

# Lara Croft Go

---

**Isabel Codina García y Borja Fernández Ruizdelgado**



Memoria del juego 3D  
Asignatura Videojuegos  
Curso 2017 - 2018 Q2 (Primavera)  
Facultad de Informática de Barcelona  
Universidad Politécnica de Cataluña

# Índice

<b>Índice</b>	1
<b>Información sobre el juego</b>	2
<b>Descripción del proyecto</b>	3
Descripción	3
Objetivos del juego	3
Características principales	4
Instrucciones	12
Diagrama de ventanas y Flow chart	13
<b>Metodología</b>	14
Planificación y Backlog	16
Diagrama de Gantt y Seguimiento Taiga	18
Versiones	20
Hitos abiertos y cerrados	21
Puntos tratados en las reuniones	21
Método colaboración	21
<b>Conclusiones</b>	22
<b>Bibliografía</b>	23
<b>Anexo</b>	25

# Información sobre el juego

El juego original, *Lara Croft Go*, cuyo logo podemos ver en la *figura 1.1*, se publicó en agosto de 2015 para Android, iOS, Windows Phone y Windows 10. Fue desarrollado por *Square Enix Montreal* y publicado por *Square Enix*.



Figura 1.1. Logo.

Es el segundo juego de la trilogía *Go* de puzzles single-player basados en turnos junto *Hitman Go* (2014) y *Deus Ex Go* (2016). Debido a su gran popularidad y las buenas críticas, se ha publicado en otras plataformas como Steam, PS4 y PSVita, macOS y Linux en diciembre 2016.

Tan solo para Android se han llegado a vender 500.000 unidades, haciendo llegar al juego al Top #8 en ventas de juegos de Puzzles. También fue marcado como *Selección de los editores*, significando que es uno de los mejores juegos publicados en *Google Play*.

Su recepción fue generalmente buena, como podemos ver en la *figura 1.2*. Destacó por su estética, su buen diseño y su fidelidad hacia la serie. Podemos observar los premios que ha conseguido en la *figura 1.3*.

Aggregate Score	
Aggregator	Score
Metacritic	84 / 100
Review Score	
Publication	Score
EGM	5 / 10
Game Informer	8 / 10
IGN	8'5 / 10
USgamer	★★★★★

Figura 1.2. Recepción.



Figura 1.3. Premios.

El juego fue desarrollado por un equipo de unas 15 personas. Se utilizó el motor gráfico Unity ya que al ser un equipo pequeño no deseaban perder tiempo en crear un motor propio y Unity se adaptaba a sus necesidades.

# Descripción del proyecto

El videojuego que hemos desarrollado es similar al juego de Lara Croft Go original: tenemos un personaje principal, sistema de movimiento y objetivos parecidos, enemigos con casi las mismas mecánicas y algunos mecanismos iguales. También hemos querido darle un aire semejante con texturas low-poly.

Nuestro proyecto es una versión mini del original, ya que los niveles son de un tamaño más reducido y solo hay 3 en total.

## Descripción

### Objetivos del juego

El objetivo principal del juego es llegar al final de cada nivel para conseguir el tesoro. Se deben evitar trampas y enemigos para cumplirlo, e incluso en ciertos momentos se deberá acabar con los enemigos para poder continuar.

También tenemos dos objetivos secundarios, uno de ellos es coleccionar las 9 gemas que hay en total. Hay 3 tipos diferentes y cada uno tiene 3 gemas repartidas en uno de los mapas. Podemos ver el icono 2D de las gemas en la *figura 2.1*.

El segundo objetivo complementario es conseguir el artefacto. Este está dividido en 3 piezas, repartidas entre los tres mapas. El artefacto lo podemos ver en la *figura 2.2*, y sus partes en la *figura 2.3*.

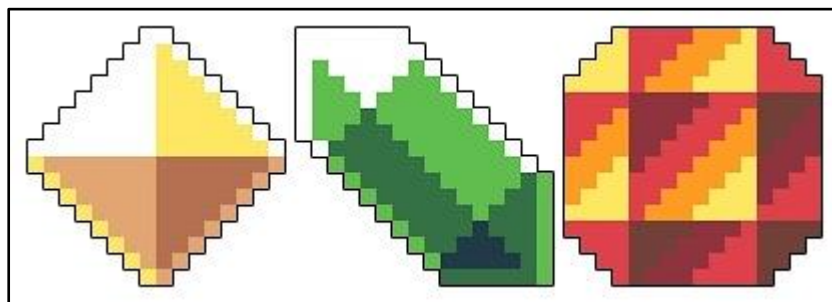


Figura 2.1. Imagen 2D de las gemas.

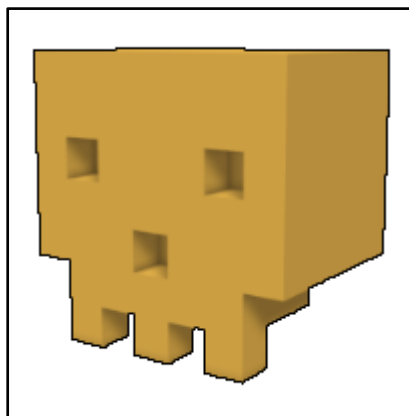


Figura 2.2. Modelo 3D del artefacto.

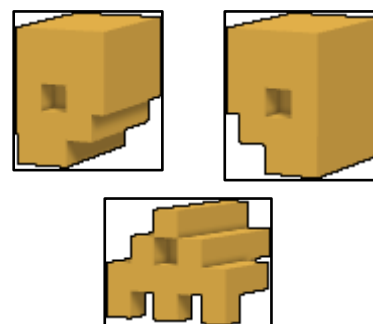


Figura 2.3. Partes del artefacto.

## Características principales

El juego, como se ha explicado anteriormente, tiene 3 niveles de dificultad progresiva. Tiene un solo personaje principal, pero tiene 3 enemigos, 4 mecanismos y 2 activables que accionan dos de los mecanismos. Cada entidad la explicaremos detalladamente a continuación:

### • Niveles

Como en el juego original, los niveles están organizados en un sistema de tiles en 3D. Todo bloque es de 1x1x1.

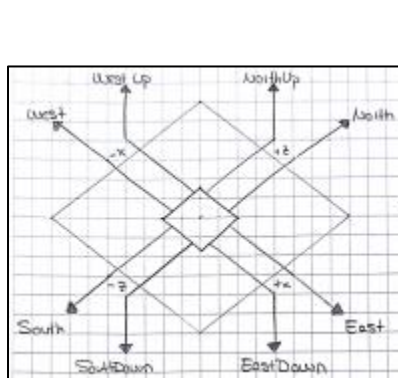
En el suelo, al ser cuadrados, solo se permiten movimientos en 4 direcciones, que hemos nombrado *North*, *South*, *East* y *West*. Como también se permite escalada, hemos tenido que añadir 4 direcciones de inicio y fin de escalada: *NorthUp*, *SouthDown*, *EastDown* y *WestUp*; y 2 de escalada: *Up* y *Down*.

Tenemos solo 4 direcciones de inicio y fin de escalada ya que, como en el original, solo hay dos orientaciones de escalada por cuestiones de visibilidad del camino.

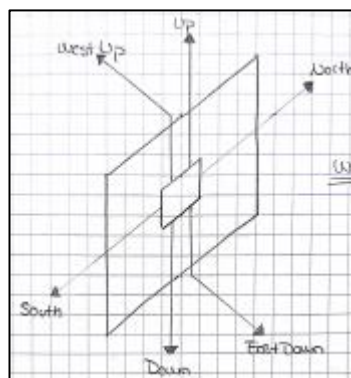
Podemos ver los posibles movimientos de cada tipo de tile en las *figuras 3.1*, *3.2* y *3.3*. En el juego, sabemos hacia donde se puede mover por el camino pintado en el bloque. La *figura 3.4* recoge todas las direcciones de cada tecla.

El sistema de obtener hacia donde es posible moverse nos llevó horas de planteamiento: al principio se pensó en un sistema de raycast, pero se descartó la idea por la complejidad de combinar el raycast con las animaciones. Finalmente, se llegó a la idea de un sistema de triggers en cada suelo o pared de los que se pudiera recoger la información al colisionar con ellos. Así pues, donde se ve el rombo en el centro de cada tile, se encuentra el trigger.

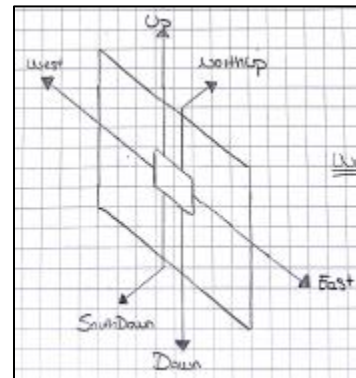
Los tiles se modelaron en MagicaVoxel. Debido al gran número de combinaciones, primero se dibujaron los mapas y cuando fueron necesarios ciertos bloques, se hicieron. La mayoría de la decoración se hizo con MagicaVoxel, excepto las piedras, antorchas y algunas flores que son sacados de la AssetStore de Unity.



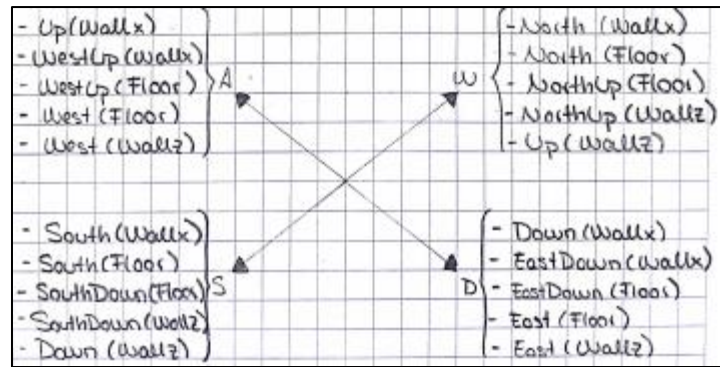
**Figura 3.1. Dibujo**  
movimientos desde suelo.



**Figura 3.2. Dibujo**  
movimientos desde pared X.



**Figura 3.3. Dibujo**  
movimientos desde pared Z.



**Figura 3.4. Agrupación direcciones por tecla.**

En las *figuras 3.5, 3.6. y 3.7* podemos ver una vista general de cada nivel. A simple vista podemos ver que el mapa es más grande pero también tienen una dificultad progresiva.



**Figura 3.5. Overview nivel 1.**



**Figura 3.6. Overview nivel 2.**



**Figura 3.7. Overview nivel 3.**



- Personaje principal: Lara

El personaje principal es el mismo que en el juego original, pero con diferente estilo. Fue modelado en MagicaVoxel y animado en [www.mixamo.com](http://www.mixamo.com).

Lara puede caminar en las 4 direcciones y escalar en 2 de ellas. Sus movimientos al caminar se pueden ver en la *figura 3.8*, y los de escalada en la 3.9.

Recoge la información de hacia dónde puede moverse al colisionar con los bloques de información de los tiles.



Figura 3.8. Animación caminar.



Figura 3.9. Animación escalar.

Su manera de acabar con los enemigos es con una patada como en la *figura 3.10*. Si hay un enemigo en la dirección del movimiento, lo matará y se moverá a su posición. Se intentó hacer con un disparo como en el original, pero en Mixamo no había ninguna animación aceptable. En un inicio la patada no hacía ningún sonido, pero posteriormente se le añadió para tener más feedback. Sabemos que hay un enemigo haciendo un Raycast hacia esa posición y se encuentra con un objeto con etiqueta de enemigo.



Figura 3.10.  
Animación atacar.



Figura 3.11.  
Animación bailar.

Al llegar al final de cada nivel se pone a bailar *La Macarena* unos segundos hasta que pasa al siguiente nivel. El motivo es que nada más empezar a animarla, vimos en mixamo las animaciones de bailar y enseguida nos vino la idea de que al acabar el nivel se pusiera a bailar. Y nos pareció una buena (y diferente) manera de finalizar un nivel. Podemos verlo en la *figura 3.11*. La posición final está diferenciada con un modelo hecho en MagicaVoxel y un sistema de partículas.

Cuando Lara muere, el nivel vuelve a empezar. Al principio Lara no hacía ninguna animación al morir: simplemente desaparecía y reaparecía automáticamente en el inicio recargando el nivel entero. Al causar poco feedback al jugador, se añadió la animación de morir de la *figura 3.12* y desaparece unos segundos más tarde.



Figura 3.12.  
Animación morir.

Las transiciones entre animaciones han sido programadas con Mecanim y se ajustaron a base de prueba y error.

Lara es quien orchestra el movimiento de las demás entidades indicando si se pueden mover o no, y en el caso del esqueleto, comunicándole hacia donde se mueve ella.

#### • Coleccionables

En cada nivel hay 4 cofres como el de la *figura 3.16*. Tres de ellos contienen una gema y el otro contiene un trozo de artefacto. Para abrirlo tan solo se debe hacer click encima del cofre desde cualquier posición del nivel dentro de un rango. Al abrir un cofre, este hará un sonido y posteriormente explotará en partículas que desaparecen al rato.



Figura 3.13.  
Gema 1.



Figura 3.14.  
Gema 2.



Figura 3.15.  
Gema 3.



Figura 3.16.  
Contenedor secreto.

Al conseguir las 3 gemas del nivel se verá encima de la cabeza de Lara la gema, que podemos ver en las *figuras 3.13*, *3.14* y *3.15*. Al conseguir un artefacto se verá el total de artefactos conseguidos hasta el momento.

Para saber qué artefactos se han conseguido ya en otros niveles, nos guardamos en el Player Preferences un entero por cada artefacto indicando si se ha conseguido ya o no.



- Enemigos

En el juego existen 3 tipos de enemigos que imitan los originales, desde un principio ya se escogió así lo que los modelos se cambiaron para hacerlos más sencillos de animar:

→ Cobra: serpiente en el juego original [Figura 3.17].

- Comportamiento idéntico: inmóvil. Si Lara se posiciona delante suyo, la matará.
- Modelada con MagicaVoxel y animada con Blender.

→ Esqueleto: lagarto en el juego original [Figura 3.18].

- Comportamiento parecido: persigue a Lara cuando la ve. Si Lara se posiciona a un bloque de distancia, la matará. Pero la detecta a distancia de 2 bloques y la persigue con una diferencia de 2 bloques.
- Modelado con MagicaVoxel y animado con [www.mixamo.com](http://www.mixamo.com).

A diferencia del original, el esqueleto no se mueve al detectar a Lara. Al principio era confuso ya que no sabías si te iba a perseguir o no. Así que se le puso una exclamación en la cabeza cuando detecta a Lara, que luego desaparece, con un sonido de alerta muy característico, haciendo un guiño a Metal Gear Solid.

→ Araña: araña en el juego original [Figura 3.19].

- Comportamiento idéntico: movimiento non-stop predeterminado. Si Lara se posiciona delante suyo, la matará.
- Modelada con MagicaVoxel y animada con Blender.

Cuando un enemigo muere, este explota en partículas de su color que desaparecen al rato. En un inicio, no hacían ninguna explosión, directamente desaparecían y podía ser que el usuario no entendiera lo que había pasado. Así que, en lugar de tener una animación de muerte, explotan para dar más dinamismo al juego.



Figura 3.17. Cobra.



Figura 3.18. Esqueleto.



Figura 3.19. Araña.

- Mecanismos

En el juego existen 4 tipos de mecanismos, dos de ellos idénticos al original, uno inspirado y otro inventado de cero. Al principio se quisieron hacer todos parecidos a los del juego original, pero debido a que nuestro mapa era “al aire libre” resultaba complicado poner el pincho de pared, entre otros.

→ Plataforma móvil: imitación del juego original [Figura 3.20].

- Comportamiento idéntico: activable desde palanca. Al activarse se coloca al nivel, y al desactivarse sube o baja.
- Reutiliza los tiles del nivel.

→ Hacha: inventado [Figura 3.21].

- Comportamiento: 4 posiciones. Se va moviendo imitando un péndulo: arriba, izquierda, abajo, derecha, arriba, derecha, abajo, izquierda, arriba.  
( ↑   ←   ↓   →   ↑   →   ↓   ←   ↑ )
- Modelado con MagicaVoxel. Animado procedimentalmente. Movido el centro de coordenadas del objeto con Blender.
- La idea surgió al pensar en las trampas de las películas de Indiana Jones.

→ Pincho: inspirado en el pincho del juego original [Figura 3.22].

- Comportamiento idéntico: activable desde botón. Al pisarse el botón, el pincho baja hasta el suelo matando a quien esté en esa posición.
- Modelado con MagicaVoxel. Animado procedimentalmente.

→ Sierra: imitación del juego original [Figura 3.23].

- Comportamiento idéntico: movimiento non-stop predeterminado. Mata a quien la toque.
- Modelado con MagicaVoxel. Animado procedimentalmente.



Figura 3.20.  
Plataforma móvil.

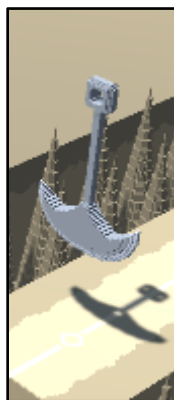


Figura 3.21. Hacha.



Figura 3.22. Pincho.

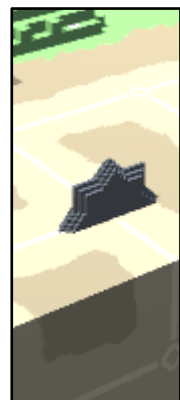


Figura 3.23. Sierra.

### • Activables

En el juego hay 2 activables también sacados del juego original, aunque uno de ellos con una apariencia totalmente distinta.

→ Palanca: imitación del juego original [Figura 3.24].

- Comportamiento idéntico: activable y desactivable al hacer click<sup>1</sup>.

Activa la plataforma móvil.

- Modelado con MagicaVoxel. Animado con Blender.

Dio problemas en un inicio ya que estaba hecho un solo objeto: palanca y base todo junto. En Blender la animación salía correcta, pero al exportarla y añadirla al proyecto, la base también se movía junto la palanca. Fue imposible arreglarlo manteniendo el objeto y se hicieron dos separados con la misma posición de eje de coordenadas para que fuera más fácil posicionarlos.



Figura 3.24. Palanca.

→ Botón: imitación del juego original [Figura 3.25].

- Comportamiento idéntico: al ser pisado, se activa. Por cualquier tipo de entidad. Activa el pincho.

- Sacado de la AssetStore de Unity.



Figura 3.25. Botón.

### • Música y sonidos

El menú principal y cada nivel tienen una canción diferente que suena de fondo.

Luego, la gran mayoría de interacciones tienen música:

- Cuando te mata una serpiente.
- Cuando te mata un esqueleto.
- Cuando te mata una araña.
- Cuando se activa y desactiva el pincho.
- Cuando se activa y desactiva la palanca.
- Cuando se mata un enemigo.
- Hacer click en un botón del menú.

Finalmente, hay algunos sonidos de ambiente como el de la cascada.

---

<sup>1</sup> El click es en el tile que contiene la palanca, para que fuera más sencillo acertar.

- Iluminación

La iluminación se ha hecho con ayuda de Aura, un asset gratuito de iluminación de la tienda de Unity.

En cada nivel hay una luz direccional que incorpora niebla y una luz ambiente de color marrón. A Lara le sigue un *spotlight* de color blanco. Y finalmente, los pinchos también tienen un *spotlight* igual al de Lara.

- Cámara

La cámara se mueve en ocho direcciones siguiendo un movimiento suave acorde a como se mueve Lara. Esta siempre se mantiene mirando al personaje a una distancia constante. Se han incorporado scripts de post procesamiento en los cuales hay *motion blur*, viñetado y corrección de color. En las *figuras 3.26 y 3.27* podemos ver el antes y después de añadir los scripts.



Figura 3.26. Cámara sin efectos.



Figura 3.27. Cámara con efectos.

## Instrucciones

Estas instrucciones se pueden ver dentro del juego desde el menú.

Teclas WASD → Moverse en referencia al mapa. Ver *figuras 4.1., 4.2. y 4.3.*

W - Hacia delante.    A - Hacia la izquierda.  
S - Hacia atrás.      D - Hacia la derecha.



Figura 4.1. Movimiento en suelo.



Figura 4.2. Movimiento en pared (X).



Figura 4.3. Movimiento en pared (Z).



Figura 4.4. Menú ingame.

El botón de la *figura 4.4.*, que aparece una vez estás jugando, vuelve al menú principal.

Para activar o desactivar las palancas o abrir los cofres del mapa como los de la *figura 4.5*, se hace click encima suyo.



Figura 4.5. Palanca y Cofre.

\*\* Teclas añadidas para facilitar corrección \*\*

Teclas O y P → Cambian de nivel cuando se está jugando.

P → Aumenta de nivel. (Ej. Del nivel 2 al 3)

O → Disminuye de nivel. (Ej. Del nivel 2 al 1)

**\*\* En la pantalla principal o de menú se debe hacer click en Play para avanzar \*\***

Tecla L → Muestra la gema del nivel.

Cabe destacar que al pulsar esta tecla puede que no funcione correctamente el nivel: no se mostrará la gema al coger los tres cofres.

Tecla V → Borra los Player Preferences con la información de los artefactos.

## Diagrama de ventanas y Flow chart

En la *figura 5.1* podemos ver el diagrama de ventanas y su Flow chart.

Al iniciar el juego aparece el menú principal. Desde él se puede acceder a jugar, a ver los controles y a ver los créditos. Siempre que se acceda a jugar se irá al primer nivel. Desde todos los niveles se puede volver al menú. El requisito para acabar el juego es pasarse los 3 niveles seguidos. Al final se verá la pantalla de créditos de ganador.

Para salir del juego se debe dar a Exit desde el menú principal.

Entre niveles se ve una pantalla de carga.

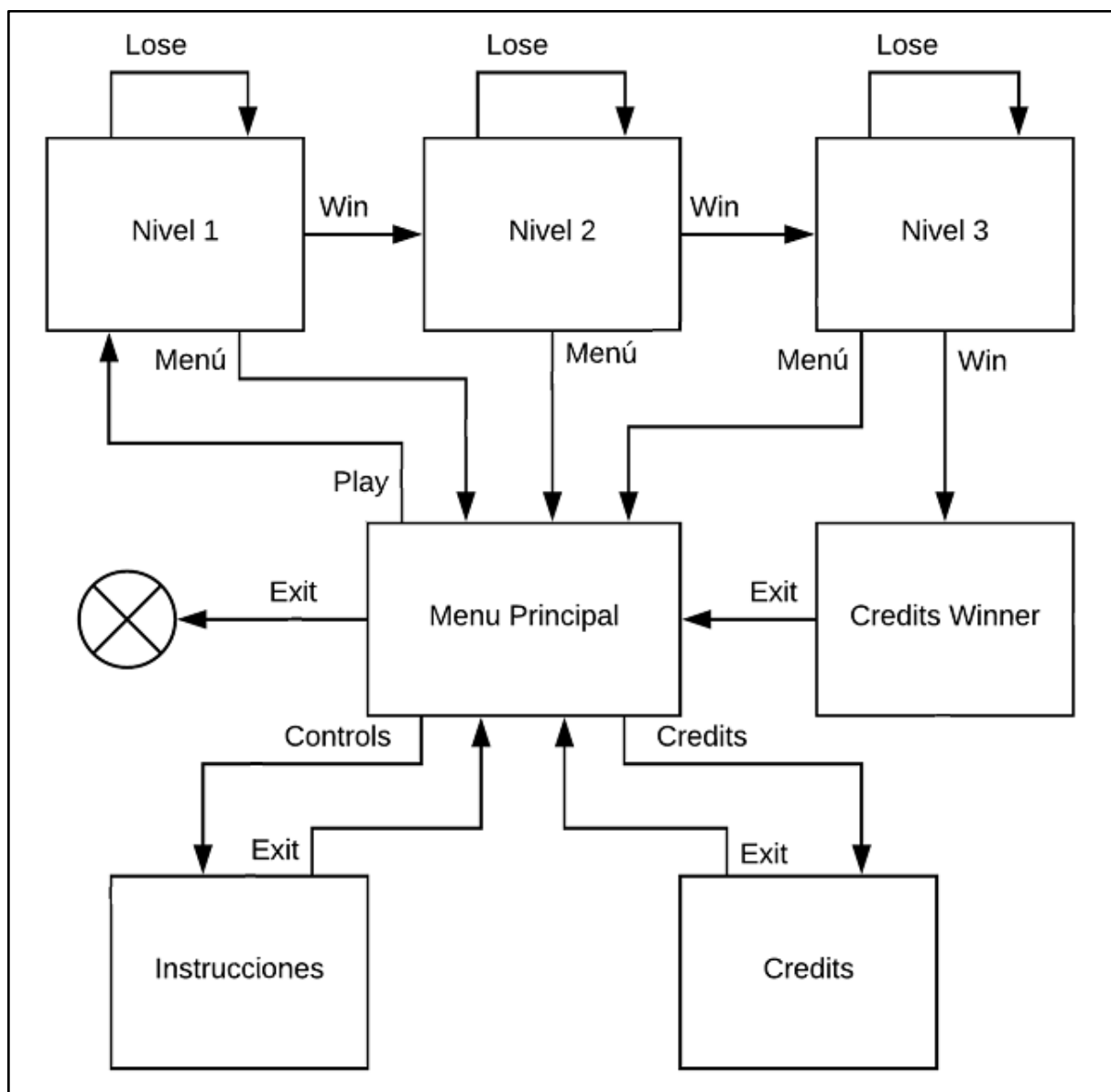


Figura 5.1. Diagrama de ventanas del juego.



# Metodología

Para hacer el proyecto utilizamos metodología Scrum. Tuvimos 6 semanas: del 21 de abril al 1 de junio. Pero decidimos tener el juego mayormente acabado para el día 29 (martes que tenemos clase de laboratorio) para así poder mostrarlo a los profesores, obtener su opinión y tener tiempo para hacer los últimos cambios. Así pues, como se puede observar en la *figura 6.1.*, repartimos las 6 semanas de manera que tuvimos:

- 2 semana de Incepción o Planificación.
- 4 semanas de Desarrollo.

Abril							Mayo																															J			
21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
Inception 1							Inception 2							Sprint 1							Sprint 2							Sprint 3							Sprint 4						

Figura 6.1. Repartición inicial.

Ambos somos de la especialidad de Ingeniería del Software así que ya estamos acostumbrados a utilizar metodologías ágiles en los proyectos así que no supuso ningún esfuerzo.

En la etapa de **Inception** nos dedicamos principalmente a entender y analizar el juego original: obtener información, jugarlo, probar situaciones, etc. Ambos compramos el juego y nos pusimos a jugarlo investigando comportamientos y anotando posibles imitaciones que podríamos hacer. También nos dedicamos a informarnos sobre qué programas íbamos a utilizar para su diseño y desarrollo y a organizar el proyecto: Git, Unity, .... Finalmente, especificamos el alcance de nuestro proyecto y escribimos las historias de usuario en el backlog.

Para hacer el backlog utilizamos la página de [tree.taiga.io](https://tree.taiga.io), que ya habíamos usado para otros proyectos, donde pudimos repartirlas en los sprints y añadirles los puntos de historia que creímos adecuados. Los puntos de cada historia se repartieron entre Front y Back. En Front era para cualquier entidad que ‘se ve o se escucha’ y en Back cuando se refería a scripts. Para organizar el backlog se crearon 7 épicas:

- |                      |                |
|----------------------|----------------|
| - Entrega.           | - Música.      |
| - Entidades móviles. | - Partículas.  |
| - Entidades fijas.   | - Iluminación. |
| - Escenas/Pantallas. |                |

La etapa de **Desarrollo** la repartimos en 4 sprints, cada uno de 1 semana. Al ser Scrum, al final de cada Sprint debíamos tener una parte del juego funcional para poder probarla y ver si nos gustaba como iba quedando. Debido a eso repartimos las tareas de manera que al final de cada sprint tuviéramos algo que mostrar a los profesores y obtener opiniones.

En el backlog tuvimos un total de 361 puntos que se repartieron en los sprints:

- Sprint 1: 120 puntos.
- Sprint 2: 84 puntos.
- Sprint 3: 107 puntos.
- Sprint 4: 50 puntos.

Como podemos ver no parecen estar “muy bien repartidos”, pero esto se debe a que decidimos esforzarnos las 3 primeras semanas más para no dejarlo todo para el último momento y así tener la última semana ya el juego casi finalizado y poder solo hacer retoques finales que nos dijeran los profesores. Aparte, al acercarse el final del cuatrimestre, también se debían entregar otros proyectos y no quisimos que nos coincidiera. En el apartado de planificación se habla del reparto de historias entre las iteraciones.

Debido a esta repartición, modificamos levemente la duración de los sprints: alargamos el primer sprint para asegurar un buen inicio y poder con todos los puntos, y acortamos el cuarto al tener menos trabajo, quedando el reparto de semanas como en la *figura 6.2*.

Abril							Mayo																															J			
21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
Inception 1							Inception 2							Sprint 1							Sprint 2							Sprint 3							Sprint 4						

**Figura 6.2. Repartición final.**

Ha habido una buena comprensión entre nosotros y un buen reparto de trabajo ya que prácticamente pasamos todo el tiempo juntos, confiamos el uno en el otro y conocemos las capacidades y los gustos del otro. Ya estamos acostumbrados a hacer trabajos juntos, sabemos que ambos buscamos la excelencia y no tenemos problema en ponernos a trabajar.

En este proyecto, para repartirnos el trabajo, uno tuvo más peso en el tema de diseño, modelado y animación de objetos, decorado y testing; y el otro se dedicó más al tema de programación y música. Aun así, cuando uno tuviera algún problema el otro ayudaría sin problema, ya fuera en hacer algo, en encontrar un error o en solucionarlo.

## Planificación y Backlog

Como antes mencionado, se utilizó Taiga para tener el backlog e irlo actualizando. Planificamos los sprints de manera que al final de cada uno tuviéramos algo que probar y enseñar para saber la opinión de los profesores. Por ello al final de cada sprint pensamos tener:

### → Sprint 1:

- Sistema de movimientos (caminar y escalar) finalizado.
- Animaciones de Lara completas.
- Dos mecanismos (plataformas y pincho) completo - *[2/4 mecanismos]*.
- Dos enemigos (cobra y esqueleto) hechos - *[2/3 enemigos]*.
- Nivel 1 formado - *[1/3 niveles formados]*.

Con ello tendríamos el Nivel 1 formado (sin decorado).

### → Sprint 2:

- Decoraciones creadas (o encontradas).
- Iluminación de Lara.
- Nivel 1 iluminado - *[1/3 niveles iluminados]*.
- Nivel 1 decorado - *[1/3 niveles decorados]*.
- Dos activables (palanca y botón) hechos - *[2/2 activables]*.
- Tercer enemigo (araña) acabado - *[3/3 enemigos]*.
- Tercer mecanismo (sierra) finalizado - *[3/4 mecanismos]*.
- Nivel 2 formado - *[2/3 niveles formados]*.
- Nivel 2 decorado - *[2/3 niveles decorados]*.

Con ello tendríamos el Nivel 1 acabado completamente y el Nivel 2 casi finalizado, pero sin luces. Habríamos hecho todos los activables y enemigos y la iluminación de Lara.

→ Sprint 3:

- Nivel 2 iluminado - *[2/3 niveles iluminados]*.
- Cuarto mecanismo (hacha) finalizado - *[4/4 mecanismos]*.
- Contenedor secreto acabado - *[1/1 contenedor]*.
- Tres gemas modeladas - *[3/3 gemas]*.
- El artefacto hecho - *[1/1 artefacto]*.
- Animación conseguir gema o artefacto terminada.
- Menú ingame colocado.
- Partículas de Polvo y Fuego finalizadas.
- Partículas al acabar con un enemigo hechas.
- Modelo de salida hecho.
- Músicas del Menú y de cada Nivel encontradas.
- Nivel 3 formado - *[3/3 niveles formados]*.
- Nivel 3 iluminado - *[3/3 niveles iluminados]*.

Al acabar tendríamos el Nivel 2 completo y el Nivel 3 casi acabado, pero sin las decoraciones. Habríamos terminado todas las entidades fijas y móviles y colocado algunos sonidos y música.

→ Sprint 4:

- Nivel 3 decorado - *[3/3 niveles decorados]*.
- Pantalla de carga.
- Pantalla de créditos.
- Pantalla de instrucciones.
- Música de Créditos.
- Sonidos de abrir contenedor secreto y de muerte.
- Documentación.
- Video.
- Preparación entrega.

Habríamos acabado todos los niveles y pantallas. Dejando para el final la documentación ya que necesitamos tener el proyecto acabado para documentarlo.

No hemos añadido capturas de nuestro Taiga ya que eran demasiado grandes y la calidad era baja. Pero hemos añadido el enlace a nuestro proyecto público de Taiga en la bibliografía o puedes encontrarlo pulsando aquí:

[https://tree.taiga.io/project/isakody-vj\\_laracroftgo/backlog](https://tree.taiga.io/project/isakody-vj_laracroftgo/backlog)

## Diagrama de Gantt y Seguimiento Taiga

El diagrama de Gantt lo dividimos en los 4 sprints y agrupamos historias parecidas para que se pueda ver mejor del gráfico, aparte añadimos el burndown del taiga y un pequeño comentario de cada sprint:

→ Sprint 1

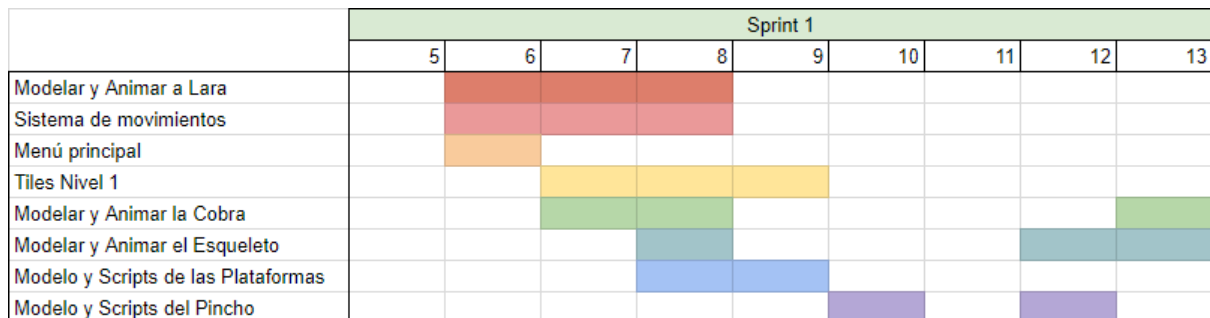


Figura 6.3. Diagrama de Gantt, Sprint 1.

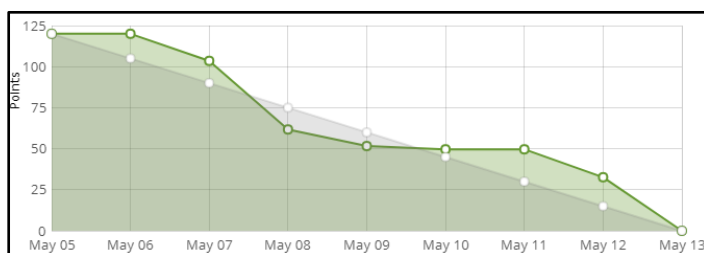


Figura 6.4. Burndown, Sprint 1.

En este sprint nos coincidió con una entrega importante de otra asignatura y por eso hay días con menos implicación como podemos ver en las *figuras 6.3. y 6.4.*

→ Sprint 2

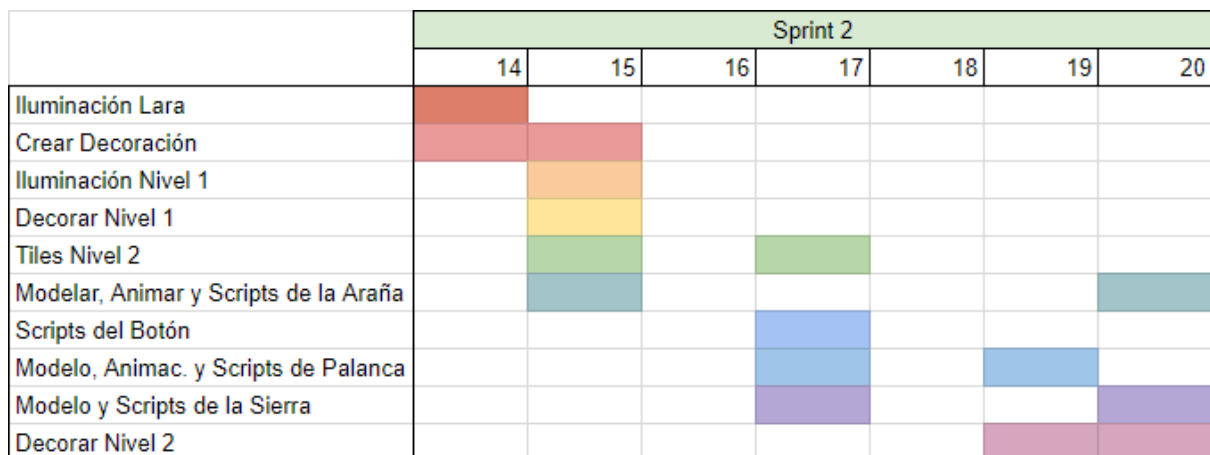


Figura 6.5. Diagrama de Gantt, Sprint 2.

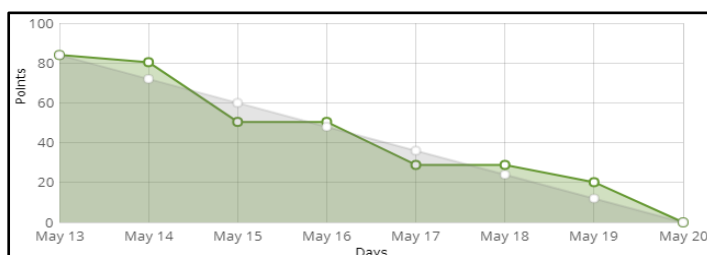


Figura 6.6. Burndown, Sprint 2.

Ya que ambos tenemos también el proyecto de final de especialidad hay días en los que debemos centrarnos en él, como se puede observar en las *figuras 6.5 y 6.6.*

→ Sprint 3

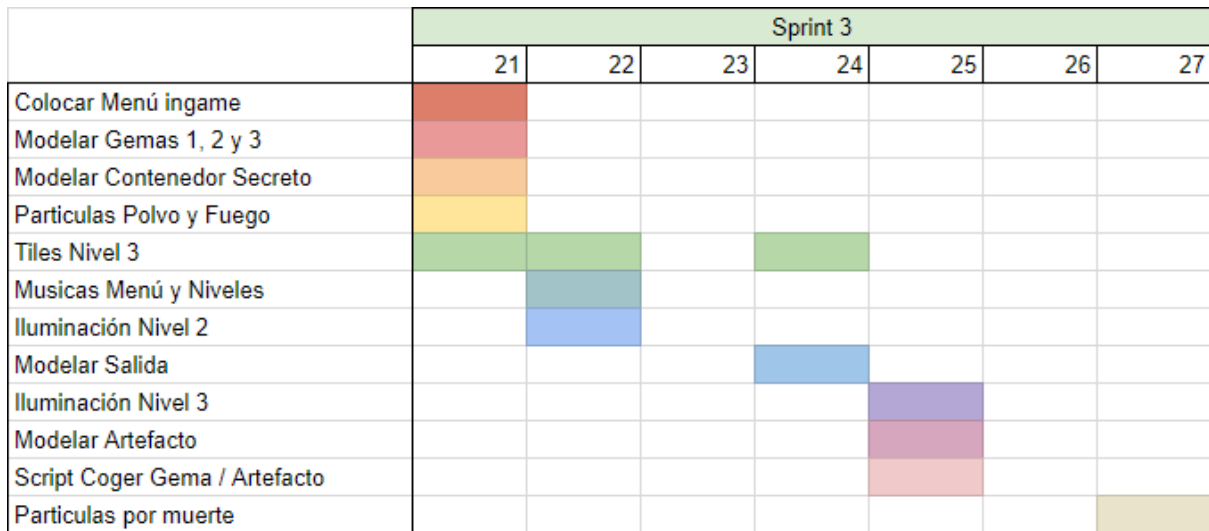


Figura 6.7. Diagrama de Gantt, Sprint 3.

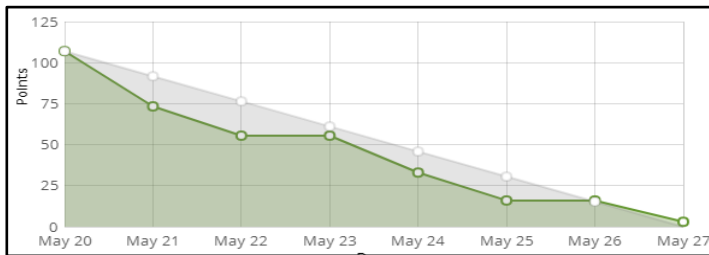


Figura 6.8. Burndown, Sprint 3.

Como se puede ver en las figuras 6.7 y 6.8, quisimos tener la mayoría del juego terminado para el día 25 que hay clase de teoría para recibir la opinión del profesor.

→ Sprint 4

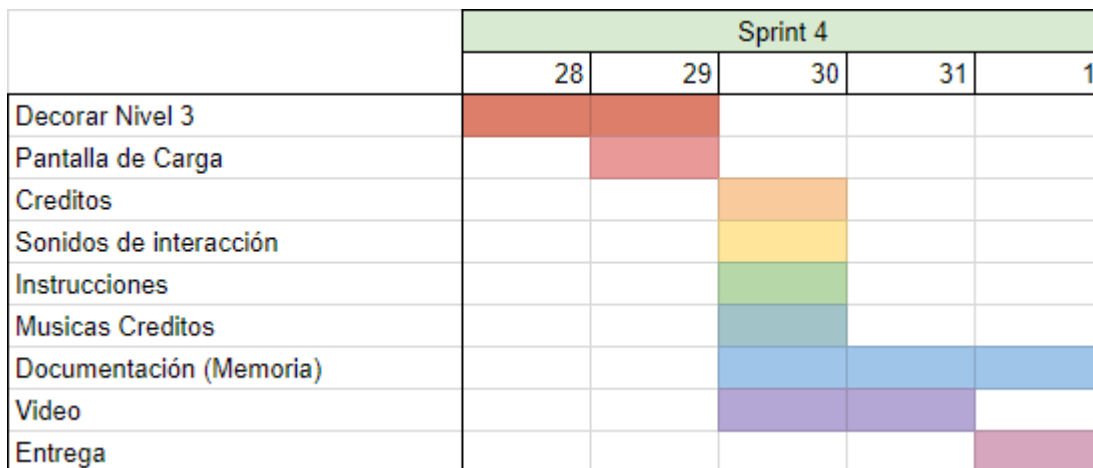


Figura 6.9. Diagrama de Gantt, Sprint 4.

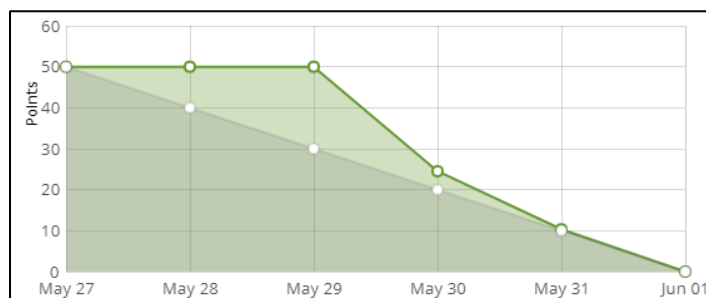


Figura 6.10. Burndown, Sprint 4.

Cabe destacar que los primeros días de la figura 6.10 son así ya que se nos olvidó poner las historias de usuario como acabadas. El reparto de trabajo se puede ver bien en la 6.9.



## Versiones

Al tener 4 sprints, tuvimos 4 versiones del juego:

→ Versión 1: Nivel 1 formado. Se podían probar las animaciones de Lara de caminar, escalar y atacar. Se probó el correcto funcionamiento de las serpientes y de ambos mecanismos, el pincho y la plataforma móvil. En esta versión encontramos varios errores de las animaciones de Lara, pero se consiguieron solucionar sin problema.

→ Versión 2: Nivel 2 acabado. Pudimos probar mejor las animaciones de escalada, ya que en el nivel 1 no había una pared importante ni movimiento lateral en escalada. Nos encontramos algún error y que el resultado no era, estéticamente hablando, muy agradable. Tampoco hubo grandes problemas en solucionarlo. En esta versión pudimos probar el buen funcionamiento de la araña y las sierras (aunque posteriormente encontramos un bug del que hablaremos en la versión 3). También pudimos añadir los activables y la iluminación. En esta versión, los profesores se dieron cuenta que nos faltaba dar más feedback al usuario en cuanto a muertes y sonidos al interactuar con entidades.

→ Versión 3: Nivel 3 formado. Pudimos probar el correcto funcionamiento de todos los mecanismos y enemigos. En esta versión nos dimos cuenta de un bug que tenían las sierras debido al Lerp al moverse: al no ser exacto se quedaban a 0,0001 de su posición final provocando que se quedaran dando vueltas de un lado a otro de la posición final sin poder llegar, muchas veces traspasando paredes o volando. Se solucionó colocando valores absolutos. En esta versión también se añadió feedback al acabar con un enemigo o cuando uno te mataba, pero no el suficiente; aún faltaban algunos sonidos.

→ Versión 4 (Final): Los 3 niveles acabados completamente. Feedback de casi todas las entidades como se mencionó en el apartado de Características principales - Música y Sonidos. Se añadieron las últimas pantallas necesarias también. Esta versión no es muy diferente de la tres, solo se añade el decorado del tercer mapa, ya que se intentó acabar el proyecto para el martes 29 para poder mostrarlo a los profesores.

## Hitos abiertos y cerrados

Como es posible observar en el Taiga, todas las historias de usuario se completaron y quedaron cerradas, indicando que todos nuestros objetivos se realizaron.

Aún así, si se hubiera tenido más tiempo, habríamos añadido animación a los árboles y habríamos mejorado la optimización.

## Puntos tratados en las reuniones

Realizábamos una reunión (Planificación Sprint) al inicio de cada sprint para detallar qué historias de usuario entraban en ella y cuantos puntos tendrían. También realizábamos la reunión Retrospectiva al final para analizar el resultado y comentar qué habíamos hecho bien y qué mal.

Al final de cada sprint también hacíamos la Demo del sprint para analizar el resultado de cada versión.

En la primera retrospectiva se habló del problema de Git al abrir el proyecto (véase apartado siguiente) y de las animaciones de Lara.

En las siguientes retrospectivas simplemente se opinó sobre el resultado y al parecernos todo correcto, se continuó.

## Método colaboración

Para compartir el proyecto empezamos utilizando un repositorio privado en Github, pero al acabar el primer sprint se hacía imposible trabajar con ello ya que el simple hecho de abrir el proyecto modificaba alrededor de 10.000 archivos. Esto causaba merge conflicts, pero la gota que colmó el vaso fue que se empezaron a perder prefabs. Entonces decidimos cambiar y probar el sistema de colaboración propio de Unity y resultó ir perfecto.

Durante los últimos tres sprints ha sido lo que hemos utilizado y no hemos tenido ningún problema, ya que se puede gestionar de manera cómoda los merge conflicts (aunque hubo pocos) y detecta solo los archivos realmente modificados.

# Conclusiones

Ambos creemos que era un proyecto complejo, pero al empezarlo desde un principio con ganas y al repartirnos bien el trabajo, se ha conseguido acabar con un buen resultado.

Hemos aprendido como funcionan los programas para desarrollar y diseñar videojuegos como Unity y Blender que nos han permitido hacernos una idea de cómo funciona el mundo de los videojuegos.

Ha sido un proyecto entretenido, aunque creemos que le hemos dedicado muchas horas diarias debido al reducido tiempo del que disponíamos para hacerlo. Aun así, las hemos empleado con buen gusto ya que la temática era interesante y la asignatura nos agrada.

# Bibliografía

- **Información Lara Croft Go**

<https://www.windowcentral.com/lara-croft-go-developer-interview>

Entrevista a un Desarrollador de Lara Croft Go para Windows Phone.

[https://en.wikipedia.org/wiki/Lara\\_Croft\\_Go](https://en.wikipedia.org/wiki/Lara_Croft_Go)

Página de Lara Croft Go en Wikipedia.

<https://www.laracroftgo.com/>

Página oficial de Lara Croft Go

- **Taiga del proyecto**

[https://tree.taiga.io/project/isakody-vj\\_laracroftgo/backlog](https://tree.taiga.io/project/isakody-vj_laracroftgo/backlog)

Nuestro proyecto en Taiga

- **MagicaVoxel**

<https://ephtracy.github.io/>

Herramienta para modelar en 3D.

- **Mixamo**

<https://www.mixamo.com/>

Página Web que permite animar personajes personales.

- **Blender**

<https://www.blender.org/>

Herramienta para realizar animaciones, entre otras cosas.

- **Tutoriales**

<https://www.youtube.com/?hl=es&gl=ES>

YouTube es una gran fuente de tutoriales.

- **Unity Tutorials**

<https://unity3d.com/es/learn/tutorials>

Tutoriales de Unity.

- **Unity Answers**

<https://answers.unity.com/index.html>

Foro de Ayuda sobre Unity.

- **Unity AssetStore**

<https://assetstore.unity.com/>

Tienda de Assets de Unity.

- **Fuentes**

<https://www.dafont.com/es/>

Página Web de Fuentes.

- **Música**

<https://www.youtube.com/?hl=es&gl=ES>

Youtube también es una gran fuente de música y sonidos.

## Anexo

Aquí adjuntamos algunas imágenes de la realización del proyecto: bocetos, capturas, ...

Primera versión inacabada de serpiente. Descartada por no parecer low-poly (*figura 7.1*)

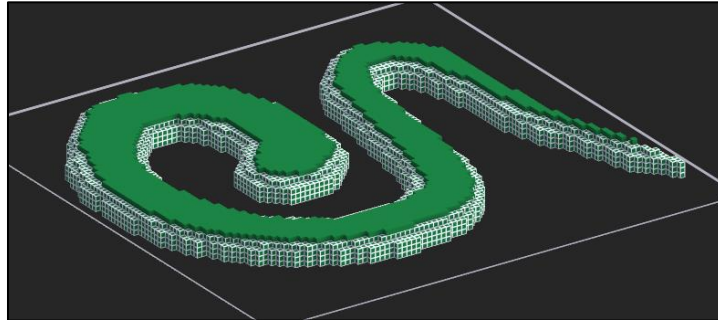


Figura 7.1. Primera versión serpiente.

Cobra con esqueleto haciendo animación de atacar en Blender (*figura 7.2.*)

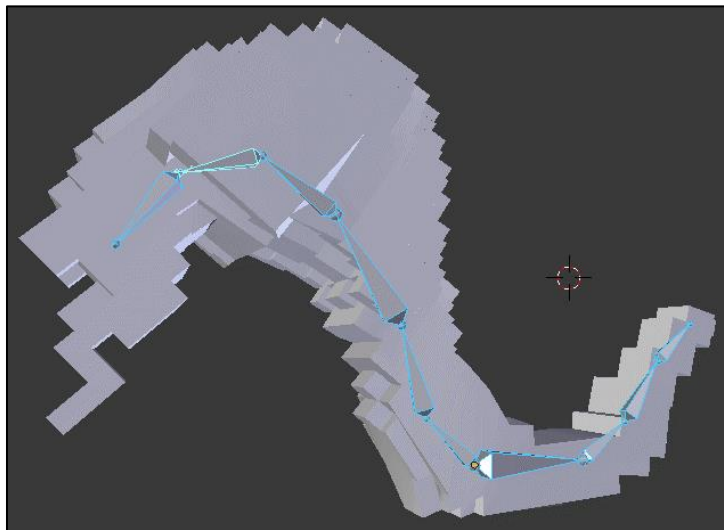


Figura 7.2. Cobra con esqueleto.



Boceto (figura 7.3), Creación (figura 7.4) y Decoración (figuras 7.5) del primer nivel:

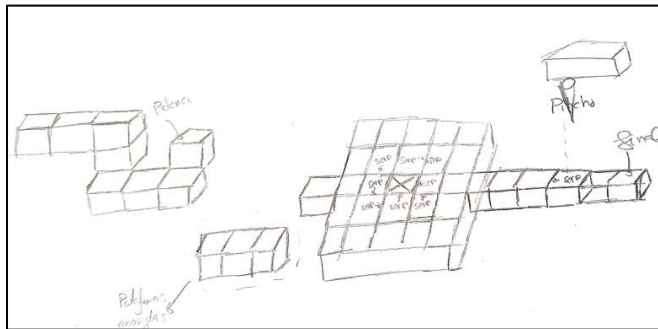


Figura 7.3. Boceto Nivel 1.

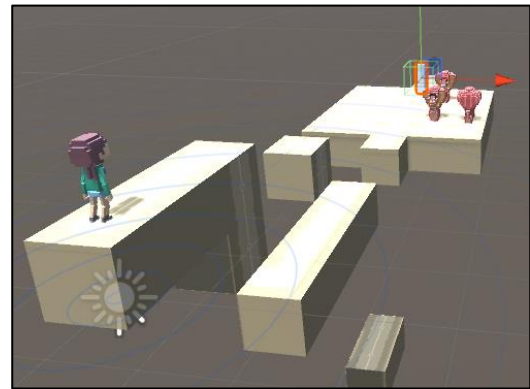


Figura 7.4. Creación Nivel 1.



Figura 7.5. Decoración Nivel 1.

Modelo de mecanismos: Versiones de Sierra (figura 7.6) y Hacha (figuras 7.7 y 7.8)

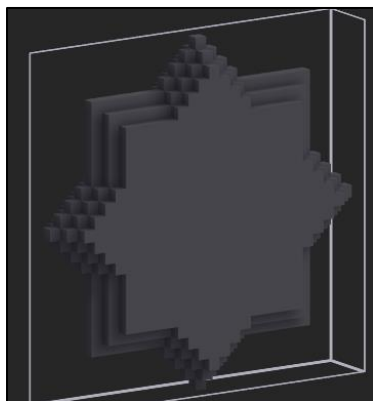


Figura 7.6. Primera versión Sierra.

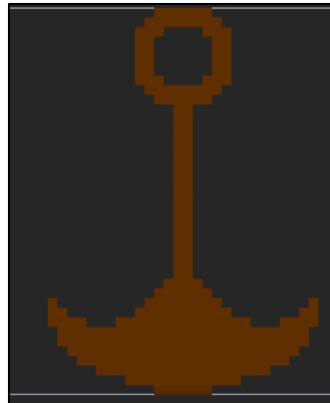


Figura 7.7. Primera versión Hacha.

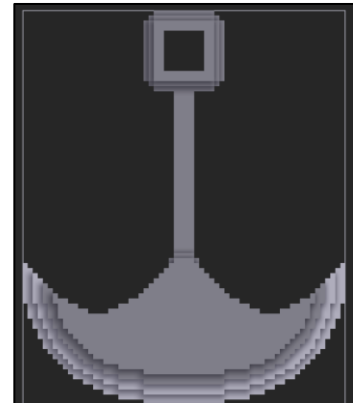


Figura 7.8. Versión final Hacha.

Decoración de niveles: Columnas (figuras 7.9 y 7.10), Elemento sin nombre finalmente no usado (7.11), Seta (7.12), Flor (7.13), Arbusto (7.14) y Puente (7.15).

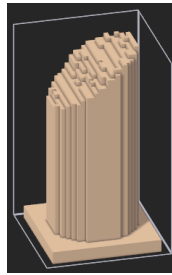


Figura 7.9. Columna Rota.

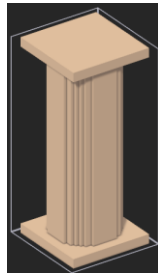


Figura 7.10. Columna.



Figura 7.11. Sin Nombre.



Figura 7.12. Seta.



Figura 7.13. Flor.

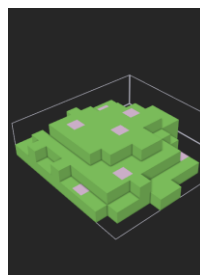


Figura 7.14. Arbusto.

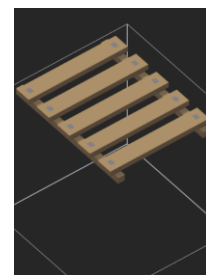


Figura 7.15. Puente.

Creación (figura 7.16) y Decoración (7.17) del Nivel 2.

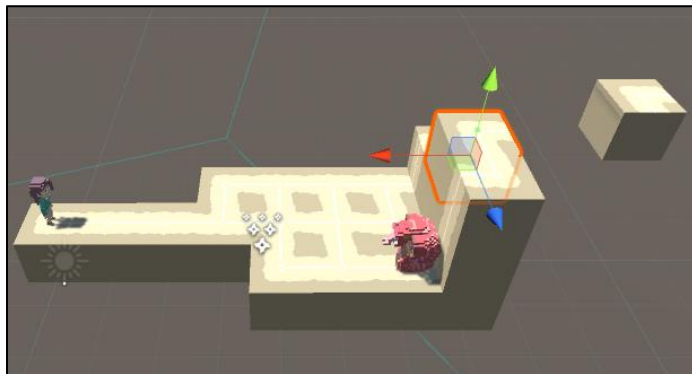


Figura 7.16. Creación Nivel 2.



Figura 7.17. Decoración Nivel 2.

Decoración nivel 3 (figura 7.18).



Figura 7.18. Decoración Nivel 3.