). El artículo completo se encuentra disponible en el Apéndice A – ACM DIGITAL HEALTH 2016.
- Comparación de las principales herramientas y librerías de desarrollo de juegos teniendo en cuenta factores como las plataformas de desarrollo, la mecánica del juego e interacciones, características de la tecnología, licencia, documentación de la tecnología, el rendimiento y el soporte para la evaluación educativa. La comparación se describe en profundidad en el Capítulo 4 - Development Tools. Este estudio ha sido publicado en un artículo en la IEEE Educon 2016 Educational Games Conference (Calvo et al., 2016). El artículo completo se encuentra disponible en el Apéndice B – IEEE EDUCON 2016.

- Implementar un sistema que satisfaga los objetivos y requisitos del proyecto. El sistema está diseñado para eliminar la dependencia entre expertos de dominio y desarrolladores de contenido educativo.

## 8.3 Trabajo Futuro

El sistema está listo para ser probado con los estudiantes en el curso de Introducción a Citopatología en la Escuela Médica de Harvard. La fecha estimada para este experimento es Febrero 2017 y, mientras tanto, vamos a seguir iterando para mejorar la experiencia final del usuario al hacer pruebas con usuarios externos y obteniendo feedback que se pueda utilizar para corregir errores y mejorar el sistema.

Una de las mejoras es la integración con una infraestructura de evaluación educativa. Una infraestructura de aprendizaje basada en analíticas se puede integrar fácilmente con la mecánica del juego de la solución propuesta. Cada desafío tiene una calificación adjunta que puede ser usada para evaluar el resultado global de un alumno cuando se está jugando una sesión. Una infraestructura de aprendizaje permite a los expertos del dominio (profesores, investigadores, etc.) saber lo mucho que sus estudiantes están mejorando. La tecnología de desarrollo educativa utilizada, libGDX, no dispone de soporte para evaluación educativa, por lo que debería usarse alguna librería ya desarrollada que ofrezca soporte para evaluación educativa. Se recomienda utilizar la implementación RAGE.

RAGE, Realising an Applied Gaming Eco-System, es un proyecto H2020 que tiene como objetivo desarrollar un marco de tecnologías orientadas a la mejora del desarrollo y mantenimiento del juegos. RAGE está formado por *assets*. Un *asset* es un software diseñado para satisfacer una cierta funcionalidad en un juego serio. Existen diferentes *assets* RAGE para cada funcionalidad que un juego serio puede requerir. El propósito principal de un *asset* es reducir el desarrollo y mejorar el mantenimiento y la reutilización del software en el desarrollo de un juego serio

RAGE Analytics es un *asset* que proporciona funciones de seguimiento, análisis y creación de gráficas con resultados (ver Figure 44). El proyecto RAGE Analytics es desarrollado por el grupo de investigación e-UCM en la Universidad Complutense de Madrid. Una de sus aportaciones es un *asset,* que se ejecuta en el lado del cliente (RAGE Analytics Tracker), disponible en diferentes lenguajes de programación (Java, JavaScript, C #) y para diferentes tecnologías de desarrollo (libGDX, Unity y cualquier videojuego de C#). RAGE Analytics Tracker dispone de una API de comunicación con la infraestructura de analíticas que se ejecuta de lado del servidor. Se recomienda utilizar RAGE Analytics Tracker para importar las funcionalidades de un sistema de analíticas educativas debido a la fácil integración del sistema con la tecnología usada en el desarrollo del proyecto, libGDX.

El elemento final del proyecto será una evaluación sumativa. A lo largo del proyecto se han realizado evaluaciones formativas con expertos para evaluar y mejorar la versión final del entorno pero se pretende realizar también una evaluación más completa con usuarios finales. Como se ha descrito previamente, una vez que el entorno es estable y que hay suficiente contenido se pretende realizar un experimento con residentes del MGH y con otros miembros del personal médico para evaluar la eficacia de este sistema. Entonces se pretende ampliar este experimento a un país con recursos limitados para comprobar si se obtienen resultados similares. Además los resultados de esta evaluación formal se pretenden contrastar con los datos obtenidos automáticamente mediante analíticas de aprendizaje.

# References


Akl, E. A., Kairouz, V. F., & Sackett, K. M. (2013). Educational games for health professionals. … *Database Syst Rev.* Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/14651858.CD006411.pub4/pdf/\npapers3://publication/doi/10.1002/14651858.CD006411.pub4

Annetta, L. A., Minogue, J., Holmes, S. Y., & Cheng, M.-T. (2009). Investigating the impact of video games on high school students' engagement and learning about genetics. *Computers & Education*, *53*(1), 74–85. http://doi.org/10.1016/j.compedu.2008.12.020

Arnab, S., Dunwell, I., & Debattista, K. (2013). *Serious Games for Healthcare: Applications and Implications. Games for Health Journal* (Vol. 2). http://doi.org/10.1089/g4h.2013.0062

Barjis, I., Samarrai, W., & Smith, D. (2009). Modeling and simulation of 3-D virtual cell as a game.

Barzilai, S., & Blau, I. (2014). Scaffolding game-based learning: Impact on learning achievements, perceived learning, and game experiences. *Computers & Education*, *70*, 65–79. http://doi.org/10.1016/j.compedu.2013.08.003

Beale, I. L., Marin-Bowling, V. M., & Guthrie, N. K. (2006). Young Cancer Patients' Perceptions of a Video Game Used to Promote Self Care. *International Electronic Journal of Health Education*, *9*, 202–212. Retrieved from http://search.proquest.com.ezproxy.library.yorku.ca/docview/61913312?accountid=15182

Boulos, M. N. K., & Yang, S. P. (2013). Exergames for health and fitness: the roles of GPS and geosocial apps. *International Journal of Health Geographics*, *12*(1), 18. http://doi.org/10.1186/1476-072X-12-18

Brox, E., Fernandez-Luque, L., & Tøllefsen, T. (2011). Healthy Gaming – Video Game Design to promote Health. *Applied Clinical Informatics*, *2*(2), 128–142. http://doi.org/10.4338/ACI-2010-10-R-0060

Caldwell, C., & Christiansen, B. (2013). The Intersection of Video Games and Patient Empowerment: Case Study of a Real World Application. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death* (p. 12:1--12:7). http://doi.org/10.1145/2513002.2513018

Calvo, A., Rotaru, D. C., Freire, M., & Fernandez-manjon, B. (2016). Tools and Approaches for Simplifying Serious Games Development in Educational Settings, (April), 1188–1197.

Chen, J. (2007). Flow in games (and everything else). *Communications of the ACM*, *50*(4), 31. http://doi.org/10.1145/1232743.1232769

Cheng, M. T., Su, T., Huang, W. Y., & Chen, J. H. (2013). An educational game for learning human immunology: What do students learn and how do they perceive? *British Journal of Educational Technology*, *45*(5), 820–833. http://doi.org/10.1111/bjet.12098

Connolly, T. M., Boyle, E. a., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review

of empirical evidence on computer games and serious games. *Computers & Education*, *59*(2), 661–686. http://doi.org/10.1016/j.compedu.2012.03.004

Connolly, T. M., Boyle, E. A., Macarthur, E., Hainey, T., & Boyle, J. M. (2012). Computers & Education A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, *59*(2), 661–686. http://doi.org/10.1016/j.compedu.2012.03.004

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., … Players, F. (2010). Predicting protein structures with a multiplayer online game. *Nature*, *466*(7307), 756–760. http://doi.org/10.1038/nature09304

Corredor, J., Gaydos, M., & Squire, K. (2013). Seeing Change in Time: Video Games to Teach about Temporal Change in Scientific Phenomena. *Journal of Science Education and Technology*, (26), 1–20. http://doi.org/10.1007/s10956-013-9466-4

De Freitas, S. (2006). Learning in Immersive worlds A review of game-based learning Prepared for the JISC e-Learning Programme. *JISC eLearning Innovation*, *3.3*(October 14), 73. http://doi.org/10.1111/j.1467-8535.2009.01024.x

Dede, C. (2009). Immersive Interfaces for Engagement and Learning. *Science*, *323*(5910), 66–69. http://doi.org/10.1126/science.1167311

Dickey, M. D. (2005). Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development*, *53*(2), 67–83. http://doi.org/10.1007/BF02504866

e-Adventure e-Learning Games. (n.d.). Retrieved from http://e-adventure.e-ucm.es/

ESA. (2015). Game : Improving the economy, 7–9.

ESA, & ESA. (2015). 2015 Essential Facts About the Computer and Video Game Industry. *Social Science Computer Review*, *4*(1). Retrieved from http://www.theesa.com/facts/pdfs/ESA_EF_2008.pdf

Forbes - "Grand Theft Auto V" Crosses $1B In Sales, Biggest Entertainment Launch In History. (n.d.). Retrieved from http://www.forbes.com/sites/erikkain/2013/09/20/grand-theft-auto-v-crosses-1b-in-sales-biggest-entertainment-launch-in-history/

Fujiki, Y., Kazakos, K., Puri, C., Buddharaju, P., & Pavlidis, I. (2008). NEAT-o-Games : Blending Physical Activity and Fun in the Daily Routine. *Computer*, *6*(2), 1–22. http://doi.org/10.1145/1371216.1371224

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, *1*(1), 20. http://doi.org/10.1145/950566.950595

Good, B. M., Loguercio, S., Griffith, O. L., Nanis, M., Wu, C., & Su, A. I. (2014). The cure: design and evaluation of a crowdsourcing game for gene selection for breast cancer survival prediction. *JMIR Serious Games*, *2*(2), e7. http://doi.org/10.2196/games.3350

Hwang, G. J., & Wu, P. H. (2012). Advancements and trends in digital game-based learning research: A review of publications in selected journals from 2001 to 2010. *British Journal of Educational Technology*, *43*(1), 6–10. http://doi.org/10.1111/j.1467-8535.2011.01242.x

IGN - GTA 5 SALES HIT $1 BILLION IN THREE DAYS. (n.d.). Retrieved from http://www.ign.com/articles/2013/09/20/gta-5-sales-hit-1-billion-in-three-days

Jarvis, S., & Freitas, S. De. (2009). Evaluation of an Immersive Learning Programme to Support Triage Training. In *Games and Virtual Words for Serious Applications* (pp. 117–122). http://doi.org/10.1109/VS-GAMES.2009.31

Jr, J. C. R., Lynch, P. J., Cuddihy, L., Gentile, D. A., Klonsky, J., & Merrell, R. (2007). The Impact of Video Games on Training Surgeons in the 21st Century. *Archives of Surgery*, *142*(2), 181–186. http://doi.org/10.1001/archsurg.142.2.186

Kazemi, D. M., Cochran, a. R., Kelly, J. F., Cornelius, J. B., & Belk, C. (2013). Integrating mHealth mobile applications to reduce high risk drinking among underage students. *Health Education Journal*, *73*(3), 262–273. http://doi.org/10.1177/0017896912471044

Kirriemuir, J., & McFarlane, A. (2004). Literature Review in Games and learning - Futurelab. Retrieved from http://www.futurelab.org.uk/resources/publications_reports_articles/literature_reviews/Literature_Review378/

Kleinert, R., Wahba, R., Chang, D.-H., Plum, P., Hölscher, A. H., & Stippel, D. L. (2015). 3D Immersive Patient Simulators and Their Impact on Learning Success: A Thematic Review. *Journal of Medical Internet Research*, *17*(4), e91. http://doi.org/10.2196/jmir.3492

Kron, F. W., Gjerde, C. L., Sen, A., & Fetters, M. D. (2010). Medical student attitudes toward video games and related new media technologies in medical education. *BMC Medical Education*, *10*, 51–11. http://doi.org/10.1186/1472-6920-10-50

Li, W. H. C., Chung, J. O. K., & Ho, E. K. Y. (2011). The effectiveness of therapeutic play, using virtual reality computer games, in promoting the psychological well-being of children hospitalised with cancer. *Journal of Clinical Nursing*, *20*(15/16), 2135–2143. http://doi.org/10.1111/j.1365-2702.2011.03733.x

Luengo-Oroz, M. A., Arranz, A., & Frean, J. (2012). Crowdsourcing malaria parasite quantification: An online game for analyzing images of infected thick blood smears. *Journal of Medical Internet Research*, *14*, 1–13. http://doi.org/10.2196/jmir.2338

Lyons, E. J. (2014). Review of Games for Health: Proceedings of the 3 rd European Conference on Gaming and Playful Interaction in Health Care. *Games for Health Journal*, *3*(1), 49–52. http://doi.org/10.1089/g4h.2013.0083

MalariaSpot - Crouwsourced serious game fighting malaria. (n.d.). Retrieved from http://malariaspot.org/en/

Mavandadi, S., Feng, S., Yu, F., Dimitrov, S., Yu, R., & Ozcan, A. (2012). BioGames: A Platform for Crowd-Sourced Biomedical Image Analysis and Telediagnosis. *Games for Health Journal*, *1*(5), 373–376. http://doi.org/10.1089/g4h.2012.0054

Mayo, M. (2009). Video games: a route to large-scale STEM education? *Science*, (January), 79–82. http://doi.org/10.1126/science.1166900

Michael, D. R., & Chen, S. L. (2005). *Serious Games: Games That Educate, Train, and Inform. Education* (Vol. October 31). http://doi.org/10.1021/la104669k

Muehrer, R., Jenson, J., Friedberg, J., & Husain, N. (2012). Challenges and opportunities: Using a science-based video game in secondary school settings. *Cultural Studies of Science Education*, *7*(4), 783–805. http://doi.org/10.1007/s11422-012-9409-z

Muenchen, Robert A. (2014). The Popularity of Data Analysis Software, (April). Retrieved from http://r4stats.com/articles/popularity/

NMC, N. M. C. (2014). *NMC Horizon Report: 2014 Higher Education Edition. Austin, Texas: The New Media …*. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:NMC+Horizon+Report:+2014+Higher+Education+Edition#0

Papastergiou, M. (2009). Exploring the potential of computer and video games for health and physical education: A literature review. *Computers & Education*, *53*(3), 603–622. http://doi.org/10.1016/j.compedu.2009.04.001

Pempek, T. A., & Calvert, S. L. (2009). Tipping the balance: use of advergames to promote consumption of nutritious foods and beverages by low-income African American children. *Archives of Pediatrics & Adolescent Medicine*, *163*(7), 633–7. http://doi.org/10.1001/archpediatrics.2009.71

Perrotta, C., Featherstone, G., Aston, H., & Houghton, E. (2013). *Game-based learning: Latest evidence and future directions*. Retrieved from www.nfer.ac.uk

Preston, J. A. (2013). Serious Game Development : Case Study of the 2013 CDC Games For Health Game Jam. *Proceedings of the 2014 ACM International Workshop on Serious Games*, 39–43. http://doi.org/http://dx.doi.org/10.1145/2656719.2656721

Reichlin, L., Mani, N., McArthur, K., Harris, A. M., Rajan, N., & Dacso, C. C. (2011). Assessing the acceptability and usability of an interactive serious game in aiding treatment decisions for patients with localized prostate cancer. *Journal of Medical Internet Research*, *13*, e4. http://doi.org/10.2196/jmir.1519

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., … Silver, J. (2009). Scratch. *Communications of the ACM*, *52*(11), 60. http://doi.org/10.1145/1592761.1592779

Robert Kleinert, Heiermann, N., Wahba, R., Chang, D.-H., Hölscher, A. H., & Stippel, D. L. (2015). Design,

Realization, and First Validation of an Immersive Web-Based Virtual Patient Simulator for Training Clinical Decisions in Surgery.

Rotaru, D. C., & Rosemary, H. (2016). Design and development of a serious game for medical training in cytopathology, 4–5.

Sajjad, S., Abdullah, A. H., Sharif, M., & Mohsin, S. (2014). Psychotherapy Through Video Game to Target Illness Related Prob- lematic Behaviors of Children with Brain Tumor, 62–72.

Scanlon, L., O'Shea, E., O'Caoimh, R., & Suzanne Timmons. (2015). Technology Use and Frequency and Self-Rated Skills: A Survey of Community-Dwelling Older Adults.

Scratch - Imagine, Program, Share. (n.d.). Retrieved from https://scratch.mit.edu/

SICKO Serious Game. (n.d.). Retrieved from http://med.stanford.edu/sm/archive/sicko/game/SICKOTitle.html

Simone, L. V. E., Pryor, W. M., Belinson, Z., Salisbury, E. L., & Velan, G. M. (2015). Cytopathology whole slide images and virtual microscopy adaptive tutorials: A software pilot. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4629310/

Stanley, C., & Byrne, M. (2011). Predicting Tags for StackOverflow Posts. *Chil.Rice.Edu*, 414–419. Retrieved from http://chil.rice.edu/research/pdf/StanleyByrne2013StackOverflow.pdf

Thompson, D., Baranowski, T., Buday, R., Baranowski, J., Thompson, V., Jago, R., & Griffith, M. J. (2010). Serious Video Games for Health: How Behavioral Science Guided the Development of a Serious Video Game. *Simulation & Gaming*, *41*(4), 587–606. http://doi.org/10.1177/1046878108328087

Tüzün, H., Yilmaz-Soylu, M., Karakuş, T., Inal, Y., & Kizilkaya, G. (2009). The effects of computer games on primary school students' achievement and motivation in geography learning. *Computers and Education*, *52*(1), 68–77. http://doi.org/10.1016/j.compedu.2008.06.008

Ushaw, G., Davison, R., Eyre, J., & Morgan, G. (2015). Adopting Best Practices from the Games Industry in Development of Serious Games for Health. *Proceedings of the 5th International Conference on Digital Health 2015*.

Vourvopolous, A., Lucía Faria, A., Ponnam, K., & Bermudel i Badia, S. (2014). RehabCity: design and validation of a cognitive assessment and rehabilitation tool through gamified simulations of activities of daily living.

Wire, A. (2014). How playing games can change medicine's future. Retrieved from http://www.ama-assn.org/ama/pub/ama-wire/ama-wire/post/playing-games-can-change-medicines-future

Wouters, P., van der Spek, E. D., & van Oostendorp, H. (2009). Current practices in serious game research: A review from a learning outcomes perspective. … *Effective Practices*, 232–250. http://doi.org/10.4018/978-1-60566-360-9

# APPENDIX A. ACM DIGITAL HEALTH 2016

## Design and development of a serious game for medical training in cytopathology


Dan C. Rotaru
Complutense University of Madrid
C Profesor García Santesmases 9,
28040 Madrid, Spain
+34 692 682 845
drotaru@ucm.es

Baltasar Fernandez-Manjon
Complutense University of Madrid
C Profesor García Santesmases 9,
28040, Madrid, Spain
+34 609 232 095
balta@fdi.ucm.es

Avni Katri
LCS Massachusetts General Hospital
50 Staniford Street
Boston, MA, 02114,
USA
+1 (617) 726-3909
amkhatri@mgh.harvard.edu

Rosemary H. Tambouret
MGH & Harvard Medical School
Boston, MA, 02115,
USA
+1 (617) 726-3909
rtamboret@mgh.harvard.edu


We are designing and developing a serious game for cytopathology medical training. This game has some challenging technical requirement as to be deployable both in PCs and in low-cost Android tablets (e.g., 50$ Kindle Fire 7 tablet) with a limited budget. We reviewed some of the existing games or gamified e-learning modules to create a shared understanding between medical experts and developers about possible game mechanics and to identify which of those approaches can be suitable for our case.

There are numerous examples of games and game-like approaches successfully used in education. Those games are called "serious games" as their main goal is not only entertainment. Serious games for health have been increasingly used in the past years and there are examples in very different areas as games targeted at HIV prevention education, cancer diagnosis, dental pain, etc. and both for training medical personnel or oriented to patients [2].

In our project we want to create an educational game that can be used as a supplement to content for an Introduction to Cytopathology course. But, the use of serious games to train medical personnel in cytopathology is still a relatively uncharted field. Our project has a limited budget and some challenging technical requirements as the game should be also aimed to train cytologists in resource-limited areas of the world. That means, for instance, that the game should run in low-cost Android tablets and be fully functional without requiring continuous Internet connection.

The educational approach is focused on training the medical personnel in medical microscopy. The ability to morphologically identify normal and abnormal cells and to locate rare abnormal cells among many normal-appearing cells is the most difficult for cytotechnology students and pathology residents to learn. The learning objectives are to be presented in a gamified way, offering exciting, innovative, and effective methods for increasing the knowledge of the learner. Knowing which story and game characteristics (e.g. game mechanics) appeal to specific types of people would help tailor game design and behaviour change procedures to maximize effectiveness.



DH '16, April 11-13, 2016, Montréal, QC, Canada
ACM 978-1-4503-4224-7/16/04.
http://dx.doi.org/10.1145/2896338.2896371


Furthermore, implementing the gaming platform must be done taking in consideration different key technical project requirements such as the target resolution for Android tablets and PCs, code maintainability, applications design, etc.

A specific analysis of serious games for medical training was needed to obtain specific game mechanics that could be useful for our project. For instance, in our case all the assets of the game must be packed with the game and cannot be downloaded dynamically, which may imply size-related concerns for the Android version of the game. Most existing games use an Internet connection (not suitable for our project) and are executed on PCs where the size of the game and its assets did not pose a technical problem. We decided to review specific games and applications that have been used in medical education to train expert and non-expert personnel in domains relying on medical microscopy.

MalariaSpot is a game-like application oriented for crowd sourced collaboration in the diagnosis of malaria done by non-medical experts [1]. The objective of this game mechanic is to tag in a given amount of time as many intracellular parasites as possible in an image of a peripheral blood smear. There is an initial introductory mini-tutorial screen explaining what a parasite is and what is not and how to interact with the image. During the game, if the player finds all the parasites of an image in the allocated time, a new image will be loaded up. Each image should be considered as a level of the game, therefore, a player can analyze several images (levels) in a single game. There are several game mechanics to reinforce the player engagement. Firstly, the players receive continuous feedback. For instance, each click is represented with an icon that indicates a correct or incorrect selection. Furthermore, if a player misidentifies a target and clicks in the wrong one (e.g. on a leukocyte) the penalty is a reduction of the remaining time available and the final score of the level. The players' score is tracked on a leader board.

In our case, image size may pose a technical problem for the tablets video memory, especially if the image is very large. A possible solution can be to slice the image in multiple tiles that are used to render the entire map. The tiles can be loaded and unloaded from the graphic memory dynamically as they are shown in the screen or hidden.

The Cytopathology virtual microscopy adaptive tutorials (VMATs) use an adaptive e-learning platform that includes whole slide images for pathology education and training of both students and specialists [6]. VMATs are designed to "adapt" to the user's decision-making and aid with possible misconceptions through immediate feedback. The gameplay is composed of a text-based

question on one part of the screen and a related image (slide) on the other side of the screen. There are several mechanics that provide a game-like behavior enhancing the gamified interaction. Firstly, the question format changes (e.g. multiple-choice, drop-down lists, drag and drop type questions, fill-in-the-blank). Furthermore, immediate feedback is provided based on the learner's responses with more information about the quiz or about a specific area on the slide.

The VMAT approach can be reused in our project as it is oriented to cytopathology and it takes as a content starting point a set of teaching slides. But VMATs are only for PC and rely on a continuous Internet connection as data is continuously collected from these interactions for adaptive purposes and to provide evidence about the effectiveness of the quizzes. The size of the slide image may also present technical problems for the tablet hardware. A possible solution is to reduce the size of the slides to an acceptable level. VMATs were created using an intelligent tutoring system called AeLP (Adaptive eLearning Platform) that is completely web-based and eases the development of adaptive learning materials. Even though there are several applications and simulations created using the AeLP (e.g. VMATs or the Western Botting vLAB [4]), those applications do not meet the requirements of our project, since they rely on a stable Internet connection and are not optimized to be displayed on low cost Android tablets. But of course there are some design choices that can be reused. For instance, the way to integrate introductory information inside the formative experience, how the tasks are associated with questions and how different media is embedded within the user interface (e.g. interactive images) to enhance the student learning.

BioGames is a training game that helps identify malaria infected cells [3]. The game has an introductory tutorial and challenges the player to identify infected and uninfected cells. For each level a set of cells is displayed to the player. The player has to label the available cells either as “positive”, “questionable” or “negative”. Players receive a score depending on their performance labeling cells. There is also a progress bar as a visual feedback and the players’ top scores are integrated with a leaderboard system to promote competition and to improve engagement. Furthermore, this type of game mechanics can be easily integrated with an analytics tracking system to collect useful interaction data.

To reuse a similar approach in our game we must consider the size of the final game. Since all the images must be packed inside the game (cannot be downloaded dynamically), the number of cell images available to the player might increase the size of the game considerably. This might become a problem for Android devices.

Cell Slider is a game developed in collaboration between Cancer Research UK and citizen science experts Zooniverse [5]. The players must examine tumor tissue samples images and identify cancerous cells by answering simple questions about what they see in the image. To increase the validity of the answers, several people review the images. Cell Slider ask its players to identify specific items in tissue samples images and includes quiz challenges composed of text-based questions and a related image.

From this review we observe there are very different approaches and game mechanics that can be identified as best practices and reused or adapted for the design of a new game or game-like application to train medical personnel in the analysis of medical images for cytopathology.

We have concluded that one possible design could be presenting the learner with a short story to set up the context, followed by a concise tutorial explaining the game mechanics and ending with the player having to overcome different challenges (i.e. levels) that can be measured to assess the learner’s progression. A challenge, in its simplest form, could be a multiple choice question about a concept, an image or a specific region of an image where the user should identify if there is any anomaly or special circumstance. There could be a set of challenges, questions or puzzles from which a randomized sample is taken every time the learner starts playing a session. The challenges may also vary in difficulty as the learner advances and could have a gamification metric associated (allowing the creation of rankings between players) increasing the content diversity for each gameplay session. This design presents the content as a progression of events - initial story, basic concepts description, progressive challenges - that can be easily understood by the player. The initial story provides a supporting narrative meant to address the learner's motivation and interest. This initial design should be aligned with the specificities of our project.

Some of the analyzed systems capture user interaction data with different purposes (e.g. adaptation, leaderboard) and we consider that this is the correct approach even if in our case the game deployed in tablets cannot rely on a continuous Internet connection.

To conclude, we believe that the review done and the conclusions obtained about designing serious games for medical training in Cytopathology provide a solid ground for developing our prototype, as part of the early stage of a PhD. Next steps in the project are the completion of the initial prototype and the evaluation with medical students at the Harvard Medical School.

## 1. ACKNOWLEDGEMENTS

The e-UCM research group has been partially funded by Regional Government of Madrid (eMadrid S2013/ICE-2715), by the Ministry of Education (TIN2013-46149-C2-1-R) and by the European Commission (RAGE H2020-ICT-2014-1-644187, BEACONING H2020-ICT-2015-687676).

## 2. REFERENCES

1. Miguel Angel Luengo-Oroz, Asier Arranz, and John Frean. 2012. Crowdsourcing malaria parasite quantification: An online game for analyzing images of infected thick blood smears. *Journal of Medical Internet Research* 14: 1–13.
2. Elizabeth J. Lyons. 2014. Review of Games for Health: Proceedings of the 3 rd European Conference on Gaming and Playful Interaction in Health Care. *Games for Health Journal* 3, 1: 49–52.
3. Sam Mavandadi, Steve Feng, Frank Yu, Stoyan Dimitrov, Richard Yu, and Aydogan Ozcan. 2012. BioGames: A Platform for Crowd-Sourced Biomedical Image Analysis and Telediagnosis. *Games for health journal* 1, 5: 373–376.
4. Patsie Polly, Nadine Marcus, Danni Maguire, Zack Belinson, and Gary M Velan. 2014. Evaluation of an adaptive virtual laboratory environment using Western Blotting for diagnosis of disease. *BMC medical education* 14, 1: 222.
5. Mark Schrope. 2013. Solving tough problems with games. *Pnas* 110, 18: 7104–7106.
6. L. Van Es Simone, Wendy M. Pryor, Zack Belinson, Elizabeth L. Salisbury, and Gary M. Velan. 2015. Cytopathology whole slide images and virtual microscopy adaptive tutorials: A software pilot. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4629310/

# APPENDIX B. IEEE EDUCON 2016

# Tools and Approaches for Simplifying Serious Games Development in Educational Settings

Antonio Calvo, Dan C. Rotaru, Manuel Freire, Baltasar Fernandez-Manjon
Dept. Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Facultad de Informática
C/ Profesor JoseGarciaSantesmases, 9 28040 Madrid, Spain
{antcal01, drotaru, manuel.freire, balta}@fdi.ucm.es

***Abstract*—Serious Games can benefit from the commercial video games industry by taking advantage of current development tools. However, the economics and requirements of serious games and commercial games are very different. In this paper, we describe the factors that impact the total cost of ownership of serious games used in educational settings, review the specific requirements of games used as learning material, and analyze the different development tools available in the industry highlighting their advantages and disadvantages that must be taken into account when using them to develop a serious game.**

***Keywords—serious games; total cost of ownership; applied games; learning analytics***

## I. INTRODUCTION

Digital games have increasingly gained relevance in society to the point of becoming one of the most popular forms of entertainment. The total revenue of the digital games sector, in the US market alone has grown from USD 2.6 billion in 1996 to over 15.4 billion in 2013 [1]. The digital games sector has suffered a moderate slowdown in the last years [1] due to the economic crisis, but is still able to generate massive revenues. For instance, in September 2013 Rockstar released *Grand Theft Auto V*, which generated more than USD 1 billion in retail sales during its first three days on sale, being the fastest entertainment product to reach this milestone, including digital games and feature films [2], [3]. However, those big budgets and numbers are orders of magnitude removed from those of serious games or educational games.

Digital games, when used to further goals such as health or education rather than simply entertain, are called serious games (SGs) [5]. Continuous education, at all ages, is becoming one of the great priorities for our society, and increasing student motivation, always an important problem in education, is now more relevant than ever. Since large and diverse portions of the population already play games regularly as part of their daily or weekly routine, serious games are a powerful tool to increase student engagement and motivation. Games provide players with challenging and immersive environments where they can fail without consequences. This safe, failure-tolerant environment allows players to experience situations from different points of view, or experiment freely and exercise their curiosity circumstances that are good precursors of learning. By providing constant feedback to the player, games can be used for procedural learning [4]. And by motivating and challenging the players to explore and overcome specific problems, serious games are gaining interest in teaching and training students. Serious games have been successfully applied to different domains such as health [6]–[9], marketing [10], research [11], and to multiple other disciplines, where they have proven to increase student motivation in the learning process [12]–[14] stimulating their focus and concentration [5], [15], [16]. Studies have shown SGs increasing academic performance [17]–[19] and providing an optimized learning process [20], [21], as part of a growing set of experiments that prove the effectiveness of correctly designed and implemented serious games [22]–[26].

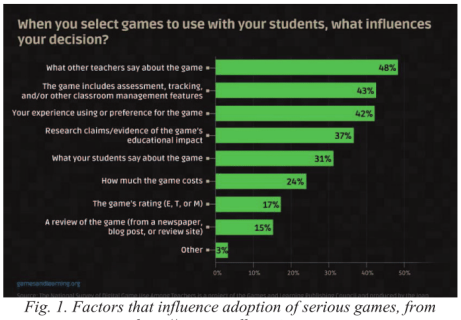



*Fig. 1. Factors that influence adoption of serious games, from http://gamesandlearning.org*

However, to generalize the use of games in formal education settings, educators need additional support. Beyond positive reviews by other teachers, games that include assessment, tracking and other classroom management features are strongly preferred (see Fig. 1). Evidence of educational impact is also important. Both factors can be addressed by using new game learning analytics techniques. However, these techniques need to be applied early and often, ideally starting during the design stages of the SG and continuing through the final implementation. Additionally, once a game has been considered educational content, it should remain available and ready to use regardless of technological changes or platform evolution. The decision to benefit from SGs or forego their usage completely hinges on the total cost of ownership (TCO)

978-1-4673-8633-3/16/$31.00 ©2016 IEEE 10-13 April 2016, Abu Dhabi, UAE
**2016 IEEE Global Engineering Education Conference (EDUCON)**
Page 1188

of the game including the cost of developing, fine-tuning, deploying and maintaining such a game.

This paper describes and analyzes the main factors that contribute to serious game TCO, and proposes strategies to keep this TCO in line with the SGs' requirements (including budget and maintainability). Section II describes the different lifecycle models for Serious Games. Section III analyzes different approaches to educational game development, mostly applicable to the game developers themselves. Section IV compares the different types of tools from a developer perspective. Finally, Section V presents our conclusions and outlines future work.

## II. Serious Game Lifecycles

The growth of the games industry and market is echoed by a corresponding growth in game development. There are different methodologies and authoring tools that improve the game development process, which requires a careful integration of multidisciplinary skills and efforts [27]. Furthermore, for SGs that have an educational purpose, this process is even more demanding, as not only are budgets much smaller: it is also necessary make the educational aspects of the game work while maintaining high player engagement [28].

### A. COTS games

Commercial Off-the-Shelf (COTS) games are, as their name indicates, commercially available digital (computer or console) games that are designed for entertainment rather than educational purposes. Some of these games can be given an alternate use in classrooms. However, making effective use of commercial games in the classroom requires careful though on how to extract this unintended pedagogical value. Indeed, as the following quote from [30] illustrates, certain games can be used to illustrate many more topics than teachersmay initially expect:

> "While some might assume that *Zoo Tycoon* might have application for biology, zoology, and ecology from the title alone, many would be surprised to learn that some of the other primary content areas for this game are economics, business, marketing, and mathematics".

The ability of the teachers to find and convey these educational aspects of the game is less relevant that their knowledge of the curriculum with which they are working to successfully achieve educational objectives [29]. And because the commercial games are not made to teach content, they will not be sufficient as the only teaching tool. Quoting from [30]:

> "As the designer, you will need to identify where there are gaps and inaccuracies in the game content, and where the strategies the game supports for solving the challenges do not align with your learning outcomes (e.g., trial and error vs. reasoned thinking) or may lead to misconceptions or an incomplete picture of the content and skills."

Before using these games, the teacher must research how to best integrate the game into the classroom. The fixed duration of classes is a significant constraint when planning and implementing game sessions in schools [29].

Multiple online sources advocate the use of COTS for classroom learning experiences, such as [29], [31]. *Civilization, Age of Empires II, CSI, Limbo, Universe Sandbox, The Sims 2, Rollercoaster Tycoon, Sim City 4, Imperium* and *Tropico* are only some examples of these kind of games used to teach history, forensics, criminal justice, physics, social sciences, civil engineering, business management etc.

Using COTS in the learning process is often more cost-effective than developing serious games from scratch [32]. However, finding a COTS game that can be used to teach a specific set of learning requirements may be impractical – such a game may simply not exist. Additionally, COTS games may require substantial modification to provide the type of game learning analytics and classroom support that teachers would take for granted in actual serious games. While it is possible to completely forego this support and rely on sharing game-play experiences through traditional classroom interactions, the lack of in-built support for evaluation requires more effort for teachers in COTS-used-as-SGs than in pure SGs.

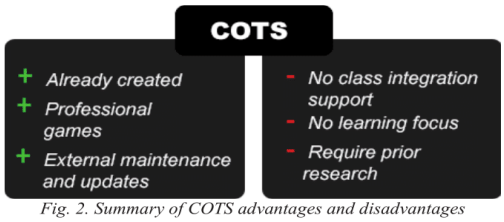



*Fig. 2. Summary of COTS advantages and disadvantages*

### B. Games as a Service

Another second way to provision serious games in learning environments is to hire the services of specialized companies that provide both the game and the associated infrastructure "as a service". These businesses use learning games as services to train students and employees or to provide tools to aid in their instruction. An example is GameLearn [33], which provides games for improving employee skills. One of their games-as-a-service (GaaS) is *Merchants*, which focuses on improving negotiation and conflict-resolution skills, while another, *Pacific*, provides leadership and team-management training. A very different GaaS, *Classcraft* [34], aims to transform any classroom into a role-playing game that fosters stronger student collaboration and encourages better behaviour [34]. Classcraft allows the teachers to introduce personalized questions in the game and rewards the students with in-game bounties.

GaaS provide a set of benefits, such as an easy integration with the learning process, control and monitoring tools, and useful training, among others. GaaS are designed to be quickly integrated with the teacher's learning process providing at least a web-based platform that enables teachers and students to easily manage all the complexities on the game without installing software. GaaS, unlike COTS, also have an analytics layer that supports student assessment to give a clear view of how the game is being played. GaaS teach a series of soft skills such as leadership, team management, productivity, negotiation and team work, among others. Their cost it is considerably lower than the costs of developing a serious game from scratch.

978-1-4673-8633-3/16/$31.00 ©2016 IEEE 10-13 April 2016, Abu Dhabi, UAE
**2016 IEEE Global Engineering Education Conference (EDUCON)**

However, there are some concerns that should be taken in consideration. The collected data in stored on the servers of the company supporting the game, which may result in data privacy concerns. The game's internationalization might suppose a problem if the desired language is not supported by the game. Additionally, GaaS companies are mostly interested in building a game once and serving it repeatedly to different corporate customers. Therefore, they are scarce or non-existent

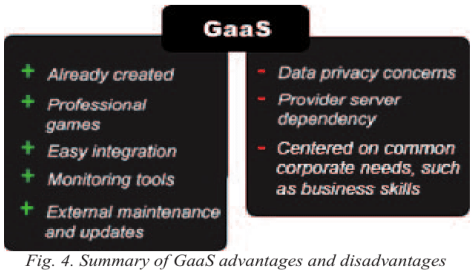



*Fig. 4. Summary of GaaS advantages and disadvantages*

in knowledge areas that are in lower demand across the enterprise.

### C. Pre-existing Serious Games

These are games that, unlike COTS, have been developed specifically for learning specific topics. Unlike GaaS, the client only has to pay for the game once (if at all – many are freely available), and unlike COTS, the game is specifically designed with learning goals in mind, making it much more amenable to classroom use.

There are many examples of specialized games. For instance, learning a new language has become a demanded skill in our society. Duolingo [35] is the perhaps the most influential language-learning game, and has been proven to significantly increase the overall average language skills of its players[36],[37]; while offering study courses for languages such as English, Spanish, French, German, Italian, Portuguese, Irish or Russian. *Citizen Science* [38]allows players to attach a real-world context to the research that is done to understand fresh water science. *CodeSpells* teaches programming. *Relive* is a first person 3D game developed by the Italian Resuscitation Council, using the Unity engine, which instructs on CPR protocol methods. Relive focuses in preparing its players to intervene in cases of cardiac arrest, offering two different game modes: a story mode where the player must achieve certain objectives in order to move forward; and a tournament mode, based on a simulated emergency scene.

As in COTS, even though there are many specialized SGs, sometimes it can be hard to find an existing specialized game that suits a specific pedagogical goal. However, if the teacher does find one, they are certainly the easiest way to introduce game-based learning into a classroom.

Unlike GaaS, some specialized games might not provide learning analytics functionalities. In order to implement the learning assessment support, the total cost will increase. On the other hand, if the game provides learning analytics functionalities, the collected data privacy policy must be considered. There might be data privacy concerns if the collected data is stored on third-party servers. If the game is executed locally and does not store player data in external servers, data privacy concerns are greatly reduced.

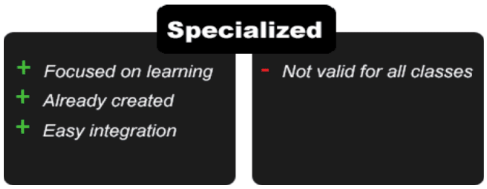



*Fig. 3. Pre-existing SG advantages and disadvantages*

### D. Developing serious games ad-hoc

There are authoring tools that simplify such a complex endeavour, even to the point of not requiring any programming skills. For instance, *Adventure Game Studio* or *RPG maker* both allow the development of an adventure or role-playing game without writing a single line of code. These tools contrast with general-purpose environments, such as *Unity*, *ShiVa* or *Game Maker Studio*, which allow the creation of much more complex games of different genres, and are suitable for amateur and indie developers, and even for professional studios of much greater size and budget.

While authoring tools assist with the actual game code, developing a game requires the use of a dizzying array of additional programs to manage and create the graphics and sounds that will make it come to life. Image editors, 3D editors –such as *Maya, Blender* or *3DS Max*–, sound authoring tools, and so on are highly specialized, and commercial game budgets are often dominated by *game asset* development, as such artistic elements are often termed. Even in low-budget SGs, it is often necessary to hire a graphic designer for some assets, buy others, and/or search for freely available alternatives.

Developing a game also requires the choice of the platforms where it is going to be played and what data is more important or relevant for the course or for students. The widespread use of the Unity game engine, providing an easy-to-use graphic user interface editor, has increased the amount of people interested in game development, including professional development teams, small indie developers and people with limited programming skills and development knowledge.

978-1-4673-8633-3/16/$31.00 ©2016 IEEE 10-13 April 2016, Abu Dhabi, UAE
**2016 IEEE Global Engineering Education Conference (EDUCON)**
Page 1190

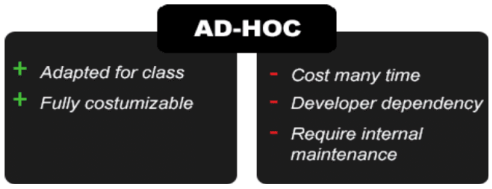



*Fig. 5. Ad-hoc SG advantages and disadvantages.*

The option of developing a game from scratch is, without doubt, the most expensive in terms of time and resources: it requires the use of multidisciplinary skills –design, implementation, content creation, etc– in order to scaffold, code and flesh out the game. Furthermore, the costs are increased if an additional layer of learning analytics is required. On the other hand, developing a serious game ad-hoc provides the highest amount of control over the final result. This is relevant when the learning requirements are very specific or demanding.

## III. Game Development And Approaches

There are a many of different tools available for digital game development, depending on factors such as game genre, purpose, platform, team size or available budget. These tools have in common the little support provided for any activity related to the design of the game. This limitation is especially relevant at the very beginning of the process, when the team is focused on capturing and evolving the concept of the digital game into a stable design ready for implementation.

This section provides a concise overview of the state of the art in game development tools, structured into four subsections.

### *A. AAA*

In the authoring tools industry, *AAA* –pronounced "triple A"– is a classification term used for authoring tools intended for teams with the highest development budgets and levels of promotion or the highest ratings by reviewers. An authoring tool considered to be AAA is therefore expected to be high quality, and focused on the development of high quality games – a.k.a. AAA games. To achieve this goal, these tools provide optimized features needed in a game (artificial intelligence, physics engine, animations, level design, etc.) as well as support for the roles that take part in the development (programmers, graphic artists, testers and designers). Their user interface is intended for experts, focused on implementation and advanced features provided by the tool to create the game, and leaving aside the features that would allow for rapid prototyping of game ideas.

Among other features, these environments provide an engine that abstracts platform-dependent features from the game code. AAA authoring tools are designed by and for programmers experienced in game development, and manage different features such as assets, input/output, network connections, the rendering pipeline, etc.

Regarding content creation, level designers can create new content for the game without modifying any code from the engine itself. These editors also provide authoring tools for animations and assets targeted at artists with little or no programming knowledge. *CryEngine* and *Unreal* Engine are two outstanding examples. Developed by *Crytek* and *Epic Games* respectively, they were initially used to create two successful commercial entertainment games (*Unreal Tournament* and *FarCry*), and have been actively improved ever since. Both have been used in dozens of other commercial games across different platforms (consoles, PC and mobile devices) and feature multiple licensing tiers, which can represent a substantial portion of the game development budget. For example, the highest tier for Unreal Engine 4 requires 5% of the total game profits.

### *B. All in one*

The AAA tools previously exposed are very powerful but at the same time their cost is too high for most developers. The team that created CryEngine numbers nearly 300 game-development professionals from over the world. There are other semi-professional tools created for teams with lower budgets.

One of the most popular is Unity, an all-in-one (AiO) tool equipped with all the necessary modules to configure all the aspects of a game, including the ability to export for different platforms (consoles, PC, Mac, Web and mobile devices). Currently, it is also being used for mobile development by professional teams. Nintendo's Wii U video game console uses Unity as the default software development kit (SDK) including a free copy with each Wii U developer license.

AiO tools lack native support for content creation, which is done with other tools such as Maya and *3D Studio*.

### *C. Frameworks*

The main focus of these tools is to provide lower-level access to the game engine. In this section we can find engines, frameworks and libraries focused on digital games, which can be commercial or free, and are designed so that the programmer makes use of its desired development environment (Eclipse, NetBeans, IntelliJ, Visual Studio, etc.). Examples are *libGDX*, *Ogre3D* or *Allegro*.

In general these tools require advanced game programming knowledge and can also target different platforms.

### *D. Specialized tools*

These tools have less functionality and potential but they simplify the game creation process in order to bring them to a more casual public. They are mostly used in education, aimed at teachers and students to allow them to make simple games with a high educational value. One of the first examples is GameMaker, developed by Mark Overmars. Scratch [39] is another popular tool, where kids learn to program while they

create digital games [40]. eAdventure [41] is a tool that started out with the goal of allowing teachers to create educational games; it focuses on the development of two different types of 2D *point-and-click* games: *first person*, where the scene is seen through the eyes of the main player, and *third person*, where the player is represented as an avatar in the scene.

## IV. Comparison Between The Different Kind Of Tools

The use of the different types of tools has effects on the monetary costs, time, and the end result. It is important to have a deeper understanding about these tools and their applicability. Below we analyze some of the properties to consider.

### A. Platforms

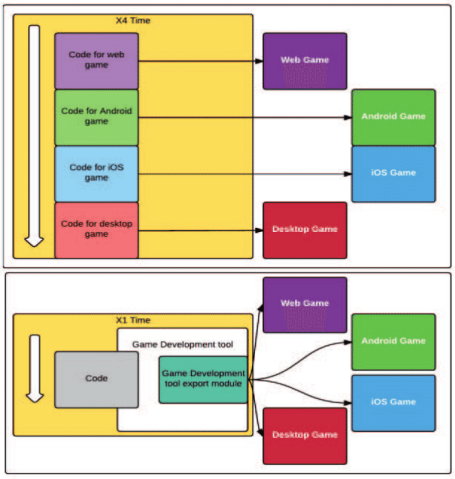



*Fig. 6. Multi-platform game development time costs comparison between the traditional method (top) and making use of a tool/framework (bottom).*

One of the first things that should be considered in a game development is the set of target platforms that it should be runnable in. In serious games, the most common environment is the desktop, where the three main Operating Systems are Windows, Mac and Linux. With the expansion of mobile devices,it is increasingly frequent to find games developed for mobile platforms, such as Android, iOS or Windows Phone. Some tools do not have the capacity to export the game for all these platforms, and lack of support for the chosen platforms often determines tool choice. Fig. 6 illustrates the time-savings realized by using multi-platform game development frameworks or tools.

AAA tools are specialized in high budget professional games and deploy to the highest amount of platforms. Their engines can create games with the latest and greatest graphics quality, physics and technologies. On the other side, frameworks and specialized tools have more platform deployment restrictions. Generally they provide support for one or two platforms (i.e. Windows and Android). As an exception, the libGDX framework supports development for Windows, Mac, Linux, Android, iOS and HTML5.

Unity has the industry-leading multi-platform support. Unity can export to a very wide range of devices across different platforms. IOS, Android, Windows Phone, BlackBerry and Tizen are the main mobile devices platforms, and are all supported by Unity. For the desktop environment Unity can export to Windows, Windows Store Apps, Mac, Linux and Steam OS (Linux). Unity also targets the web through its Unity Web Player plugin or by directly exporting to WebGL. The past and current generation of game consoles have sold an important amount of units, as we can see in Fig. 7.

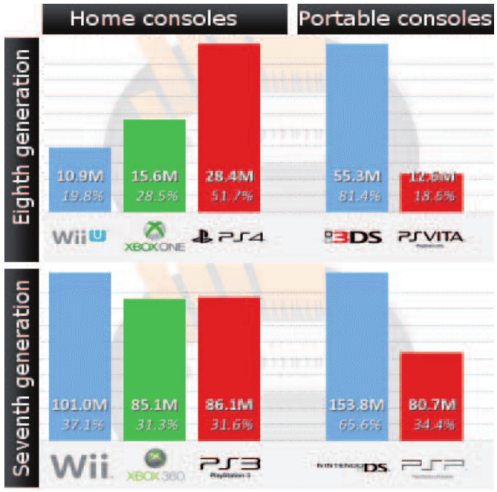



*Fig. 7. Consoles sales as of 14th November 2015, from http://www.vgchartz.com/*

Unity allows the developers to target PlayStation 4, PlayStation 3, Xbox One, XBox 360, PlayStation Mobile, PlayStation Vita and Wii U free of charge. Publishing to a console requires an additional approval process that varies from platform holder to platform holder. Unity also provides support for Virtual Reality and Augmented Reality platforms such as Oculus Rift, Gear VR and PlayStation VR. Microsoft HoloLens support has been announced –as of April 29, 2015– and is still under development as of November 2015. Other

platforms supported by Unity are Android TV, Samsung Smart TV and Windows Universal Platform.

The difficulty to publish games on different platforms must be considered by the developers when choosing a desired engine or framework. For instance, while libGDX does a good job of easing deployment to different platforms, multi-platform deployment typically entails multiple rounds of debugging and additional testing. The libGDX code base is written in Java and provides great cross-platform integration between desktop –Windows, Mac, Linux– and Android devices as long as the developer keeps the source code compatible with Java version 6 during compilation. For HTML5 deployment, libGDX uses Google Web Toolkit to compile the source to highly optimized JavaScript code that runs across all browsers.

Unity offers a faster solution by providing a user interface for the deployment process. Unity's solution requires less platform-specific knowledge from the developers and can speed up the development process. There is typically some minimal effort required, such as integrating with each platform's store for in-app purchases.

Developing games with technologies that might get discontinued and deprecated is arisk that can be prevented by carefully choosing the most suitable development framework or engine. For instance, *Science Pirates: The Curse of Captain Brownbeard* is a case of an educational game that is now difficult to play because of an obsolete target platform. The proposed solution is to invest in maintainable game development frameworks or engines that can deploy to multiple platforms from the same codebase, that are properly documented, have the support of active communities and provide features that can satisfy the game's requirements.

### *B. Game mechanics and interactions*

Another thing to keep in mind is the kind of game and the mechanics that must be developed. Games may vary in features, mechanics, and user interaction. Some tools only support the development of a specific type of games, especially the specialized tools –eAdventure focuses on *point'n'click* game development– that generally only allow the development of one or two genres. RPG Maker is a specialized tool targeting RPG games. AAA and all-in-one tools allow the creation of a much broader spectrum of game genres, e.g. platforms, first or third person shooters, adventures, simulations, strategy games, etc.

Specialized tools provide easier ways to create simple games, faster and with lower costs because these games require fewer scripts and elements, and their graphic user interface editor is highly optimized for the creation of a specific game genre. A caveat is the difficulty to innovate or add new features to these kinds of games. As an example, while eAdventure, allows the creation of *point'n'click* games in few hours assuming pre-existing graphical assets, with all mechanics and characteristics that characterize the adventure graphics games, adding additional game mechanics, such as 3D shooting mini-games, would be difficult or impossible.

### *C. Features*

In order to choose the best game development framework or engine, the nature of the graphics –2D or 3D– must be taken into consideration. Some tools can only make 2D games. It is necessary to keep in mind that not all the tools have the same affordances for development.

Unity supports both approaches, but it was designed mainly for 3D game development while libGDX, which has been originally designed for 2D games, also supports 3D. Unity has officially supported 2D development since the release of its 2D module, introduced in Unity 4.3 (November, 2013). The 2D dedicated engine in Unity was introduced because of the specific challenges in 2D game development, for instance the use of sprites. Unity's 2D module introduced tools to work with sprites and a physics specialized for 2D mechanics, tiles editor, sprite animations, masks, improved packages manager and a better performance for the 2D engine.

Game development frameworks are different than game engines because they provide different features. Game development frameworks are a collection of lower level libraries, tools and other frameworks used to create games. Game engines encapsulate powerful logic designed to create games and provide more high-level features. Game engines may have a graphic user interface editor while a framework is mainly code-based. Unity has a visual editor that can increase development speed. Developers have to consider whether the tool is actively supported or there is a risk of it becoming discontinued. The frameworks and engines have to continuously update their features with the new emerging technologies, improving the game's development workflow and guaranteeing that deployment to newly-appeared platforms will remain an option.

Also, engines such as Unity and Unreal have been designed considering the profiles of game developers and also designers, providing easy-to-use *drag and drop* editor user interfaces. They also provide *asset stores*, repositories of different kinds of assets –graphics, sounds, game objects, etc.– that can either be downloaded for free or for a fee.

Yet another important feature is ease of testing. Engines and tools that export to mobile devices allow the developers to test the game in the platform while the game is developed. Having to export the game to a different platform every time a new feature must be tested or to find and fix a bug can be tedious. It is important that the tool allows the game to be easily played from within the same platform that it has been developed in. Engines like Unity or Construct2 have this property, but not all frameworks do. For example, while libGDX does allow such testing, AndEngine or E3roid do not.

### *D. License and Cost*

The license of a game development framework or engine may be a critical factor when choosing which to use. In order to be competitive, Unreal Engine's license has been made free, with the condition of paying a 5% royalty on purchases of games and applications released by the developers after the first USD 3,000 of revenue per product per quarter.

LibGDX is licensed under Apache 2.0 and maintained by its user community. The code is open source, available at GitHub. Developers can use it free of charge, without strings attached in commercial and non-commercial projects.

Unity has a free version –the Personal Edition– available to developers with revenue under USD 100,000 in the previous fiscal year. Unity's Professional Edition license has a minimum cost of USD 75/month and packs additional features not available in the Personal Edition: customizable splash screen, Unity Analytics Pro, prioritized bug handling, Game Performance Profiling, Unlimited Revenue and Funding, professional editor skin, etc. Unity offers support to large enterprises with its tailored Enterprise Solutions, a package with additional features such as source code licensing, on-demand support and volume discounts available on Unity licenses. Unity Education aims to support learners, educators and educational institutions through special offers on licenses, content, services and resources –e.g. Professional Skills Standard and Curricular Framework– a grand program for secondary institutions, discounted tickets for students and faculty to Unity's Unite developer conferences, etc.

### E. Documentation

The framework or engine documentation reduces the developer's learning curve and is an important factor to improve reusability and maintainability. The documentation is composed of several kinds of documents and contents such as wikis, code documentation, manuals and tutorials, example code and demos, books and additional third-party and community support.

Wikis help developers understand the tools' abstract –and most important– concepts. Manuals and tutorials teach "best practices", code examples and demos. An active community helps the developers with specific problems, issues and misunderstandings which can be used to further clarify the documentation and direct tool improvements.

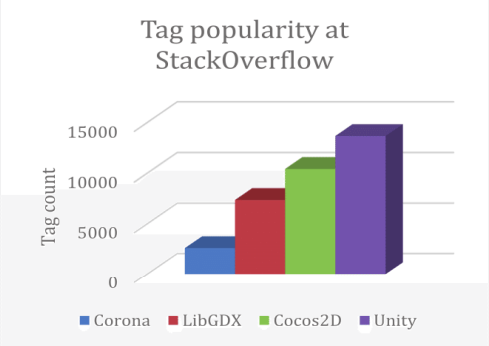



*Fig. 8. Tags popularity about some game development tools in StackOverflow community http://stackoverflow.com/*

Community size is closely related with the tools' efficiency, effectiveness, and growth potential. It is important to consider not only the documentation, but also the community behind the tool. The community's main goal is to help developers –indie or professional teams– resolve their unexpected problems and doubts. In addition, one of the key components of software engineering is reusability, and in this case it is important to consider the contributions of the other game developers that use the same tool. Unity has an official documentation composed of source-code documentation, feature manuals and tutorials exemplifying the creation of different types of games. Furthermore, Unity has a community with easy find tutorials and developers eager to help with solutions.

*StackOverflow* is a well-known developers' question-and-answer [42] with a large number of easily-searchable questions on game programming [43]. As of November 2015 there were +2500 tags about Corona framework, +7300 about libGDX, +10400 about Cocos2D, and +13600 about Unity, as illustrated in Fig. 8.

### F. Performance

Performance is another key factor considered by the developers when choosing a game development framework or engine. By using game development frameworks, developers can achieve better performance than a fully-featured game engine, since they generally do not provide so many canned game features. Some game development frameworks, such as libGDX, allow developers to access low-level programming APIs to squeeze the most performance from the underlying graphics technology – currently, either OpenGL or DirectX.

A good example of framework is libGDX, which focuses on avoiding garbage collection for Dalvik/JavaScript by careful API design and the use of custom collections. It also provides a single code base for all the platforms which greatly increases code maintainability (see Fig. 6). Additionally, it recommends using the Facade programming pattern to isolate platform-specific code, and provides utility methods to check the platform in which the code is being run.

The Unity AiO game engine leverages its GUI editor to allow developers to *drag'n'drop* game entities and components into the game scene, with the possibility of adding behaviours as C# or JavaScript code snippets. Unity also provides environmental variables to easily check the current platform, allowing the execution of platform-specific code. The GUI editor allows fast changes in the objects' positions or appearances, and uses the same code snippets for all the targeted platforms.

### G. Educational assessment

Most frameworks and engines do not consider educational assessment. However, the serious games created with these tools can and should use additional services or tools to enable educational assessments and learning analytics.

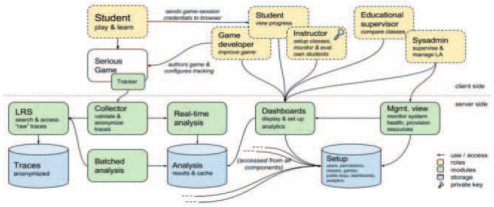



*Fig. 9. Learning Analytics architecture at RAGE project. http://rageproject.eu/*

Most game development frameworks and engines do not provide direct support for educational assessment. For example, libGDX has no direct support for any educational assessment mechanism. The developer must integrate any specific educational feature, which costs both time and resources. Developers using frameworks that lack the support for educational assessment mechanics should not develop the required features' infrastructure from scratch. As we can see in Fig. 9, developing a learning analytics infrastructure from scratch is an effort-consuming activity due to its complexity.

Instead it is recommended to integrate a service that provides a similar functionality out of the box. There are different analytics services available, each providing different features, and differentiated by target platform, cost, specific reports, data storage policies, and so on. Google Analytics is the de-facto industry standard for web and cloud-based analytics and dashboards. It provides a Rest API that can be used through all platforms but lacks official Software Development Kits on many platforms (e.g. Sony's PlayStation 4 or Microsoft's Xbox One). Google Analytics offers data visualization tools and the ability to define custom dashboards. Game Analytics is an alternative to Google Analytics that adds an extra layer of functionality and reports designed for games. Flurry Analytics offers similar features to Google Analytics, while eliminating certain thresholds. Although these analytics services were not created specifically for learning assessment, they can be adapted for this purpose.

Unity offers, in its Professional Edition license, the Unity Analytics Pro feature. Unity Analytics Pro gathers data from the game and transfers that data to cloud-based storage. The gathered data is processed, analyzed and sent to the Unity Analytics Dashboard. Like Google Analytics, it is very business-oriented, but can be adapted to provide a layer of learning analytics.

eAdventure was designed to create games with educational purposes in mind. It allows the developer to integrate specific educational features, such as adaptation profiles or evaluation profiles, within the game. Each profile is composed by a set of rules where a condition must be satisfied in order to trigger a series of effects. Developers can define different ways to evaluate or adapt the game in order to target the game to different kinds of contexts and public. Evaluation profiles can generate XML reports for automated processing or HTML reports for a game instructor or player. Adaptation profiles are executed when a set of external properties are satisfied, at the start of each game scene. An adaptation profile execution may change the content of the scene in terms of available objects, puzzle difficulty, etc.

Realising an Applied Gaming Eco-system (RAGE) is a project developing an open source infrastructure to deploy learning analytics and other educational assessment features. RAGE aims to develop, transform and enrich advanced technologies from the leisure games industry into self-contained gaming assets that support game studios at developing serious games easier, faster and more cost-effectively. The project assets will be available along with a large volume of high-quality knowledge resources through a self-sustainable ecosystem, which is a social space that connects research institutions, gaming industries, intermediaries, education providers, policy makers and end-users [REF]. One of these assets is the Learning Analytics asset, which allows teachers to monitor students and receive alerts and warnings depending on the student's game state, progress or score. The learning analytics asset provides a tracker client available in different programming languages, including JavaScript, Java and C#, which can be integrated with other frameworks and game engines such as libGDX and Unity.

## V. CONCLUSIONS

The rise in economic relevance of the game industry has increased the number and quality of tools to create games. There is a large game development community and a broad spectrum of authoring tools, some aimed at the creation of specific games, such as eAdventure, for *point'n'click* games, and others designed to create a much broader variety of game genres, such as Unity.

While there are tools to create a wide range of different games genres, serious games have special requirements and there is still no proven and widely-accepted approach to create an effective SG. The developer must choose the best options depending on the game requirements, and considering factors such as the target platforms, desired interactions and mechanics, game genre, learning objectives, time constraints, and resources and supported technologies. Therefore, there are many aspects to consider and some of the technical characteristics are frequently neglected.

In this paper we have introduced and analysed the main aspects that a developer has to consider before choosing a specific tool to develop a game, or before choosing to develop a game at all. A game development framework might be preferred in cases where the developers already know the required programming language and the game requirements are aligned with the framework's features. Frameworks are chosen over game engines by developers that want to learn or exercise the core principles of game development and by small work teams. Developers with high time constraints may choose a specialized authoring tool in order to optimize the time spent developing the game. *All in one* (AiO) authoring tools might be the best option for most developers since they balance the amount of provided features, learning curve and targeted game mechanics. Unity is probably the most influential AiO tool, excelling in many of its provided features, and is able to

compete with some *AAA* tools in some graphics features while, at the same time, being able to support multi-platform deployment and compete with game-development frameworks in terms of control over the game. However, Unity still requires programming.

Our analysis reflects that most of the available tools for creating games are conceived for the general game market and lack specific assessment features to analyse player behaviour or learning aspects. Only some specialized tools have these features (e.g. eAdventure) and those tools usually do not provide other relevant characteristics (e.g. multiplatform support).

Furthermore, the demand of SGs adapted to specific scenarios is growing, as we see teachers trying to integrate serious games within their classes and educational curricula. However, these teachers rarely have the programming skills that they would need to modify and adapt these games for different scenarios.

As final conclusion, one of the largest gaps in learning games is the lack of a software ecosystem that facilitates the creation of video games for educational purposes. There is also a lack of tools that can be easily integrated with the games in order to assess the learning outcomes through analytics while providing the main features listed in Section IV, such as exporting to several platforms, having a large quality community, efficiency and effectivity, etc. Through the different cases reviewed in this paper we have also identified the necessity of a software ecosystem that would facilitate the adaptation of serious games to specific scenarios and needs, such as allowing teachers to adapt existing games for their classes (using COTS) without programming knowledge. Some of those aspects are being currently addressed in the H2020 European project RAGE, which develops, transforms and enriches advanced technologies from the leisure games industry into self-contained gaming assets (modules) that support game studios at developing applied games easier, faster and more cost-effectively. A key and relevant asset is the entire infrastructure required to simplify the application of learning analytics in serious games.

## ACKNOWLEDGMENTS

The e-UCM research group has been partially funded by Regional Government of Madrid (eMadridS2013/ICE-2715), by the Complutense University of Madrid (GR3/14-921340), by the Ministry of Education (TIN2013-46149-C2-1-R), by the RIURE Network (CYTED 513RT0471) and by the European Commission (RAGE H2020-ICT-2014-1-644187 and BEACONING H2020-ICT-2015-687676).

## REFERENCES

[1] ESA, "Games: Improving the economy," pp. 7–9, 2015.

[2] "IGN - GTA 5 SALES HIT $1 BILLION IN THREE DAYS." [Online]. Available: http://www.ign.com/articles/2013/09/20/gta-5-sales-hit-1-billion-in-three-days.

[3] "Forbes - 'Grand Theft Auto V' Crosses $1B In Sales, Biggest Entertainment Launch In History." [Online]. Available: http://www.forbes.com/sites/erikkain/2013/09/20/grand-theft-auto-v-crosses-1b-in-sales-biggest-entertainment-launch-in-history/.

[4] S. Jarvis and S. De Freitas, "Evaluation of an Immersive Learning Programme to Support Triage Training," in *Games and Virtual Words for Serious Applications*, 2009, pp. 117–122.

[5] S. De Freitas, "Learning in Immersive worlds A review of game-based learning Prepared for the JISC e-Learning Programme," *JISC eLearning Innov.*, vol. 3.3, no. October 14, p. 73, 2006.

[6] E. A. Akl, V. F. Kairouz, and K. M. Sackett, "Educational games for health professionals," ... *Database Syst Rev*, 2013.

[7] E. Brox, L. Fernandez-Luque, and T. Tøllefsen, "Healthy Gaming – Video Game Design to promote Health," *Appl. Clin. Inform.*, vol. 2, no. 2, pp. 128–142, 2011.

[8] S. Arnab, I. Dunwell, and K. Debattista, *Serious Games for Healthcare: Applications and Implications*, vol. 2. 2013.

[9] J. C. R. Jr, P. J. Lynch, L. Cuddihy, D. A. Gentile, J. Klonsky, and R. Merrell, "The Impact of Video Games on Training Surgeons in the 21st Century," *Arch. Surg.*, vol. 142, no. 2, pp. 181–186, 2007.

[10] T. A. Pempek and S. L. Calvert, "Tipping the balance: use of advergames to promote consumption of nutritious foods and beverages by low-income African American children.," *Arch. Pediatr. Adolesc. Med.*, vol. 163, no. 7, pp. 633–7, 2009.

[11] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, and F. Players, "Predicting protein structures with a multiplayer online game," *Nature*, vol. 466, no. 7307, pp. 756–760, 2010.

[12] M. D. Dickey, "Engaging by design: How engagement strategies in popular computer and video games can inform instructional design," *Educ. Technol. Res. Dev.*, vol. 53, no. 2, pp. 67–83, 2005.

[13] J. Kirriemuir and A. McFarlane, "Literature Review in Games and learning - Futurelab," *futurelab*, 2004. [Online]. Available: http://www.futurelab.org.uk/resources/publications_reports_articles/literature_reviews/Literature_Review378/.

[14] D. R. Michael and S. L. Chen, *Serious Games: Games That Educate, Train, and Inform*, vol. October 31. 2005.

[15] C. Dede, "Immersive Interfaces for Engagement and Learning," *Science (80-. ).*, vol. 323, no. 5910, pp. 66–69, 2009.

[16] J. P. Gee, "What video games have to teach us about learning and literacy," *Comput. Entertain.*, vol. 1, no. 1, p. 20, 2003.

[17] T. M. Connolly, E. A. Boyle, E. Macarthur, T. Hainey, and J. M. Boyle, "Computers & Education A systematic literature review of empirical evidence on computer games and serious games," *Comput. Educ.*, vol. 59, no. 2, pp. 661–686, 2012.

[18] G. J. Hwang and P. H. Wu, "Advancements and trends in digital game-based learning research: A review of publications in selected journals from 2001 to 2010," *Br. J. Educ. Technol.*, vol. 43, no. 1, pp. 6–10, 2012.

[19] C. Perrotta, G. Featherstone, H. Aston, and E. Houghton, *Game-based learning: Latest evidence and future directions*. 2013.

[20] J. Chen, "Flow in games (and everything else)," *Commun. ACM*,

vol. 50, no. 4, p. 31, 2007.

[21] M. Mayo, "Video games: a route to large-scale STEM education?," *Science (80-. ).*, no. January, pp. 79–82, 2009.

[22] L. A. Annetta, J. Minogue, S. Y. Holmes, and M.-T. Cheng, "Investigating the impact of video games on high school students' engagement and learning about genetics," *Comput. Educ.*, vol. 53, no. 1, pp. 74–85, 2009.

[23] S. Barzilai and I. Blau, "Scaffolding game-based learning: Impact on learning achievements, perceived learning, and game experiences," *Comput. Educ.*, vol. 70, pp. 65–79, 2014.

[24] M. T. Cheng, T. Su, W. Y. Huang, and J. H. Chen, "An educational game for learning human immunology: What do students learn and how do they perceive?," *Br. J. Educ. Technol.*, vol. 45, no. 5, pp. 820–833, 2013.

[25] M. Papastergiou, "Exploring the potential of computer and video games for health and physical education: A literature review," *Comput. Educ.*, vol. 53, no. 3, pp. 603–622, 2009.

[26] H. Tüzün, M. Yilmaz-Soylu, T. Karakuş, Y. Inal, and G. Kizilkaya, "The effects of computer games on primary school students' achievement and motivation in geography learning," *Comput. Educ.*, vol. 52, no. 1, pp. 68–77, 2009.

[27] J. Blow, "Game Development: Harder than you think," *Queue*, vol. 1, no. 10, p. 28, 2004.

[28] P. Moreno-Ger, D. Burgos, I. Martínez-Ortiz, J. L. Sierra, and B. Fernández-Manjón, "Educational game design for online education," *Comput. Human Behav.*, vol. 24, no. 6, pp. 2530–2540, 2008.

[29] R. Sandford, M. Ulicsak, K. Facer, and T. Rudd, "Teaching with Games," *Comput. Educ. Educ. Gr.*, vol. 112, p. 12, 2006.

[30] R. Van Eck, "COTS in the classroom: A teacher's guide to integrating commercial off-the-shelf (COTS) games," *Handb. Res. Eff. Electron. gaming Educ.*, pp. 179–199, 2008.

[31] "ELTSandbox - videogames usage in classrooms." [Online]. Available: http://eltsandbox.weebly.com/blog/category/lessons with games.

[32] R. Van Eck, "Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless," *Educ. Rev.*, vol. 41, no. 2, pp. 16–30, 2006.

[33] "Gamelearn - Soft Skills Training Through Video Games." [Online]. Available: https://game-learn.com/.

[34] "Classcraft - Make Your Classes Unforgettable." [Online]. Available: http://www.classcraft.com/?utm_expid=68436248-15.WG4DGSkHTDqoAs056aTRRQ.0.

[35] "Duolingo - Learn a language for free. Forever." [Online]. Available: https://en.duolingo.com/.

[36] V. Jašková, "Department of English Language and Literature Duolingo As a New Language-Learning Website and Its Contribution To E-Learning Education," 2014.

[37] R. Vesselinov and J. Grego, "Duolingo Effectiveness Study," *City Univ. New York, USA*, no. December 2012, 2012.

[38] "Citizen Science - Play it online!" [Online]. Available: http://citizenscience.gameslearningsociety.org/node/23.

[39] "Scratch - Imagine, Program, Share." [Online]. Available: https://scratch.mit.edu/.

[40] M. Resnick, B. Silverman, Y. Kafai, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, and J. Silver, "Scratch," *Commun. ACM*, vol. 52, no. 11, p. 60, 2009.

[41] "e-Adventure e-Learning Games." [Online]. Available: http://e-adventure.e-ucm.es/.

[42] Muenchen, Robert A., "The Popularity of Data Analysis Software," no. April, 2014.

[43] C. Stanley and M. Byrne, "Predicting Tags for StackOverflow Posts," *Chil.Rice.Edu*, pp. 414–419, 2011.