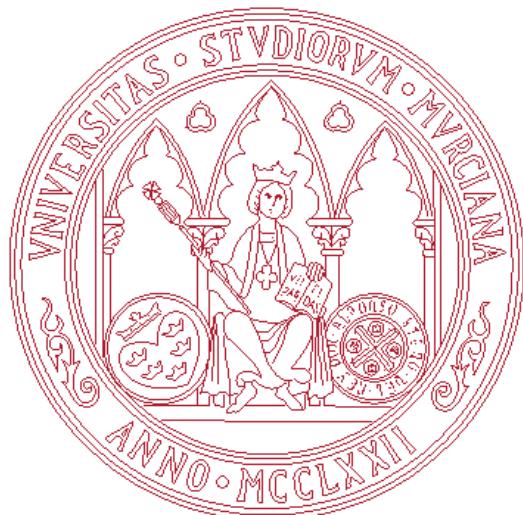


Trabajo fin de grado

Desarrollo de aplicaciones sobre el castillejo de Monteagudo con Unity

Grado en Ingeniería Informática



Alumno: Jose Miguel Sánchez Almagro

Tutor: Paco Guil

8 de septiembre de 2020

Declaración firmada sobre originalidad del trabajo

D./Dña. **Jose Miguel Sánchez Almagro**, con DNI **48839721A**, estudiante de la titulación de **Ingeniería Informática** de la Universidad de Murcia y autor del TF titulado **“Desarrollo de aplicaciones sobre el castillejo de Monteagudo con Unity”**.

De acuerdo con el Reglamento por el que se regulan los Trabajos Fin de Grado y de Fin de Máster en la Universidad de Murcia (aprobado C. de Gob. 30-04-2015, modificado 22-04-2016 y 28-09-2018), así como la normativa interna para la oferta, asignación, elaboración y defensa de los Trabajos Fin de Grado y Fin de Máster de las titulaciones impartidas en la Facultad de Informática de la Universidad de Murcia (aprobada en Junta de Facultad 27-11-2015)

DECLARO:

Que el Trabajo Fin de Grado presentado para su evaluación es original y de elaboración personal. Todas las fuentes utilizadas han sido debidamente citadas. Así mismo, declara que no incumple ningún contrato de confidencialidad, ni viola ningún derecho de propiedad intelectual e industrial

Murcia, a **8 de Septiembre de 2020**

Fdo.: Jose Miguel Sánchez Almagro
Autor del TFG

Agradecimientos

Al primero que debo agradecer la ayuda prestada en la realización de este trabajo es a mi tutor Paco Guil, por aceptar realizar este trabajo a pesar del numeroso número de trabajos que realiza cada año, y por todo el tiempo invertido para lograr tener el mejor resultado posible y resolver los problemas que yo no sabía solucionar.

Este trabajo cuenta con modelos 3D que fueron cedidos por el grupo de trabajo del CSIC que se encuentra actualmente trabajando en las ruinas arqueológicas del entorno del castillo de Monteagudo. Darles las gracias por la cesión del uso de los modelos a pesar de no haber podido realizar finalmente una colaboración como a ambas partes nos hubiera gustado.

Fuera del entorno académico, no puedo evitar acordarme de mis amigos y personas que me han arropado no solo durante el tiempo que estuve realizando este trabajo, sino desde que comencé los estudios de grado. Gracias por todos los momentos vividos que hicieron de esta una etapa que mereciera la pena vivir.

Y por último y a quien más debo, mi familia. Los que están ahí día a día, me hacen la vida más fácil, y por supuesto más feliz. Gracias a mis padres por todos esos consejos y apoyo recibido durante este tiempo. Sin ellos todo habría sido infinitamente más difícil.

Índice general

Resumen	4
Extended Abstract	5
Introducción	9
Castillejo de Monteagudo	9
Reyes y contexto histórico	10
Castillejo en la actualidad	10
Estado del arte	12
Visores virtuales	12
Trabajos fin de grado de videojuegos y juegos serios	18
Realidad aumentada	19
Análisis de objetivos y metodología	24
Análisis de objetivos	24
Planificación	26
Metodología	29
Realización final	29
Diseño y resolución del trabajo realizado	32
Software utilizado y su versión	32
Modelos 3D	34
No compilación	38
Interfaz de la aplicación	40
Otros elementos	44
Aplicación de realidad aumentada	44
Conclusiones y vías futuras	47

Resumen

En este trabajo fin de grado se han desarrollado dos aplicaciones entorno a una reconstrucción virtual de lo que podría haber sido el castillejo de Monteagudo en el pasado.

La primera y más importante de ellas consiste en una visita virtual al castillejo, utilizando un personaje en primera persona muy similar al estilo de muchos videojuegos, que recurren a este tipo de cámara y controles para sumergirte en la experiencia, siendo éste el principal motivo para nuestra elección de este control. Durante la visita, los usuarios podrán caminar libremente por los patios, salas y pasillos, como si de una visita real se tratara. Además, contarán con un panel lateral que le indicará en qué sala se encuentran en cada momento. Esta plataforma virtual se encuentra disponible para su uso en ordenadores con el sistema operativo Windows, para navegadores webs utilizando la tecnología HTML5, y para dispositivos móviles o tablets con Android.

Mientras que esta primera aplicación proporciona una visión interior del castillejo, la segunda se desarrolló para proporcionar otro punto de vista, haciendo así más completa la inmersión del usuario en la experiencia. Gracias a esta aplicación móvil los usuarios pueden observar el castillejo desde fuera. Para realizarla se ha utilizado un paquete de realidad aumentada en Unity, de manera que los usuarios pueden visualizar el castillejo como si lo tuvieran presente y en cualquier superficie plana: en una mesa, en el suelo, en la cama, etc. Además, el castillejo se fija una vez se instancia, por lo que se permite al usuario que está viendo el castillejo moverse alrededor de donde éste ha aparecido, pudiendo así observarlo desde cualquier ángulo y distancia, dando incluso la posibilidad de observar las mismas habitaciones que antes ha estado recorriendo con la otra aplicación.

El objetivo de estas aplicaciones es acercar la cultura e historia de la región a los murcianos o cualquier interesado, siendo además el castillejo una construcción prácticamente desconocida en la actualidad y de cierta importancia artística, como veremos más adelante.

Extended Abstract

In this paper, I will relate how was the develop of two Unity apps, focused on the discover of the castillejo de Monteagudo to everyone who is interested and wants to know more about the history and culture from Murcia.

This fortified palaced named castillejo de Monteagudo is in the district of Monteagudo, near the city of Murcia. It is part of the castle of Monteagudo and the constructions of the zone. Both constructions had their more important period with the muslim civilization, between the 9th and 12th centuries, when several kings used them as residence or defense zone. In fact, the coming christian kings kept using Montegudo's castle as residence, due to their beauty and importance.

Despite of that, most citizens of Murcia are not aware of the existence or the importance of castillejo, due to governments have ignored it during several years. This changed on the last years, when the local government of Murcia financed an investigation on the zone. Besides, there is actually a future project which will restore the ruins, creating a whole park around them, where people will be able to know the zone and their importance, being able to sped a nice day with family and friends thanks to the gardens which the project will have.

Thanks to this financing, researchers discover every day new things every day about what the area was like, the people who lived there, and the structure of the buildings, with the castillejo between them. Investigations have provided us a vision of how the castillejo was. In this degree, a 3D model of the castillejo is used, although it is an old version, due to nowadays the investigators know that several things have changed since this model was created. Knowing that, the centre of attention of this project must not be the historical accuracy, but the virtual immersion of the player, and a vision of how future apps can provide a view of how Monteagudo was several centuries ago.

The first one of both apps we are talking about can be used on computers or mobile devices, except Apple computers and devices. It consists on controlling a player, as if it were a videogame. Actually, the controls and the player's view are inspired by and used in video games. This can be difficult por people who is not used to videogames, who would prefer more basic controls or other type of movement. This is known by us, in fact, of having had more time, an automatic navigator would have been developed, with which users could have visited the room they wanted without move the player, only selecting it from a list o panel. Despite the known issues of the control system, it provides an immersion feeling, which is what we looked for in this project.

When the scene starts, the player appears on a patio, surrounded by a wall, formed by defensive towers. The entrance of the palace is with a door, which face to some stairs. After climbing them, we are finally in. The palace is empty on the sided, but with the information panel we can see the use of every room. This information is not

accurate and should only be taken as an orientation and immersion to the experience. The central and most important place is the inner patio. It is formed by four terrain pieces, which have common trees in Murcia: orange trees, lemon trees, pine trees and palm trees. At the end of the trees, the palace has two water tanks, which could be used to water the trees. In them, we have introduced fishes, who are decorative and give an animated view of the zone. Besides, we also added swallows on the sky. The opened rooms of the patio are the most beautiful of all the palace, having coloured and ornamented arches, and columns who support them. Around the palace, the player can see the external terrain thanks to windows on some rooms. The music which sounds in all the visit is also arabic.

So that users can run the program on any device, the game has a configuration menu, which you can access from the pause menu. It has different levels of quality. The lowest level disables animals, particles, post process, shadows and general quality of textures. The next level only enables the postprocess, because it is very demanding to the hardware and the most notable quality improve. We enable things until arrive to the top quality, which has everything to the maximum level of quality and changes the 3D model. The standard model has fake ornamentation on the arches, lowering the hardware demand but lowering too the quality of the palace. With this new model, all the ornamentation is real.

For using the app on mobile devices as smartphones or tablets, the app provides an adapted interface to the users. It has a joystick on the lower left corner of the screen, which allows to move the player on the castle. To move the sight of the character the user can touch the right side of the screen, acting that like the mouse on computers. Besides, the user can open the information panel with a red button, and all the functionality is adapted for touch screens.

For help the users to use the app, help panels are shown at the start to the visit and when it is necessary, like for entering to the palace.

Another add-on included on the project was the possibility of changing the 3D model of the castle after building the project. This consists in creating or modifying the 3D model in Blender or another software, export it in .obj format, and include it in a specific folder of the project. If it is done, when the visit starts the model of the castle changes without the users noticing it. Thanks to that, investigators can keep developing and changing their model and include it in the app, without using Unity or another software. In the final version of the project I disabled the option because it is not going to be used for the moment. Besides, there are right now two models in the project, the real and the fake, so if there are other included changes must be made on the configuration script. However, the asset for importing the .obj archives and the script which loads them is done, so with some changes it can be used.

About what I found more difficult, I would say the first person controller, and I will explain why. The controller used is the one included on the Standard Assets of

Unity, because it works well, it is free, and have several options, like steps sounds. It worked on desktop builds, but when I added a joystick for moving the character on mobile devices, I had to make it compatible with the first person controller used for the desktop version. After a time trying to import different joysticks, prefabs, and read their scripts, I realised how to make it work, modifying the fps controller script. Nevertheless, the scripts that the fps controller used had to be moved to my personal scripts folder on the project, instead of the folder included with the standard assets of unity. This is because Unity has an order for execute the scripts, independently the function used (Start() or Awake()), and the scripts of the standard assets are executed before than all the others, what caused a problem, because I needed to compile before the joystick scripts. After to do everything I said, the first joystick assets imported worked until one day, which stopped to work. Then, I needed to import another joystick and make it compatible with the fps controller again. As I explained, the process for include a joystick was not really difficult, but annoying.

Another functionality that I thought of including was an animated background for the menu. This is used in some videogames, one example of them is Minecraft, which has a faded background that look over an aleatory terrain slowly. The idea was to do something similar, without fading and with different cameras, providing a different view each. Actually, a first script was made, and worked, calculating the difference between two points and advancing the camera between them. But there were two problems. The first one was with the script. The position movement was done and worked perfectly, but not the rotation, which was a complicated movement to programme because Unity uses Quaternions for representing the rotation, so I needed a mathematical base to do it, and more time obviously. The other problem was the performance. If loading the visit scene took some time, it would be the same with the main menu, making it uncomfortable for the users. Due to that and the Quaternions problem I decided to drop the idea and focus in other things.

About the 3D model, it is not perfect and had a lot of fails. The first one, textures was not assigned, so the model had the basic colours. They had to be manually assigned on Unity, and it would be a problem for import the model without building it with the project. The other problems are more difficult to fix, and the most of them are related to the normals of the model. Because of many mistakes, normals were not correctly, so it became a problem for seeing the model correctly. Francisco de Asís fixed a lot of errors found on the model, but there was too much to fix, and there are still bad normals and visual errors, which the user can see. Another problem of the model is the size or the precision of the models, because in some points, the models does not fit exactly between them. There are four independent models: the interior palace, the stairs, the fortress, and the terrain, and if the character looks on certain points, it will see fit problems. In fact, when the player is on the exterior patio, the terrain is disabled to avoid some flaws.

I will talk now about the augmented reality (AR) app. It was initially thought for giving the users the possibility of seeing the fortified palace from the outside,

considering that they can only see the interior with the previous app. This is, therefore, a secondary app, supporting the first one, and which would be useless without the other. For develop this app, we used the AR Fundation package, available in Unity. This platform unifies in a common interface the most used AR platforms for developers for Android and iOS apps: ArKit and ARCore.

The app has two options for seeing the castle. One uses the plane detection AR technique, the other uses image detection. Normally, the users will not have the image on a paper for seeing the castillejo, so they will use the option with planes detection. The performance of that is dependent on the AR Fundation package, and it is not the best of all, but acceptable. In addition of seeing the 3D model of the castillejo, we will also see the trees that we introduced manually on Unity.

Both apps are a good result and allow to the users to know the castillejo structurally, nevertheless, there is a lack of interaction with the user which would give more possibilities and enterntainment for use the apps. Theses apps work good for a first impression of how could be a future app which integrate more functionality, and would have possibly some games, a kids mode, characters, or even a virtual reality mode. All of theses things need time, but also need a base, which can be this project. However, there can be another vision of what an application for the castillejo can be, because, if finally, there is a theme park oriented to know the zone, an enterntainment app could not be the best option. In that case, would be better include more information, possibly a virtual museum, and other type of controls. There is a whole world of possibilities, and the apps developed in this thesis are only a base, a starting point of what this could be in a future.

Introducción

El propósito inicial de este trabajo era realizar una colaboración con el grupo de arqueólogos del CSIC que se encuentran actualmente en las ruinas arqueológicas de la zona del castillo de Monteagudo. Sin embargo, por diversas circunstancias no se ha podido realizar dicha colaboración, y he realizado este trabajo en solitario junto a mi tutor Paco Guil. Pese a ello, han tenido la amabilidad de cederme una reconstrucción en tres dimensiones de una versión desactualizada del castillejo.

El trabajo se reorientó entonces, eliminando del mismo la rigurosidad histórica que iba a caracterizar al primero, junto a su inicial objetivo de ser una herramienta de exposición para el trabajo de los arqueólogos, a ser un resultado audovisual más propio de un videojuego. Esto, además, trajo consecuencias durante el desarrollo del trabajo, tal y como se indicará más adelante.

No por ello hay que olvidar el significado del castillejo y su importancia artística, siendo el centro de la investigación arqueológica que se está realizando actualmente.

Castillejo de Monteagudo

El castillejo es un Castillo menor en dimensiones y con distinto propósito que el de Monteagudo, también visto como un palacio fortificado, que sería un término más apropiado a mi parecer dada la naturaleza y forma de ser del mismo. Formado por un patio central que recuerda al patio de los leones de la Alhambra de Granada, destaca por la numerosa cantidad de torres que cuenta en su estructura, dándole al mismo forma de muralla defensiva.

El interior, por otro lado, se asemeja al de un palacio. De ahí que me incline por la utilización de la denominación como palacio fortificado. Cuenta con dos largos pasillos a los lados de la estructura, que cuenta con varias salas rectangulares. En las esquinas, dichas salas dan a espacios que no se encuentran reflejados en el modelo, y podrían ser escaleras que ascendieran a las esquinas de las torres. Todo el castillejo está decorado con torres y arcos con decoración ornamental.

El patio es la zona con más personalidad del castillejo, y el lugar que le da sentido al mismo, haciendo las veces de jardín para la relajación o retiro espiritual de los reyes que habituaban el lugar. Sobre el mismo, podemos decir se encuentra dividido en cuatro zonas de césped separadas entre si por caminos de piedra o mármol, que se unen en un punto central. En los lados norte y sur del patio encontramos dos albercas que contenían agua, y podían ser utilizadas para el riego de la vegetación. Una vez pasadas las albercas encontramos unas salas decoradas con arcos, teniendo estos con la mayor decoración ornamental de todo el castillejo, aumentando así la carga artística del mismo. Estas salas, orientadas hacia el patio, otorgaban unas vistas privilegiadas

del mismo, sirviendo de lugar de reuniones o retiro de los reyes y su corte.

Reyes y contexto histórico

Para entender la importancia del castillejo debemos remontarnos a la fundación de la ciudad de Murcia tal y como la conocemos hoy en día.

En el año 825, el emir de al-Ándalus Abderramán II mandó fundar la llamada ciudad Medina Mursiya sobre una posible civilización anterior. Aunque no hay consenso entre los historiadores, todo apunta a que se trataba de una asentamiento romano llamado Murtia, establecido en la zona por la abundancia de agua y riqueza del territorio [1].

Tras esto, llegó la primera taifa de Murcia, al principio con independencia pero luego anexionada a la taifa de Sevilla. Tras esto, la debilidad de las taifas incentivó la reconquista por parte de los reinos cristianos, lo que obligó a la taifa de Sevilla a pedir ayuda a los almorávides, que frenaron el avance cristiano y unificaron todo el territorio musulmán. Tras esto, los almohades aprovechan la debilidad de los almorávides para hacerse con la mitad del territorio que los mismos ostentaban en la península, haciéndose fuerte los almohades en el oeste y los almorávides en el este [2].

Fue entonces, cuando Murcia se convirtió en el centro de los territorios almorávides, siendo el lugar habitual de residencia del rey lobo (Muhammad Ibn Mardanis). Es este el período de mayor prosperidad de la historia de la ciudad murciana. Los musulmanes aprovecharon la fertilidad de la tierra y la abundancia de agua de aquel entonces para desarrollar modernos sistemas de riego y agricultura. Además, el rey lobo convirtió el castillo y el castillejo de Monteagudo en su residencia habitual, siendo la zona protagonista de varias edificaciones entre las que destacan el castillo de Larache, una gran alberca que hacía las veces de lago, y más. Estos años de esplendor fueron sin embargo breves, desde 1147 hasta 1172, año de muerte del rey lobo, que le sucedió la posterior ocupación de los almohades del territorio almorávide que aún quedaba presente en la península, y la destrucción de edificaciones como el castillejo.

Tras la reconquista de Murcia, el rey Alfonso X establecería su lugar de estancia en Murcia en el castillo de Monteagudo [3].

Castillejo en la actualidad

Tras el abandono del castillo de Monteagudo y dicha zona, señores ocuparían los terrenos de la ladera del castillo, siendo además las construcciones que habían destruidas, abandonadas, o sepultadas, hasta en el siglo XX, se comenzara a valorar la riqueza histórica del lugar, empezando una concienciación para la protección de los

restos que aún quedaban. Actualmente, un grupo de arqueólogos investiga en la zona cuando el Ayuntamiento de Murcia financia dichas excavaciones, produciéndose cada día nuevos descubrimientos que cambian la percepción que teníamos del lugar y aumentando nuestro conocimiento.

Entre dichos descubrimientos, se encuentra la maduración del castillejo, que aún hoy día sigue en desarrollo, siendo la versión que se aprecia en este trabajo anterior a los descubrimientos más recientes.

Estado del arte

Se han desarrollado numerosas plataformas para recorrer lugares virtualmente, desde recorrer virtualmente museos de todo el mundo hasta el mismo Google Street View, que nos permite situarnos visualmente en cualquier calle del mundo y verla como si estuviéramos físicamente allí. Cada una de estas plataformas tiene un propósito y forma de ser diferente, formando todas ellas un amplio abanico de posibilidades en lo que a este ámbito se refiere.

En este apartado no solo vamos a hacer un recorrido por las mismas, también analizaremos otros trabajos fin de grado sobre videojuegos, juegos serios o incluso turismo, que nos darán una visión sobre lo que se ha realizado en otros casos en situaciones similares.

Visores virtuales

Comenzaremos hablando de los visores que tenemos en el mercado y podemos encontrar con una simple búsqueda en internet. El más conocido es el anteriormente mencionado Google Street Maps. Con este visor podemos visualizar calles u otros lugares a los que una cámara haya llegado para fotografiar el lugar. Y este matiz es el que marcará una importante diferencia en lo que a visores se refiere. Si utilizan imágenes reales o no. En caso afirmativo, la experiencia de usuario y la razón de ser del visor cambia sobremanera al caso que tratamos en este trabajo. Cuando se realiza la reconstrucción de un lugar con imágenes, el usuario puede observar el lugar que visita con un nivel de detalle y realismo que ninguna otra plataforma puede lograr, siendo ésta su principal baza. Sin embargo, utilizar imágenes en lugar de objetos virtuales limita la experiencia de usuario, siendo esta arcaica en lo que a libertad de movimiento se refiere.

En la siguiente imagen, podemos observar una instantánea de la plaza Cardenal Belluga utilizando el servicio de Street View. Estando en dicho punto, lo que el usuario puede realizar es moverse 360 grados sobre si mismo, observando la plaza desde dicho punto. Si quiere moverse debe realizarlo pulsando en cualquier lugar al que quiera avanzar. Esto provocará que el usuario de un salto de unos metros, cambiando instantáneamente la posición de donde se encuentra y las imágenes de alrededor, pudiendo volver a moverse siguiendo el mismo procedimiento, ahora desde otro punto.



Figura 1: Plaza del Cardenal Belluga con Google Street View.

Esta manera de moverse por el entorno es eficaz, ya que consigue su propósito de una manera sencilla, pero no deja de ser arcaica y torpe, estropeando la experiencia de usuario. Sin embargo, es todo a lo que se puede aspirar actualmente cuando reconstruimos un lugar con imágenes. Y esta misma técnica la encontramos en muchos lugares más, repitiendo el mismo modo de funcionamiento, con sus matices en cada plataforma.

Otro ejemplo de este tipo de visores es la visita virtual que podemos realizar al castillo de Peracense [4], en la Comunidad de Aragón. En este caso, el usuario puede incluso utilizar unas gafas de realidad virtual, como las Google Cardboard, y observar las imágenes moviendo la cabeza como si se encontrara en el lugar, proporcionando una mayor experiencia de inclusión ambiental. Por supuesto, también podemos movernos con el ratón en el caso de no disponer de un visor de realidad virtual.

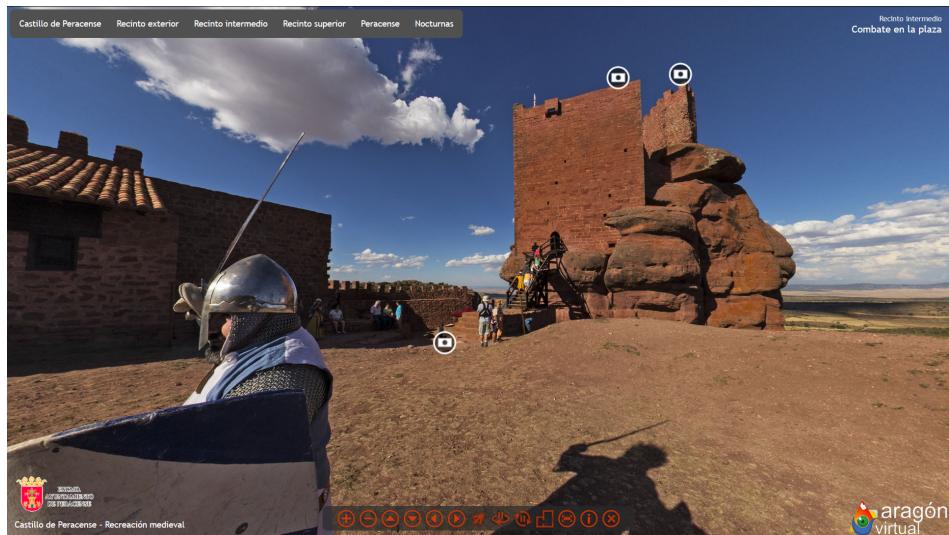


Figura 2: Recinto intermedio del castillo de Peracense.

Además, la plataforma incorpora otros elementos que facilitan la experiencia al usuario. El mismo puede por ejemplo visualizar un plano del castillo para elegir donde situarse, ya que, en esta plataforma, a diferencia de como ocurría con Street Maps, el usuario no puede moverse "libremente" por el entorno, puede ir a ciertos lugares específicos del mapa que se encuentran marcados con una cámara dentro de un círculo. Esto, que a priori puede parecer un inconveniente, permitió que cuando se tomaron las fotografías para el visor, se aprovechara y se recrearan situaciones de la época, con personas ataviadas con la vestimenta de la misma. Además, si el usuario no utiliza los controles durante cierto tiempo, la imagen comienza a girar horizontalmente, siendo éste un añadido que mejora la experiencia final.

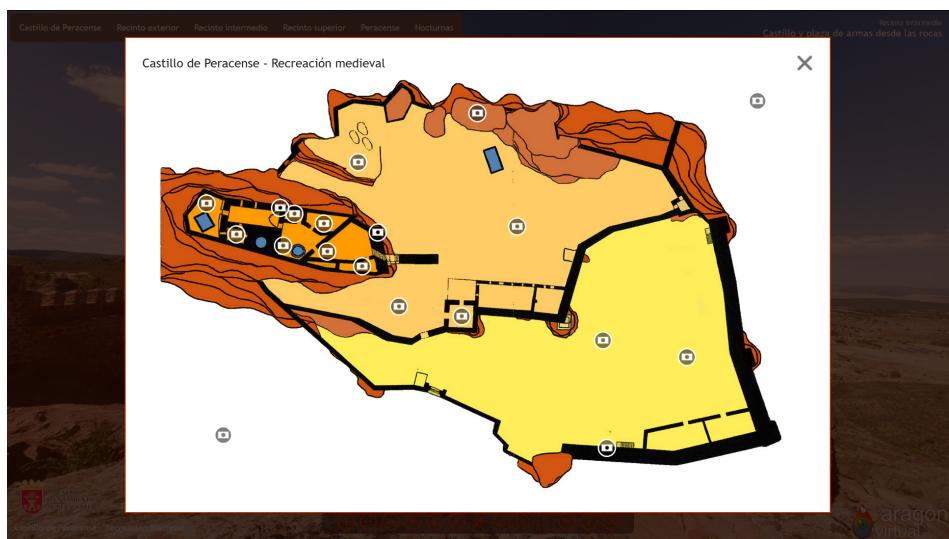


Figura 3: Plano del castillo de Peracense.

Al comienzo del apartado hablábamos de Street View, plataforma desarrollada y

mantenida por Google. No se trata de la única que han creado con un funcionamiento similar.

En *Arts and culture* [5] podemos realizar visitas a distintos lugares muy turísticos del globo. Entre ellos encontramos el palacio de Versalles o el museo británico.

En cada visita tenemos multitud de actividades y opciones. Entre ellas, visitas virtuales a estancias, jardines u otras estancias de los lugares mencionados. Podríamos, por ejemplo, realizar una visita virtual al museo británico, la cual se vería de la siguiente manera:

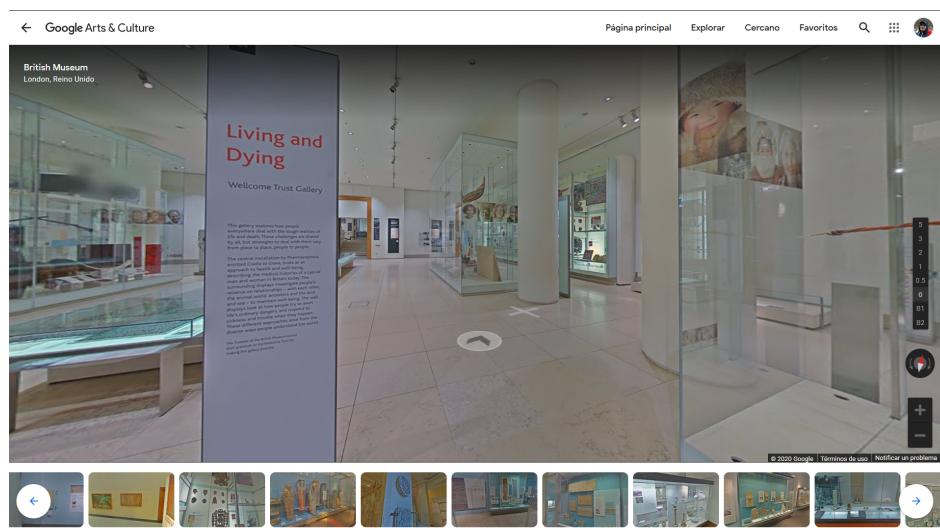


Figura 4: Visita virtual al museo británico.

En la parte inferior vemos lugares destacados del piso que trata de la cultura egipcia. En la parte derecha podemos escoger el piso al que queremos ir, y el resto de la interfaz es muy similar a lo que encontrábamos en Street View. De hecho, la forma de moverse por el entorno es igual. La tecnología que hay detrás de ambos visores es la misma.

Otro ejemplo español de visita virtual a otro castillo es el caso del castillo de Mirabel, en Extremadura. La similitud con el castillo de Peracense es total, con sus diferencias en la interfaz y el añadido de que en este caso, además de círculos rojos que nos permiten cambiar de estancia, encontramos etiquetas de información que despliegan una ventana con un texto explicativo.

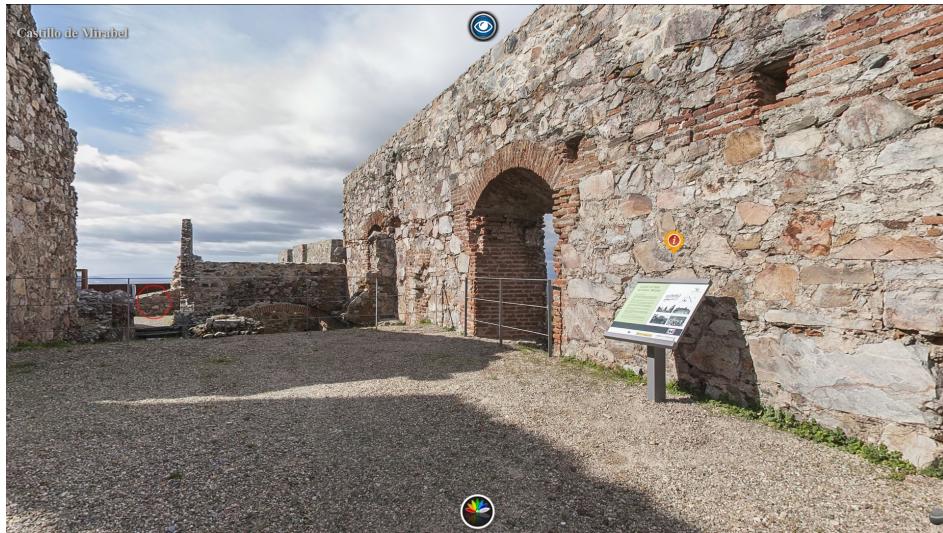


Figura 5: Instantánea del castillo de Mirabel.

Podríamos seguir nombrando ejemplos de visitas similares, pero en el caso de las reconstrucciones con fotografías todos los casos son prácticamente iguales. Nos cabe ahora preguntarnos que ocurre en el caso de las visitas en entornos virtuales, es decir, sin fotografías, con objetos en 3D. La visita que se desarrollará en este trabajo será de este tipo.

El Castillo de Falaise, si bien no es una visita virtual al uso, tiene elementos en común con este trabajo. Este castillo francés ofrece visitas guiadas a aquellos que quieran recorrer el castillo, tanto por el exterior como por el interior, y con la visita se le proporciona a los usuarios una tablet con la que pueden aumentar las posibilidades de su visita [6]. En la misma, situándote en cada sala, puedes disfrutar de una reconstrucción de la misma, sobreponiendo objetos 3D a lo que capta la cámara del dispositivo. En la siguiente imagen se puede apreciar la diferencia entre la realidad y lo que los visitantes ven con la tablet.



Figura 6: Instantánea de la visita al castillo de Falaise.

Cuando Paco y yo planteamos el trabajo en un principio, pensamos en realizar algo similar utilizando la tecnología de realidad aumentada de la que hablaré más adelante. Teníamos dudas sobre si podríamos realizarlo debido a que los investigadores no veían adecuado colocar marcadores en recinto donde se encuentran los restos del castillejo, e intentarlo con geolocalización podía no resultar viable, ya que tampoco sabíamos cuál sería la precisión del GPS. Sin embargo, esta posibilidad se descartó cuando la colaboración con los arqueólogos llegó a su final, ya que no se realizó ninguna visita a los restos del palacio y tampoco el futuro proyecto iba a estar ligado con ellos. A pesar de ello, debido a nuestra intención de utilizar la tecnología de realidad aumentada se acabó desarrollando una aplicación que hace uso de la misma, pero a eso llegaremos más adelante.

Descartando por tanto la posibilidad de desarrollar una aplicación que aprovechara una visita real al castillejo, apostamos por una visita completamente virtual, que el usuario pudiera disfrutar desde casa, o desde cualquier lugar con la aplicación móvil. Continué entonces mi búsqueda con castillos que tuvieran una reconstrucción 3D, en la que el usuario pudiera moverse por la misma. Solamente pude encontrar reconstrucciones virtuales en vídeos. Uno de ellos es esta visita virtual al castillo de Alcalá. Si se reproduce se puede apreciar que se trata de una reconstrucción propia de un videojuego. Sin embargo, en este otro ejemplo, la imagen es más parecida a unos dibujos animados que a unos modelos 3D de un videojuego.

En nuestro caso, el objetivo es poder realizar una visita a un entorno en 3D, es decir, completamente virtual, pero permitiendo al usuario libertad total de movimientos. Debido a que no estaremos tratando con imágenes no tendremos las limitaciones de avanzar solamente a ciertos puntos y dando "saltos", como ocurría en los primeros ejemplos.

Por lo tanto, lo que se quiere lograr es más propio de un videojuego que de una

visita como las que hemos visto hasta ahora. Por lo tanto, se podría decir que estamos hablando de un videojuego o juego serio. A continuación, se van a explorar trabajos fin de grado de la siguiente temática.

Trabajos fin de grado de videojuegos y juegos serios

Con una simple búsqueda en Youtube se pueden encontrar diversos trabajos fin de grado que ha consistido en el desarrollo de un videojuego. Los aspectos que nos interesan de dichos trabajos son: el motor utilizado, la interfaz, otros aspectos gráficos, y otros.

Los videojuegos tienen otros aspectos que de momento no trataremos como son la línea argumental, personajes, música, entre otros. Lo que no significa que se descarte la posibilidad de utilizarlos, si no que no son los puntos más importantes de este proyecto.

El alumno de la universidad de Jaén, Alejandro Gómez López, desarrolló el videojuego Flame Knights Chronicles como trabajo fin de grado [7]. En su apartado en el que analiza tres motores gráficos a utilizar (Unity, Unreal Engine y CryEngine) crea una tabla en la que se resumen las características más importantes de un motor de videojuegos, lo cual puede resultar útil en nuestro caso:

Comparativa Motores de Videojuegos	Unreal Engine 4	CryEngine 3	Unity 3D
Curva de Aprendizaje	Media	Difícil	Fácil
Calidad Gráfica	Muy Alta	Alta	Media
Características de modelado	Sí	Sí	No
3D y construcción de escenas avanzada			
Apto para equipos pequeños	No	No	Sí
Basado en componentes	Sí	No	Sí
Calidad del Motor de físicas	Muy Bueno	Excelente	Bueno
Gratis al inicio del proyecto	Sí	Sólo licencias para estudiantes	Sí
Gran biblioteca de recursos	No	No	Sí
Comunidad extensa	en Aumento	No	Sí
Documentación propia	Excelente	Buena	Excelente
Tutoriales de terceros	Sí	No	Sí, muchos

Tabla 1 Comparativa Motores de Videojuegos

Figura 7: Tabla de motores de videojuegos.

Este trabajo es de 2016, por lo que ciertos aspectos podrían haber cambiado. Sin embargo, la mayoría de aspectos siguen siendo igual actualmente que entonces, y estoy de acuerdo con su conclusión de que Unity es la mejor opción en nuestro caso.

Unity se trata de una plataforma de desarrollo de videojuegos orientada al sector móvil o al desarrollo de videojuegos que no sean grandes proyectos. Destaca por su

facilidad de uso, la cantidad de soporte que goza en internet, y otros aspectos como su eficiencia en dispositivos móviles o incluso navegadores, siendo esto último algo muy importante en nuestro proyecto.

Además de estos tres, hay muchos motores de videojuegos en el mercado. Uno de reciente emerión es Godot, un motor Open Source con compatibilidad con numerosos lenguajes de programación y muchas virtudes. Siendo un motor no comercial (al contrario de los tres mencionados anteriormente) es capaz de conseguir resultados muy profesionales que no quedan lejos de las opciones comerciales.

Sin embargo, aún se encuentra en fase de desarrollo, no cuenta con tanto soporte como Unity, y su eficiencia podría no ser tan buena.

Además, Unity es el motor utilizado en la facultad por varias asignaturas desde primero.

En este trabajo encontramos también un apartado en el que se trata la historia que tendrá el videojuego. Este es un aspecto que podría ser importante en nuestro caso, ya que explicar o exponer al usuario la historia sobre el castillejo o la zona que lo rodea podría ser esencial para que valorara y tuviera contexto sobre lo que está visitando.

En este caso, la manera de mover el personaje en el videojuego es igual a Street View y los otros ejemplos anteriores: pulsando en distintos puntos de la pantalla. Ya hemos descartado el uso de este tipo de movimientos por ser menos intuitivos, pero su uso en un videojuego nos indica la gran aceptación de este tipo de mecánicas.

El aspecto de la interfaz difiere mucho de lo que se busca en este trabajo, siendo muy centrada en la temática del videojuego. En nuestro caso, buscaremos una interfaz limpia y no intrusiva en la experiencia del usuario.

El resto de los aspectos del videojuego no son importantes en nuestro caso.

Una vez hemos analizado un trabajo que consistía en el desarrollo de un videojuego de entretenimiento, pondremos ahora el foto en la realidad aumentada.

Realidad aumentada

La realidad aumentada (AR) es una tecnología nueva por lo poco extendida que se encuentra hoy en día, sin embargo, no se trata de una tecnología de creación reciente. Esta fue desarrollada hace muchos años, y ha ido implementándose en diferentes lugares poco a poco. A pesar de ello, sigue siendo una gran desconocida para la mayoría de la población, habiendo conseguido la realidad aumentada el liderazgo en la carrea de ambas tecnologías.

En primer lugar, voy a hablar brevemente de ella para quien no la conozca o no se sienta familiarizado con ella. Además, hablaré un poco de historia y de los tipos que tenemos actualmente en el mercado.

La realidad aumentada es una tecnología que capta imágenes del exterior a través de una cámara y sobrepone en la imagen captada objetos virtuales. Usualmente se utiliza en dispositivos móviles, ya que todos disponen actualmente de una cámara y libertad para poder explotarla, al contrario que por ejemplo un ordenador, ya que éstos se encuentran fijos en una posición. Esta tecnología ha ido formando parte de nuestra vida poco a poco sin ni siquiera saberlo. De hecho, antes de la irrupción de los smartphones en el mercado, cuando solo teníamos videoconsolas portátiles, tuvimos el lanzamiento de un videojuego con sello español que utilizaba esta tecnología. Estoy hablando de Invizimals. Este videojuego lanzado por la compañía española Novarama para la consola PSP (Play Station Portable) fue todo un bombazo en el momento de su anuncio en el E3 de Los Ángeles, ya que por aquel entonces (2009) no se había explotado esta tecnología, y fue toda una sorpresa [8].

El siguiente gran lanzamiento aunque desconocido ya que no llegó a la masa de la población, fue Ingress.

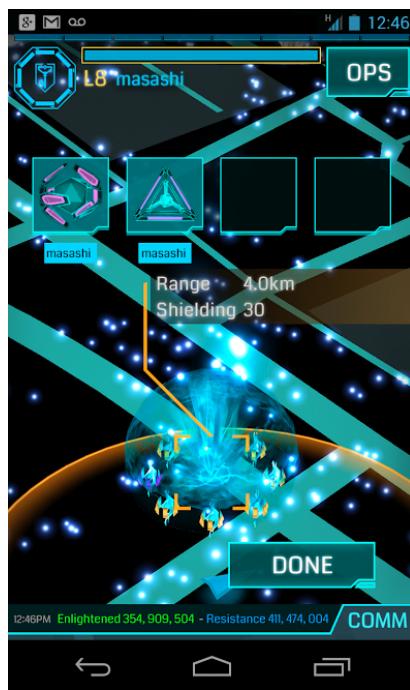


Figura 8: Captura de Ingress.

Este sencillo videojuego, consistía en la conquista y descubriendo de distintos “checkpoints” que encontrabas cada ciertos kilómetros en pueblos y zonas urbanas. De la mano de Google, captó el interés de desarrolladores y curiosos de la tecnología, yo entre ellos, ansiosos de ver el siguiente lanzamiento que aprovechara esta tecnología. Ese gran lanzamiento fue el ya conocido mundialmente Pokemon Go.



Figura 9: Logo de Ingress.

El videojuego de Nintendo seguía una mecánica muy similar al de Google, teniendo gimnasios pokémon y paradas pokémon donde antes había *checkpoints* en el videojuego de Ingress. La novedad que introdujeron fue el ecosistema Pokemon añadido a la posibilidad de capturar estas criaturas con el teléfono cuando aparecían aleatoriamente en el mapa. La captura de dichos pokemon podía realizarse de manera virtual o utilizando también realidad aumentada. Eso si, no todos los móviles eran compatibles con esta posibilidad, y eso es algo que trataré más adelante.



Figura 10: Captura de un pokémon con Realidad Aumentada a la izquierda y sin ella a la derecha.

Esta fue la gran explosión de realidad aumentada, ya que todos los fans de pokémon y jóvenes se lanzaron a las calles en pleno verano a capturar pokemons, sintiéndose auténticos entrenadores pokemon. Sin embargo, mientras Google lanzaba su videojuego Ingress a finales de 2012, llevaba a cabo una investigación y equipo de desarrollo del otro gran nombrado de realidad aumentada de aquellos años: las Google Glass.



Figura 11: Primera versión de las Google Glass.

Estas serían unas gafas cualesquiera, con graduación o sin ella, que llevarían una pantalla incorporada en sus cristales y un pequeño dispositivo en una pata, que haría las veces de cerebro, incluyendo la cámara y un pequeño panel táctil para controlarlas. Con estas gafas, los usuarios que las llevaran podrían ver información sobre lo que ven en ese preciso instante, como el nombre de ciertos lugares, la ruta para llegar a una ubicación, el tiempo en ese instante, o incluso grabar aquello que el usuario está viendo.



Figura 12: Imagen de la realidad aumentada de Google Glass en su vídeo presentación.

Esto último trajo consigo una serie de problemas de violación de la privacidad cuando las gafas salieron a la venta de manera cerrada, con unas pocas unidades y un alto precio. Esto provocó un rechazo general a las gafas y los usuarios que las utilizaban, y las sumergieron en una etapa de oscuridad en las que quedaron relegadas, y durante el cual Google limitó el número de anuncios y noticias que lanzaba sobre ellas. Desde entonces, Google ha focalizado sus gafas al mercado empresarial, saliendo además numerosas alternativas de distintos actores en el mercado [9]. Actualmente, Apple podría estar cerca de presentar las suyas.

Pero entonces, cabe preguntarnos. Si esta tecnología parece ya tan avanzada y aceptada, ¿por qué sigue siendo una desconocida para nosotros y no ha tenido el uso esperado? Esta sea quizá una pregunta difícil de responder, pero el principal motivo sigue siendo sorprendentemente el elevado coste hardware para utilizala. Actualmente, para utilizar el paquete de realidad aumentada AR Fundation disponible en Unity los dispositivos móviles, éstos deben ser compatibles mediante hardware con dicho paquete. Esto es una barrera inicial a aplicaciones que se desarrollan con dicha tecnología, ya que no todos los usuarios podrán disfrutar de ellas. Y ese es el caso de este trabajo, ya que la aplicación de realidad aumentada ya comentada anteriormente utiliza dicho paquete, no pudiendo ser utilizada por tanto en aquellos dispositivos que no lo soportan. Esto, unido a una falta de interés de los usuarios a la realidad virtual y aumentada ya que no se ofrecen productos que cumplan la expectativa del público, ha provocado el lento avance de ambas tecnologías.

Análisis de objetivos y metodología

Tras analizar distintas tecnologías y plataformas, ha llegado el momento de plasmar y detallar lo que se realizará en este visor del castillejo de Monteagudo.

El mismo estará formado por dos distintas aplicaciones. Una de ellas, consistirá en realizar una visita virtual como si de un videojuego se tratara al castillejo, introduciendo así al usuario en lo que dicho palacio tiempo atrás. La otra aplicación, con una orientación al sector móvil, permitirá al usuario visualizar el castillejo como si en su propia mesa se encontrara. Gracias a técnicas de realidad aumentada podrá escudriñar el castillejo con su teléfono móvil y desde la posición que desee hacerlo.

Análisis de objetivos

La primera aplicación de la que hablaremos será el visor virtual del castillejo, y la primera cuestión a resolver es cómo se moverá el usuario por el castillejo. Anteriormente se ha tratado esta cuestión, llegando a la conclusión de que lo que proporciona una mejor experiencia ya que estamos hablando de algo más cercano a un videojuego o juego serio, era usar un movimiento continuo, sin seleccionar puntos en la pantalla como ocurría en Street View. Introduciremos por tanto un personaje que representará al usuario, y que tendrá un control de los que se utilizan en videojuegos. En ellos, encontramos diversos tipos, siendo los más importantes el movimiento en tercera persona y en primera. La ventaja del primero es que el jugador puede ver al personaje mientras juega y tener una visión más amplia, dando así una mayor importancia a la estrategia. En la vista de primera persona, el jugador se sitúa en el personaje, viendo lo que este ve. La ventaja de este tipo de controles es la inmersión en el videojuego, logrando una mejor experiencia. Este es el control usualmente usado en shooters como Counter Strike o Call of Duty, además de abundar también en los videojuegos de terror.

Una vez hemos resuelto esta cuestión, debemos tratar el tema del modelo 3D del castillejo. Este será proporcionado por los investigadores, y nosotros nos encargaremos de realizar los ajustes visuales necesarios y colocarlo en nuestro juego. Los problemas y el transcurso surgido con esto serán tratados más adelante.

Uno de los objetivos que teníamos al principio era que esta aplicación resultara de utilidad a todos aquellos que quisieran conocer todo lo relacionado con el castillejo, sin embargo, tras no realizar la colaboración ya previamente mencionada con el grupo de arqueólogos no se llegó a realizar el trabajo de rigurosidad histórica que pretendíamos desde un principio y que era esencial, por cuestiones obvias. A partir de este punto, la información usada pasó a ser escasa o no correcta (no así la que se encuentra en la ventana de Historia). Esto nos llevó a descartar el uso de puntos de información que íbamos a usar, al estilo del castillo de Mirabel. Iban a ser ciertos puntos concretos del

castillejo en los que el usuario podría pulsar para obtener datos sobre algo en concreto, como por ejemplo el estilo arquitectónico de un arco. Para no dejar la aplicación sin ningún tipo de información durante la visita decidimos colocar un panel lateral que indicara al usuario en qué sala se encuentra en cada momento y algo de información sobre ella, aunque haya sido inventada.

La interfaz de la aplicación va a consistir por tanto en un menú principal, desde el cual podrás ir a una ventana de ayuda en la que hablará brevemente sobre lo que consite la aplicación, una ventana de historia que explicará a grandes rasgos que era el castillejo y su importancia, y otra ventana que será la visita. En ella, en la versión web y Windows tendremos un menú de pausa con una ventana para configurar las opciones gráficas de la visita, pudiendo elegir entre rendimiento y calidad, un botón para volver al principio de la visita, y otro para salir de la misma. Además, durante la misma el usuario podrá desplegar el panel lateral anteriormente explicado con información sobre las salas. En la versión para dispositivos Android se introducirá un joystick para poder mover al personaje y otro botón para desplegar el panel de información. El resto de elementos y el diseño de la interfaz serán iguales.

Además de lo comentado, había muchas opciones más para completar la aplicación, sin embargo, el tiempo a dedicar es limitado y no todo podía realizarse. Vamos a comentar brevemente algunas de ellas.

- Una era situar un pequeño panel en la pantalla con el que el usuario pudiera elegir acudir a una sala automáticamente. Esto hubiera sido interesante para aquellos usuarios que no se sintieran familiarizados con el uso de los controles, sin embargo, elimina un poco el factor de integración que se siente al mover tu mismo al usuario.
- Otro punto era, al igual que el castillo de Peracense, mover la cámara horizontalmente cuando el personaje no se mueve. También se comenzó a realizar un fondo para el menú principal que iba a consistir en distintos planos en movimiento del castillejo con una cámara, lo cual generaba más problemas de rendimiento y cargaba en exceso la aplicación, principal motivo para ser descartado. En su lugar, para que el usuario pudiera ver el castillejo desde fuera se realizó la aplicación de AR. En esta, si el teléfono del usuario es compatible, podrá ver el castillejo desde su casa en una mesa, en el suelo o la superficie que el mismo desee, permitiendo así tener una visión general del castillejo, desde cualquier ángulo y con una inmersión única.
- El castillo de Peracense tenía también un mapa, sin embargo aquí eso no tenía sentido ya que el castillejo es pequeño y realmente tiene pocos puntos de interés, por lo que no es necesario ni tener un mapa ni guiar la visita. Premiamos la libertad a la sobrecarga de automatización.
- La idea más interesante que no ha podido realizarse era utilizar el GPS de los dispositivos móviles para, mientras que los visitantes se encontraran en el

castillejo, en la aplicación se pudiera cambiar de sala automáticamente tras moverse la persona que la está usando en el propio castillejo. Esto iba a probarse cuando se realizara la visita al castillejo con los arqueólogos, sin embargo, estalló la pandemia del COVID-19 y la cuarentena en España, quedando así la visita cancelada y sin la posibilidad de probar la cobertura o fiabilidad del sistema.

Una vez explicados los objetivos se va a exponer la planificación que realicé al comienzo de la realización del proyecto. En esta, aún no conocía qué iba a poder realizar finalmente y si iba a realizar la colaboración con los arqueólogos, por lo que aparecen objetivos no mencionados aquí arriba o ya descartados.

Planificación

El trabajo ha contado con dos fases de trabajo. En la primera, realizada durante los meses de febrero a julio, ambos incluidos, ha habido un trabajo de análisis de objetivos, trabajo con los modelos 3D y software relacionado, reuniones con los arqueólogos y Paco Guil y aprendizaje y puesta a punto del software utilizado, especialmente del lenguaje de programación Latex, utilizado para la realización de esta memoria.

La segunda fase queda reflejada en el siguiente calendario, realizado a fecha de 11 de agosto se relata el planteamiento inicial que se tenía desde ese día hasta una fecha próxima a la entrega del trabajo, 6 de Agosto. En esta fase se encuentra la realización final del trabajo tras tener todo planteado y sobre la mesa.

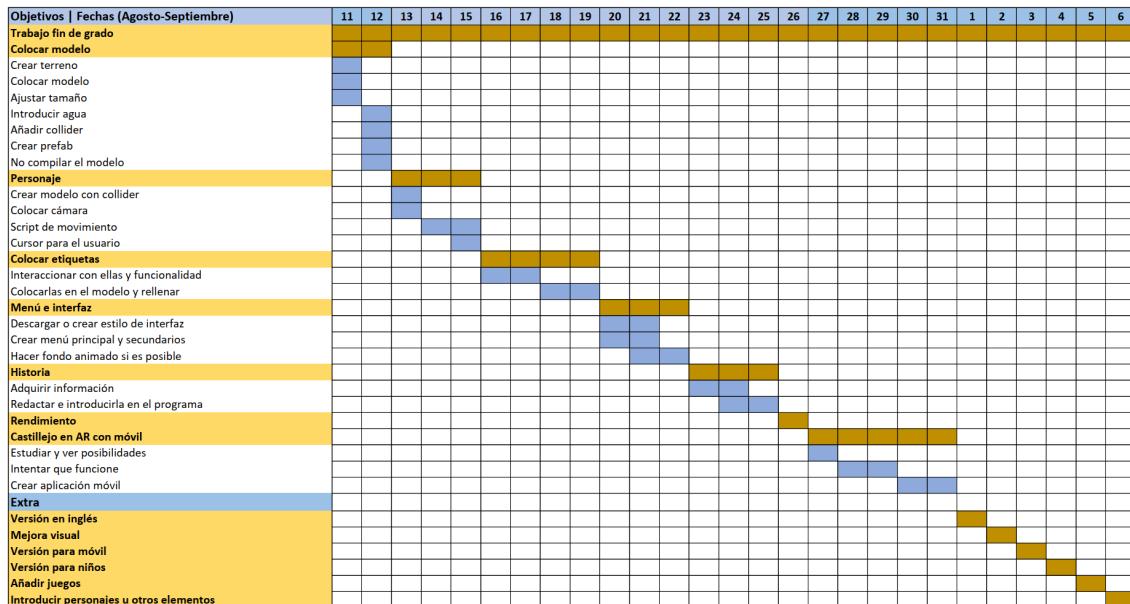


Figura 13: Planificación de la segunda fase.

La idea de este calendario es que cada día se realizaran aquellas tareas con el cuadro coloreado de color azul. Los hitos (objetivos marcados con color marrón), tienen

marcada su fecha de realización esperada en marrón oscuro, y engloban minifases de la fase de realización del trabajo.

Al final de este apartado, se adjunta una imagen con el calendario final seguido. Este contendrá las fechas en las que se ha realizado finalmente cada objetivo, y la eliminación o inserción de nuevos objetivos.

En la imagen podemos apreciar que ciertos hitos están ampliamente detallados, especialmente los primeros, mientras que del resto no se dispone ni de un desglose de tareas. Esto es porque se ha realizado un trabajo previo al inicio de esta planificación que consistió en una toma de contacto con el material proporcionado por los arqueólogos, además de que las primeras tareas a realizar han sido ya estudiadas en alguna asignatura de la carrera. El resto, o no han sido estudiadas, o tampoco han sido previamente preparadas.

A continuación, se va a comentar cada hito y sus correspondientes objetivos. Explicando en qué consiste cada uno y por qué se han asignado ese número de días para él. Comenzamos por el primero.

Colocar modelo

Este hito consiste en la importación del modelo del castillejo a Unity, su colocación, perfeccionamiento, y finalmente, conseguir que cargue en tiempo de ejecución.

En primer lugar se realizará un terreno que simulará la colina sobre la que se encuentran los restos del castillejo en la actualidad:



Figura 14: Imagen del castillejo en la actualidad.

Después se situará el modelo en la misma, aumentando su tamaño hasta conseguir que sea lo más realista posible. Se introducirá agua para ambas albercas del interior del palacio. Se añadirá al modelo un collider para poder moverse por el palacio e interaccionar con el mismo, y se exportará un "prefab", que consiste en exportar el modelo con todos los ajustes realizados para poder luego volver a introducirlo en cualquier escena y que tenga las mismas características que el original. Esto iba a ser utilizado para no compilar el modelo, pero finalmente no se ha utilizado. En el siguiente apartado de la memoria se explicará en detalle.

Personaje

Consistirá en introducir un personaje sin cuerpo que representará al usuario cuando se mueva por el castillejo. Se deberá programar o usar un script ya realizado para moverse con el personaje por la aplicación, y el mismo podría tener un cursor para seleccionar cosas.

Colocar etiquetas

Este hito es importante y podría tener cierta complejidad. Consistirá en colocar ciertos puntos de interés en el castillejo que proporcionarán información adicional a la visual al usuario, para así poder saber qué está viendo y realizar así una visita más completa. No he realizado nada parecido hasta ahora, y es por eso por lo que el hito no se encuentra desglosado en objetivos. Calculo que durante cuatro días tendré tiempo de estudiar e investigar como realizarlo y hacerlo. Si no es así se realizará un reajuste en la planificación.

Menú e interfaz

En este hito se realizarán los menús. Y lo digo en plural porque puede ser que se desarrolle dos menús: uno que contenga un fondo animado con el castillejo y otro que no. Esta diferenciación se realiza por motivos de rendimiento, no solo para equipos con potencia limitada, también para que la versión de navegadores no sea lenta o de una mala experiencia al usuario. Además de diseñar la forma del menú y las entradas que tendrá, se desarrollará el diseño de la interfaz.

Historia

Una vez que la estructura de la aplicación se encuentre realizada en este punto, será el momento de introducir toda la información sobre el castillejo, el contexto histórico, o la historia que explique el origen del lugar.

Rendimiento

En el hito de rendimiento se estudiará cómo se comporta la aplicación en diferentes situaciones, plataformas y equipos para valorar si el rendimiento es adecuado. En el caso de que no lo sea, se reducirá la calidad gráfica de la aplicación para que la experiencia no se vea empañada.

castillejo en AR con el móvil

En este hito que durará, según la planificación, 5 días, me informaré, estudiare distintas opciones e intentaré crear una aplicación de realidad aumentada (AR) para dispositivos móviles que permita visualizar el castillejo como si lo estuvieras viendo encima de una mesa, el suelo, o cualquier superficie que el usuario desee. Al no haber realizado nada similar con Unity hasta ahora y de una manera menos académica que cuando estudiamos realidad aumentada en la asignatura de Visión Artificial necesitaré presumiblemente más tiempo que en resto de hitos. Incluso puede que después de intentarlo no consiga realizarlo.

Extra

El resto de los hitos se encuentran bajo la etiqueta de 'Extra'. Son aportaciones y opciones que podrían enriquecer aún más la aplicación, que solo se realizarían en el caso de que los plazos se estén cumpliendo mejor de lo previsto o que sobre tiempo hasta la entrega del TFG. Algunos de los puntos son sencillos, como traducir la aplicación a inglés, y otros algo más complejos, como como introducir personajes o algún juego para niños.

Metodología

El método de trabajo que seguiré consistirá en realizar primero un hito, objetivo a objetivo, y cuando se encuentre terminado marcarlo así en el calendario y redactar un resumen de este mismo o los problemas que haya podido tener en la memoria. De esta manera la memoria se va realizando de manera paralela al trabajo, sin interrumpir en la realización de este, y pudiendo plasmar el proceso sin olvidar los detalles del mismo.

Realización final

Finalmente, el cuadro de planificación inicial ha quedado de esta manera:

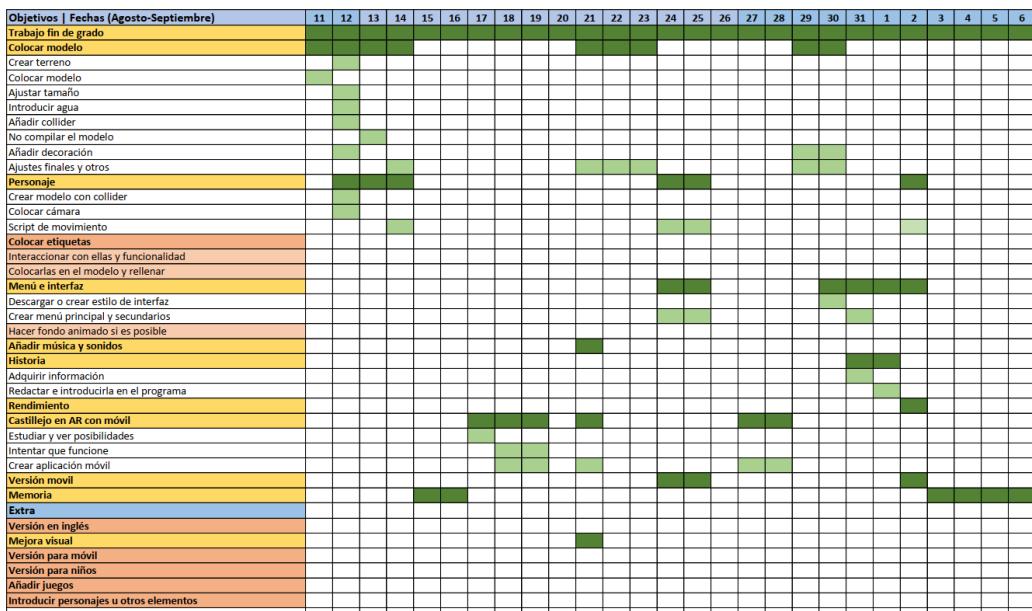


Figura 15: Calendario final de realización de objetivos.



Figura 16: Leyenda.

Como se puede apreciar, no se han seguido las fechas establecidas en un inicio. De hecho, ni si quiera se respeta el orden, realizando tareas del primer hito al final y tareas del hito de realidad aumentada al principio. Esto no tiene por qué ser algo negativo, sin embargo, si es útil, ya que permite realizar un análisis de lo que ha ocurrido y como evitarlo en futuras ocasiones, tal y como vimos en asignaturas como GPDS.

Lo que ha ocurrido es que este trabajo era inicialmente una colaboración, y al no haberse esta realizado de la manera deseada ha habido nuevos elementos que han ido cambiando lo que se disponía al principio para realizar el trabajo.

Sobre la metodología, se siguió el patrón de realizar un hito y documentarlo tan solo con el primero, ya que después, tanto Paco como yo centramos nuestros esfuerzos en realizar el trabajo lo antes posible, por posibles problemas que pudieran suceder y realizarlo de manera más ágil. Esto provocó que la memoria quedara relegada hasta

la finalización de las aplicaciones. Al contrario de lo que se podría pensar, esto ha supuesto algunas ventajas. La primera de ellas es que en la realización de la memoria se contaba con el trabajo terminado, por lo que la perspectiva final de lo que había sido y el camino recorrido era mayor, pudiendo así realizar una mejor memoria. Por otro lado, ha permitido recalcar lo más importante del mismo sin olvidar detalles, ya que el trabajo y la memoria han sido realizados en el espacio de un mes, lo cual es un tiempo compactado y sin haber realizado otras tareas, por lo que no ha supuesto problemas que algo se realizara hace tiempo y no se recordara.

Sobre los objetivos no realizados, mayormente ha sido por falta de tiempo o por haber dado prioridad a otras tareas. Por ejemplo, en el caso de las etiquetas, al no haber trabajado juntamente con los investigadores no se disponía de información arquitectónica o artística suficiente del modelo como para introducir etiquetas de información. En su sustitución, se ha desarrollado un panel lateral de información que indica en cada momento en qué sala se encuentra el usuario e información sobre ella.

Por último, comentar que ese calendario final es una aproximación, ya que se han realizado otras tareas que no se reflejan o se encuentran englobadas en otros objetivos e hitos. Lo más destacable se detallará más adelante.

Diseño y resolución del trabajo realizado

A continuación se va a exponer el proceso seguido de la realización del trabajo, recalando aquellas situaciones que han supuesto un reto o han planteado una mayor dificultad.

Software utilizado y su versión

En primer lugar, dedicaremos unas breves líneas a hablar sobre el software utilizado en la realización de las aplicaciones.

El motor de videojuegos utilizado, tal y como se ha comentado anteriormente, es **Unity**. La elección del motor ha venido condicionada por varios motivos, el primero de ellos, el ya conocimiento del mismo debido a su uso en distintas asignaturas de la carrera.



Figura 17: Logo de Unity.

Un motor de videojuegos es una herramienta compleja, que necesita de un tiempo para poder ser utilizada con comodidad para poder explotar y conocer todas sus opciones. Por lo tanto, si se hubiera escogido otro modelo se habría necesitado un considerable tiempo de adaptación que, además, tampoco se hubiera visto traducido en un mejor resultado.

Y ese es el siguiente motivo de la elección de este motor. Unreal Engine, la otra gran opción en el mercado, es un motor más orientado a obtener una mayor calidad

visual y un juego más serio que con Unity, más orientado a un mejor rendimiento, facilidad de uso, y desarrollo de aplicaciones móviles o web.

En el resto del software utilizado encontramos **Blender**, un programa Open Source de Modelado en 3D, entre otras opciones, utilizado ampliamente por una amplia comunidad de diseñadores. No es solo utilizado para videojuegos, también para la realización de cortos o incluso películas en 3D.



Figura 18: Logo de Blender.

Para la confección de la memoria se ha utilizado el lenguaje **Látex**, siendo la memoria escrita con **Visual Studio Code**, un editor de código Open Source.



Figura 19: Logo de Látex.



Figura 20: Logo de Visual Studio Code.

Por último, para el control de versiones y backup se ha utilizado la plataforma de **GitHub**, también utilizada durante la carrera y la comunidad de Software. Para la gestión del repositorio se ha utilizado el programa de escritorio llamado **GitKraken**.



Figura 21: Logo de GitHub.



Figura 22: Logo de GitKraken.

Modelos 3D

Este ha sido uno de los principales quebraderos de cabeza y puntos conflictivos que sin la ayuda de Paco no podría haber resuelto tan bien.

Los arqueólogos trabajaban con el software 3ds Max para crear los modelos del castillejo o el terreno de la zona. Esto genera archivos que no pueden ser utilizados en Blender ni en Unity, por lo que era necesario exportarlos a un formato con el que pudiéramos trabajar. En un primer momento conseguimos exportar nosotros mismos los archivos, pero sin tener las texturas, por lo que tuvimos que pedirles finalmente un modelo con texturas ya exportado. Sin embargo, el modelo que nos dieron no tenía las texturas asignadas, por lo que finalmente fueron asignadas en Unity.

Pero primero, había que hacer un trabajo previo con los modelos, ya que estos estaban divididos en cientos de archivos, algunos de ellos sin utilidad en el modelo. Invertimos tiempo entonces en eliminar y ordenar todos esos archivos para reducir el tamaño del modelo, haciéndolo así usable y menos pesado. A pesar de esto, el modelo seguía siendo pesado, y es que el número de vértices era muy elevado. Se reemplazó la ornamentación original por una falsa, que daba un buen resultado visual. En las siguientes dos imágenes se puede observar la diferencia:



Figura 23: Ornamentación falsa.



Figura 24: Ornamentación real.

A pesar de los problemas ya comentados, había uno más grave, sobre todo por las consecuencias visuales que conlleva, y es el problema de las normales. Estas no estaban bien calculadas en el modelo original, lo que causaba que las esquinas no se vieran correctamente, que se vean huecos, las texturas se muevan, o por ejemplo no se vean los límites de las albercas.

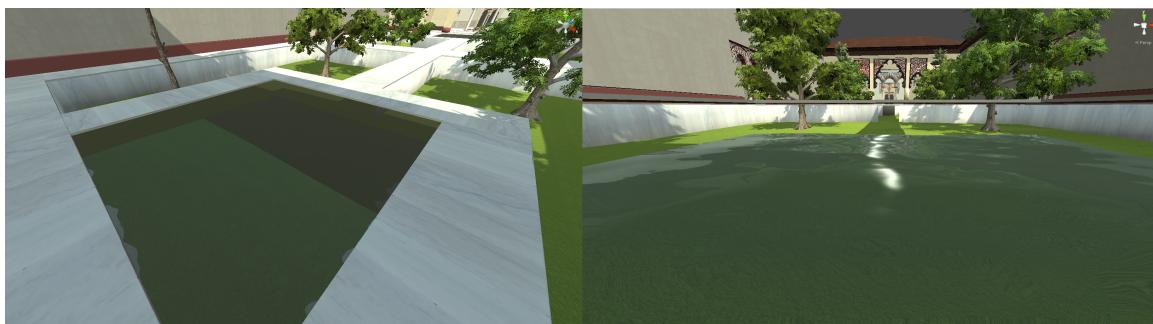


Figura 25: Albercas con problemas de normales.



Figura 26: Albercas con los problemas solucionados.

Se intentaron solucionar estos problemas, aunque aún sigue habiéndolos, ya que para dejar el modelo perfecto es necesaria una cantidad de tiempo demasiado grande.

Una vez que pensábamos teníamos el modelo ya situado y con un buen resultado los arqueólogos nos enviaros archivos que no pudieron enviarnos al principio. Esto incluía

la muralla defensiva que rodeaba el interior del castillejo, todo el terreno exterior, y una escalera para poder subir del patio exterior al interior del palacio. Tuvimos entonces que unir el modelo que teníamos ya realizado con los nuevos recibidos. Estos contenían nuevos fallos que ya no daba tiempo a resolver, aunque el resultado final es incluso mejor que al principio, ya que el castillejo dejaba de verse incompleto e incluso parecía imponente con esa nueva muralla. En la siguiente imagen se puede apreciar el castillejo anterior con el terreno que realicé en un primer momento y acabé descartando y el aspecto final.

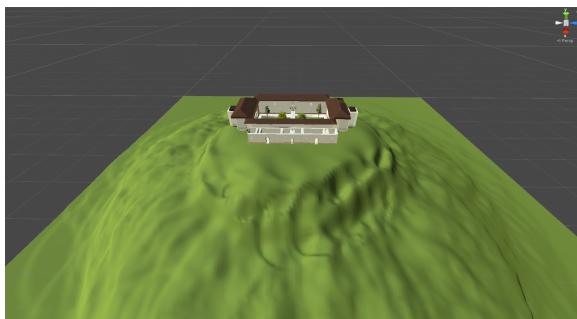


Figura 27: castillejo y terreno inicial.



Figura 28: castillejo inicial.



Figura 29: castillejo y terreno final.



Figura 30: castillejo final.

Finalmente hubo dos problemas a resolver:

- El primero y menos importante de ellos fue que, cuando el personaje se encontraba en el patio exterior, se veía el terreno en dos zonas del patio, lo cual quedaba visualmente mal. Desactivé entonces el terreno cuando el usuario se encontraba en el patio para activarlo cuando subiera a la parte superior del castillejo.



Figura 31: Problemas con el terreno en el patio exterior.

- El otro problema vino con la puerta que daba acceso a las escaleras, ya que se encontraba cerrada y era necesario que el personaje la atravesara si quería que pudiera subir al castillejo, algo importante a mi juicio ya que aportaba una sensación única de inmersión. Atravesar la puerta no era una opción por tanto ya que rompe directamente con el principio de inmersión a mi parecer, decidí entonces resolverlo al estilo videojuegos. Cuando en los videojuegos la capacidad del hardware no permite tener en memoria mucha información, los desarrolladores cargan una escena solo cuando se ha salido de la otra, produciéndose así transiciones cuando el personaje pasaba cierta zona del mapa o entraba a un lugar a través de una sala. Decidí entonces recurrir a ese principio, haciendo que cuando el usuario se acercara a la puerta le apareciera un mensaje para poder acceder a la escalera.



Figura 32: Panel para entrar al interior del castillejo.



Figura 33: Panel para salir al patio exterior.

Esto se realizó utilizando los Triggers de Unity. Los mismos consisten en activar cierta función de un script cuando se interacciona con un objeto. Se colocaron entonces objetos alrededor de las puertas que interactuarían con el usuario, y se programó que el usuario pudiera cambiar de sala al pulsar una tecla. Además, hubo que solucionar otros errores surgidos a partir de utilizar este sistema, como un problema con el panel de información de las salas, sin embargo, el resultado final fue bueno.

No compilación

En la primera reunión con los arqueólogos se habló de la posibilidad de que el modelo 3D del castillejo se cargara en el programa en tiempo de ejecución, de esta manera no se encontraría previamente compilado. Esto se solicitó por un motivo: que los arqueólogos pudieran ir mejorando y cambiando el modelo con el paso del tiempo, y que así pudieran introducirlo en el visor sin tener que utilizar Unity y estar continuamente compilando el programa en las distintas plataformas. La colaboración con ellos terminó después de que yo realizara esto, por lo que expondré en lo que consistió ya que me llevó un tiempo realizarlo y es funcionalidad que sigue vigente, aunque no se vaya a utilizar.

La primera opción que se exploró para realizar esto fueron los llamados **AssetBundles** de Unity. Se tratan de archivos que se crean a partir de *prefabs* y que se cargan en tiempo de ejecución. Se realizó una prueba con los mismos y funcionaban perfectamente. Cargaban rápido y sin problemas. Son ampliamente utilizados en la comunidad de Unity y suelen utilizarse para cargar los modelos de las escenas de Unity poco a poco, para no sobrecargar la escena y penalizar así el rendimiento. Los pasos para utilizar los AssetBundles se pueden observar en esta instantánea de un vídeo de Youtube:

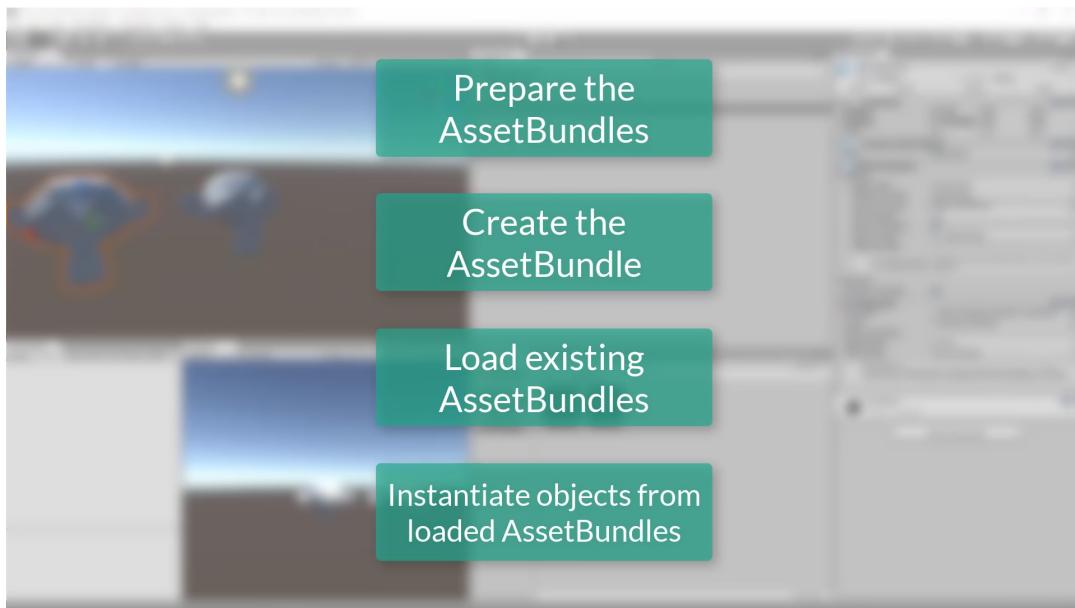


Figura 34: Instantánea del vídeo que muestra los pasos para utilizar los AssetBundles.

Como se puede apreciar en la imagen tenemos dos primeros pasos de preparación y creación de los AssetBundles. Esto genera un problema ya que para generar los AssetBundles debes utilizar el editor de Unity, debido a que no es posible generar estos archivos en tiempo de ejecución. Por lo tanto, si los arqueólogos quisieran más tarde actualizar el modelo tendrían que importarlo a Unity y generar un AssetBundle

a partir de él. No deja de ser un mal menor, pero sin lugar a duda es un impedimento para aquellos que nunca han utilizado Unity ni tienen intención de hacerlo.

Se intentó entonces encontrar la manera de cargar directamente los archivos que se exportan desde Blender a la escena de Unity, sin utilizar prefabs ni importar previamente los modelos al proyecto. La extensión de archivo que se utiliza comúnmente para importar modelos en tres dimensiones a Unity es .fbx, de hecho es el que se estuvo utilizando desde el comienzo de la realización del proyecto en Unity. Parece lógico por lo tanto intentar cargar este tipo de archivos en tiempo de ejecución, algo que, efectivamente, se puede realizar. En la tienda de Assets de Unity se puede encontrar un Asset que permite importar en tiempo de ejecución numerosos tipos de archivo a cualquier escena, entre ellos, nuestro formato .fbx. Este Asset sin embargo no era gratuito, en el momento de la realización del proyecto tenía un precio de algo más de 22 euros:

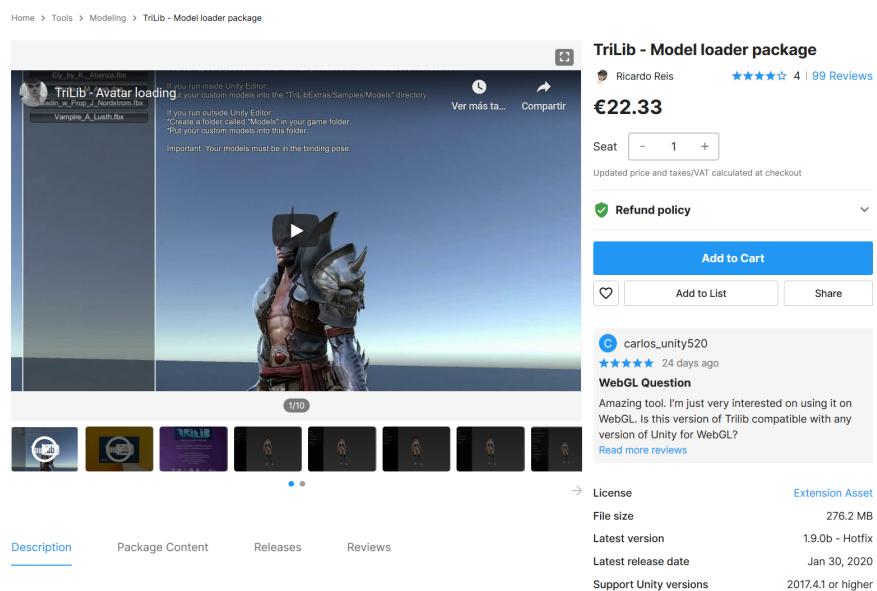


Figura 35: Instantánea del Asset TriLib.

El único Asset de este tipo que se pudo encontrar fue el llamado **Runtime OBJ Importer**, que como el propio nombre indica, se trata de un script que permite importar archivos con extensión .obj en tiempo de ejecución:

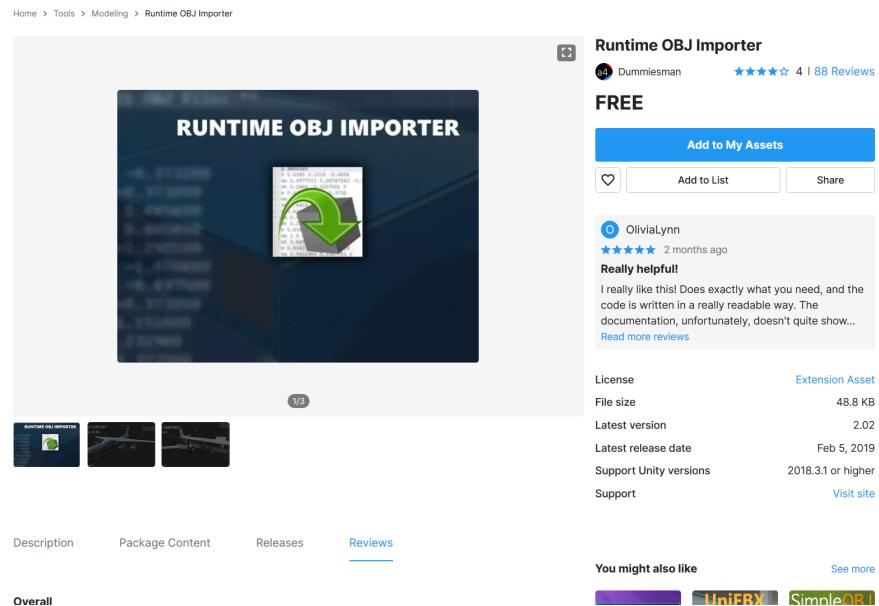


Figura 36: Instantánea del Asset Runtime obj importer.

Esto podía sernos útil, aunque usualmente no se importan archivos .obj a Unity es algo que se puede realizar. Se generó junto al archivo .obj del modelo un archivo .mtl que permitiría cargar las texturas y se intentó utilizar el script. Funcionaba perfectamente, aunque añadía un cierto tiempo de carga de la escena que no ocurría con los AssetBundles. Es algo esperado ya que este tipo de Assets no están orientados para lo mismo que los AssetBundles.

Para que la escena pueda cargar por tanto el modelo, se programó un script que se ejecutaría al comienzo de la escena en caso de querer utilizarlo y se encargaría de cargar el modelo, además de añadir un collider a cada elemento de este. Este script busca en la ruta que le indiquemos el archivo. Si no lo encuentra el archivo no será cargado y la escena se mostrará sin el modelo. Es por lo tanto importante que el archivo se encuentre en la ruta indicada (preferiblemente una ruta dentro de la carpeta del ejecutable), y ésto debe realizarse manualmente, ya que dicho archivo no se introduce en la carpeta durante la compilación.

Interfaz de la aplicación

Para el aspecto estético de la interfaz se utilizó un paquete de fondos con un toque ornamental y decorativo árabe. La fuente utilizada en la gran mayoría de textos de la aplicación es Almanar [10], con la excepción del uso de Aladdin o Arial en algún caso. Este último puede verse en las ventanas de Historia, Ayuda, y en el desplegable de opciones de rendimiento que veremos más adelante.

Además del común menú principal, hay dos ventanas más. Una de ellas ofrece un

poco de contexto histórico sobre el castillejo y Monteagudo. La otra, es una ventana de Ayuda que contiene información sobre el proyecto.

En la visita, he introducido una serie de paneles iniciales que indican al usuario como realizar la visita.



Figura 37: Paneles iniciales de ayuda.

Además, hay un menú de pausa con unas funciones básicas para el usuario y una ventana de configuración gráfica.

Rendimiento



Figura 38: Panel de opciones de rendimiento.

Dicho panel es el que se puede observar en la imagen superior. El usuario puede seleccionar la calidad gráfica que desea. A mayor calidad menor rendimiento de la aplicación. Esto es especialmente útil en la versión web y en dispositivos móviles, donde no se cuenta con tanta potencia. El panel cuenta con las siguientes opciones de calidad:

- **Muy baja.** La opción gráfica de Unity se establece en muy baja, la mínima que podemos escoger. Esto afecta a la iluminación, sombras, y texturas, entre otros. Se desactiva el sistema de partículas que es meramente decorativo y el postprocesado, que ofrece una calidad visual importante.
- **Baja.** Debido a la diferencia de calidad visual que supone activar el postprocesado es lo que se activa al subir de Muy baja a Baja. El resto continúa siendo igual.
- **Media.** La calidad de Unity es subida a baja, un paso por encima de las dos anteriores. También se activa el sistema de partículas.
- **Alta.** En este nivel aumentamos la calidad gráfica a Alto, saltándonos un escalón, ya que no hay más opciones secundarias para activar.
- **Muy alta.** Nos situamos en el nivel más alto de calidad disponible. La calidad gráfica que seleccionamos de las que ofrece Unity es Ultra, la máxima disponible. Además se realiza un cambio en los modelos. Se sustituye el modelo que contiene las yeserías (ornamentación) falsas por el que las tiene reales. Esto da un plus de realismo y calidad.

Además, este panel cuenta con un cuadro en el que desactivar la presencia de animales. Si se desactiva se dejarán de crear golondrinas, eliminándose poco a poco las que hayan en ese momento, y se desactivan los peces. Esto puede suponer también una pequeña mejora de rendimiento. A pesar de ello, se mantienen los sonidos de pájaros para no eliminar ese efecto inmersivo.

Panel de información

Anteriormente se ha mencionado la existencia de un panel de información, el cual tiene el aspecto visual que se puede ver en la siguiente instantánea:



Figura 39: Panel de información a la derecha.

Se puede apreciar el título de cada sala y su descripción. Este panel funciona con triggers, como lo hacía el método de entrada al castillo.

Cada sala contiene en su interior una caja transparente al usuario. Cada vez que el mismo entra en una, esta solicita cambiar el nombre y la descripción del panel por la de la sala actual. Sin embargo, esto solo podrá realizarse si ya se ha salido de la sala en la que el usuario se encuentra. Esto es así para evitar que se cambie el panel cuando aún no se ha salido de la sala anterior. Para realizar esto, se han utilizado los conocimientos aprendidos en la asignatura de Programación Concurrente, ya que estamos en un caso similar a cuando dos hilos o procesos intentan cambiar una variable compartida al mismo tiempo. Si la variable está siendo ocupada por una sala, la otra que desee cambiarla deberá esperar en una cola (de una sola posición) hasta que la sala anterior libere el panel cambiando el valor de una variable.

Además, si se sale de una sala y no se ha entrado en ninguna, se marca que el

usuario está en una sala desconocida.

Otros elementos

Para hacer la experiencia más completa se han introducido elementos adicionales. Los mismos son:

- **Postprocesado.** La introducción de esta mejora gráfica es significativa en la imagen que percibe el usuario. No se trata de una opción que busque el realismo, sino resaltar la imagen y hacerla más viva.
- **Animales.** Modelamos y texturizamos unas golondrinas y unos peces. Estos Aportan un aspecto variable e incluso más realista, en especial los pájaros. Programamos un script para el movimiento tanto de las golondrinas como de los peces, y otro para la generación o destrucción de las golondrinas.



Figura 40: Imagen de las golondrinas sobre el castillejo.



Figura 41: Imagen de los peces en una alberca.

- **Árboles.** En el patio central del castillejo debía haber vegetación para recalcar la belleza y sentido de ser del mismo. Cuando realicé un terreno al comienzo del trabajo introduce en el modelo árboles que incluye Unity para los terrenos (estos se pueden apreciar en la figura 25), sin embargo, con el nuevo modelo decidí introducir árboles típicos de la zona de Murcia. Descargué así los modelos de un limonero, de una palmera, y de un pino. El naranjo es ficticio, utilicé la base de otro árbol y le introduje naranjas manualmente. El aspecto y el realismo de estos árboles es superior a los que teníamos anteriormente.
- **Música árabe ambiental.**

Aplicación de realidad aumentada

Sobre la aplicación de realidad aumentada, en primer lugar hay que destacar que el diseño de la interfaz es diferente al usado en la primera aplicación, dado que buscaba un aspecto más sobrio y neutro, separado del aspecto de la otra.

Esta aplicación cuenta con otra ventana de ayuda y dos escenas de realidad aumentada. La tecnología usada para el uso de la realidad aumentada es el paquete AR Fundation disponible en Unity, que proporciona una interfaz común de los paquetes AR Kit y AR Core. Con este paquete tenemos disponible la detección de planos y de imágenes, ambos usados en la aplicación en dos escenas. Las imágenes que se pueden utilizar para situar el castillejo son las siguientes:



Figura 42: Estrella en círculo.

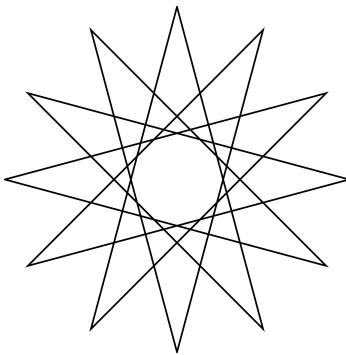


Figura 43: Estrella.

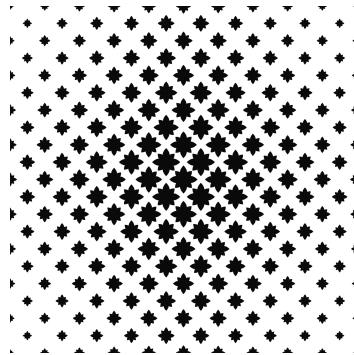


Figura 44: Monochroma.

Sin embargo, la única que da un rendimiento aceptable es la estrella con círculo que la rodea, las otras no son detectadas o se tarda demasiado tiempo en hacerlo. Esto se produce porque AR Fundation no es capaz de encontrar suficientes puntos de interés en las imágenes y no es capaz de detectarlas.

La detección de planos también tiene un rendimiento pobre, ya que necesita de ciertos segundos para detectarlos. Sin embargo, una vez que lo hace rinde mejor que las imágenes, que sufren de pequeños fallos.

A continuación, encontramos tres imágenes de la aplicación. Una del menú principal, otra del diálogo que aparece para colocar el castillejo y una tercera del castillejo una vez instanciado:



Figura 45: Menú de la aplicación.



Figura 46: Diálogo para instanciar.



Figura 47: castillejo con AR.

Conclusiones y vías futuras

El resultado final es bueno teniendo en cuenta los inconvenientes encontrados que han retrasado el proyecto y ocasionado problemas. La interfaz presenta una línea única de diseño, es posible disfrutar la aplicación en distintas plataformas, el rendimiento es bueno y el aspecto estético también. Sin duda, el resultado podría haber sido muy distinto de haberse realizado la colaboración, pudiendo haber sido incluso algo completamente diferente si se hubiera tratado de una aplicación exclusivamente de realidad aumentada, como en el caso del castillo de Falaise. Sin embargo, la orientación final del proyecto es buena y es una buena base para lo que podría ser una aplicación de este estilo en el futuro.

Además, se trata de una aplicación que realza la cultura y la historia de la región de Murcia, siendo esto importante para conocer nuestro pasado y que se explote la riqueza cultural que tenemos y sigue apareciendo día tras día bajo tierra. Con estas aplicaciones e iniciativas se acerca la cultura a la población y se dan a conocer nuevos elementos de nuestro patrimonio hasta ahora desconocidos.

Sobre aspectos que mejorar o cómo ampliar el trabajo en un futuro, se han mencionado muchas cosas que podrían haber completado el trabajo, pero lo que más sentido tiene a mi parecer es introducir personajes o algún tipo de minijuego o elemento con el que los usuarios puedan interaccionar, ya que la experiencia es más cercana a un juego serio que a una visita. Con esto, el motivo de utilizarla no sería solo la curiosidad histórica, sino también entretenimiento y ocio, pudiendo orientar esto también a los niños e introducirlo en las escuelas, ya que son los niños el eslabón más familiarizado con las tecnologías y los que mayor partido sacan a estas iniciativas.

Referencias

- [1] Autores varios. *Historia de Murcia*. Wikipedia.
https://es.wikipedia.org/wiki/Historia_de_Murcia
- [2] Miguel CM. *¿Qué es un murciano? Historia de la Región de Murcia*. Youtube. 2016.
https://youtu.be/P_R-ck03vKw
- [3] Autores varios. *Castillo de Monteagudo*. Wikipedia.
https://es.wikipedia.org/wiki/Castillo_de_Monteagudo
- [4] Castillo de Peracense. *Visita virtual*:
<http://www.peracense.es/castillo/visita-virtual/>
- [5] Arts and Culture.
<https://artsandculture.google.com/>
- [6] Visita al castillo de Falaise.
<https://www.france-voyage.com/francia-ciudades/falaise-1924/castillo-falaise-12102.htm>
- [7] Alejandro Gómez López. *Desarrollo de un videojuego. [Flame Knights Chronicles.]*
- [8] Daniel G. Aparicio. *'Invizimals' y la magia de la realidad aumentada, según sus creadores*. 20 minutos. 2009.
<https://www.20minutos.es/videojuegos/noticia/invizimals-novarama-entrevista-477981/0/>
- [9] Israel Fernández. *La historia de Google Glass: por qué fracasaron y por qué, esta vez, van a triunfar*. Nobbot. 2017.
<https://www.nobbot.com/futuro/historia-google-glass-fracasaron-triunfar/>
- [10] DaFont. Fuente Almanar
<https://www.dafont.com/es/almanar.font>