



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Ingeniería de Software

**Virtualización de un juego serio para el
aprendizaje de la programación**

Autor:

Luis Alberto Centeno Bragado



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Ingeniería de Software

Virtualización de un juego serio para el aprendizaje de la programación

Autor:

Luis Alberto Centeno Bragado

Tutores:

Alma María Pisabarro Marrón
Carlos Enrique Vivaracho Pascual

A mis padres, por la fuerza y el apoyo que siempre me han brindado.

A Ester, por ser el motivo que me empuja a superarme día tras día.

Agradecimientos

Muchas son las personas que han colaborado en este proyecto de alguna manera. Quisiera agradecer en primer lugar a mi tutora, Alma Pisabarro, por confiar y contar conmigo para realizar este trabajo desde el primer momento. También agradecerle las revisiones y correcciones que han hecho que el trabajo saliera adelante. También quiero dar las gracias a mi cotutor, Carlos Vivaracho, por la ayuda brindada a lo largo de todo el trabajo.

Gracias a ellos es posible realizar este tipo de investigaciones que contribuyen a la mejora de la enseñanza de la informática, ofreciendo a los alumnos nuevas metodologías para tener éxito en las asignaturas. Les animo a que no cesen nunca de trabajar en esta línea.

Resumen

Vivimos en un mundo digital, en el que las nuevas tecnologías participan activamente en todos los ámbitos de nuestras vidas. En unos pocos años hemos asistido a una auténtica revolución en el universo de los juegos. Podemos jugar con cientos de personas a la vez separadas a miles de kilómetros de distancia, compitiendo con unos y cooperando con otros. En este trabajo se unen tecnología, juegos y sistemas de aprendizaje para ofrecer un modelo de docencia que motive y ayude a los alumnos. Para ello se evalúa el uso de diversos motores para la creación de videojuegos multijugador y tecnologías de desarrollo colaborativo, en los que basarse para la creación de un producto que sirva a profesores y alumnos, aplicando técnicas de gamificación.

Abstract

We live in a digital world, where the new technologies actively participate in all the fields of live. In a few years, we have attended to an actual revolution in the games universe. We can play with hundreds of people at the same time, separated thousands of kilometres between them, competing with some of them and cooperating with others. In this project, technology, games and learning systems join together to offer a teaching model that motivates and helps the students. For this we evaluate the usage of several engines for creating multiplayer videogames and technologies of collaborative development, on which we can base to create a product that ease to teachers and students, applying gamification methods.

Índice general

| | |
|---|-----------|
| 1. Introducción | 1 |
| 1.1. Juego y Gamificación. Conceptos clave | 1 |
| 1.1.1. Qué es el juego | 1 |
| 1.1.2. El juego en la sociedad y su papel en el aprendizaje | 4 |
| 1.1.3. Gamificación | 5 |
| 1.1.4. Juego Serio | 7 |
| 1.2. Motivación | 7 |
| 1.3. Objetivos | 9 |
| 2. Contexto | 11 |
| 2.1. Precedentes - Yincana de Programación | 11 |
| 2.1.1. Descripción del juego | 11 |
| 2.1.2. Elementos del juego | 13 |
| 2.1.3. Variantes | 14 |
| 2.1.4. Evaluación de la actividad | 15 |
| 2.1.5. Conclusiones sobre la yincana | 15 |
| 2.2. Yincana virtual de programación | 16 |
| 2.2.1. Descripción del juego | 16 |
| 2.2.2. Elementos del juego | 17 |
| 2.2.3. Resultados esperados | 18 |
| 3. Estado del arte | 21 |
| 3.1. Panorama actual del videojuegos | 21 |
| 3.1.1. Videojuegos más jugados | 21 |
| 3.1.2. La comunicación en los videojuegos multijugador | 22 |
| 3.2. Juegos Serios Similares | 24 |
| 3.2.1. CodinGame | 24 |
| 3.2.2. CodeCombat | 26 |
| 3.2.3. Code Hunt | 27 |
| 4. Análisis de tecnologías | 29 |
| 4.1. Motores de videojuego | 29 |
| 4.1.1. Unity | 30 |
| 4.1.2. Blender | 31 |

| | |
|---|-----------|
| 4.1.3. Unreal | 31 |
| 4.1.4. Source Engine | 32 |
| 4.1.5. GameMaker Studio | 32 |
| 4.1.6. CryEngine | 32 |
| 4.1.7. LibGDX | 33 |
| 4.2. Plataformas para el desarrollo de software colaborativo | 33 |
| 4.2.1. Collabedit | 34 |
| 4.2.2. CodeShare | 35 |
| 4.2.3. JSFIDDLE | 36 |
| 4.2.4. SCRAPfy | 37 |
| 4.2.5. Cloud9 | 38 |
| 4.3. Comunicación en el videojuego | 39 |
| 4.3.1. Servicios de chat en videojuegos | 39 |
| 4.3.2. Servicios de VoIP en videojuegos | 39 |
| 4.3.3. Servicios VoIP externalizados | 39 |
| 4.4. Conclusiones sobre el análisis de tecnologías | 40 |
| 4.4.1. Sobre los motores de videojuego | 40 |
| 4.4.2. Sobre las plataformas de desarrollo de software colaborativo | 42 |
| 4.4.3. Sobre la comunicación necesaria para la yincana | 43 |
| 4.4.4. Resumen de las tecnologías necesarias | 43 |
| 5. Unity. Conceptos básicos | 45 |
| 5.1. Qué es Unity | 45 |
| 5.2. Estructura de clases | 46 |
| 5.3. Organización de ficheros en Unity | 47 |
| 6. Proyecto: Yincana virtual de programación | 49 |
| 6.1. Metodología | 49 |
| 6.2. Planificación inicial | 51 |
| 6.2.1. Distribución temporal | 51 |
| 6.2.2. Entorno tecnológico | 51 |
| 6.2.3. Estimación de costes | 52 |
| 6.2.4. Pruebas a realizar | 54 |
| 6.2.5. Análisis de riesgos | 54 |
| 6.3. Historias de usuario | 55 |
| 6.4. Implementación | 63 |
| 6.4.1. Sprint 1 | 63 |
| 6.4.2. Sprint 2 | 68 |
| 6.4.3. Sprint 3 | 75 |
| 6.5. Estado final del proyecto | 84 |
| 7. Conclusiones | 85 |

| | |
|--|------------|
| 8. Líneas de trabajo futuro | 87 |
| 8.1. Extensiones de funcionalidad | 87 |
| 8.2. Corrección de comportamientos inesperados | 88 |
| | |
| Apéndices | 91 |
| | |
| A. Acrónimos | 91 |
| | |
| B. Manual de instalación | 93 |
| | |
| C. Manual de uso | 95 |
| | |
| D. Contenido del CD | 97 |
| | |
| E. Enunciados de los desafíos de la yincana virtual | 99 |
| | |
| | |
| Bibliografía | 109 |

Índice de figuras

| | |
|--|----|
| 1.1. Elementos del juego. Fuente: [1]. | 2 |
| 3.1. Ejemplo de juego en modo práctica en CodinGame. | 24 |
| 3.2. Ejemplo de juego <i>Clash of Code</i> en CodinGame. | 25 |
| 3.3. Interfaz de juego de CodeCombat. | 26 |
| 3.4. Interfaz de juego de Code Hunt. Fuente: [24]. | 27 |
| 4.1. Interfaz de Collabedit. | 34 |
| 4.2. Interfaz de CodeShare. | 35 |
| 4.3. Interfaz de JSFIDDLE. | 36 |
| 4.4. Interfaz de SCRAPfy. | 37 |
| 4.5. Interfaz de Cloud9. | 38 |
| 5.1. Nivel de abstracción que proporciona Unity. | 45 |
| 5.2. Esquema parcial de <i>UnityEngine</i> | 46 |
| 6.1. Esquema de trabajo de Scrum. | 50 |
| 6.2. Escena del juego tras la implementación del primer sprint. | 65 |
| 6.3. Objeto que implementa el mapa escuela tras el primer sprint. | 66 |
| 6.4. Objeto que implementa el mapa desafio1 tras el primer sprint. | 66 |
| 6.5. Objeto que implementa el objeto del jugador tras el primer sprint. | 67 |
| 6.6. Objeto que implementa la cámara tras el primer sprint. | 67 |
| 6.7. Gráfico Burn Down tras el primer sprint. | 68 |
| 6.8. Escena del juego tras la implementación del segundo sprint. | 72 |
| 6.9. Objeto que implementa el mapa desafio2 tras el segundo sprint. | 72 |
| 6.10. Objeto que implementa el mapa desafio3 tras el segundo sprint. | 73 |
| 6.11. Objeto que implementa los puntos de teletransporte entre mapas tras el segundo sprint. | 73 |
| 6.12. Objeto que implementa el formulario de la prueba 1 tras el segundo sprint. | 74 |
| 6.13. Gráfico Burn Down tras el segundo sprint. | 75 |
| 6.14. Escena del juego tras la implementación del tercer sprint. | 80 |
| 6.15. Objeto que implementa el formulario de la prueba 2 tras el tercer sprint. | 81 |
| 6.16. Objeto que implementa el formulario de la prueba 3 tras el tercer sprint. | 81 |
| 6.17. Objeto que implementa el menú de pausa tras el tercer sprint. | 82 |
| 6.18. Objeto que implementa el controlador del juego tras el tercer sprint. | 82 |
| 6.19. Objeto que implementa el menú de pausa tras el tercer sprint. | 83 |

| | |
|--|-----|
| 6.20. Gráfico Burn Down tras el tercer sprint. | 84 |
| E.1. Mapa de la escuela. | 105 |

Índice de tablas

| | |
|--|----|
| 2.1. Porcentaje de usuarios de internet por grupos de edad. Año 2017. Fuente: [11] | 20 |
| 4.1. Comparativa de motores de videojuego | 41 |
| 4.2. Comparativa de plataformas de desarrollo colaborativo | 42 |
| 4.3. Resumen de características y herramientas empleadas para su resolución | 44 |
| 6.1. Ordenador portátil empleado durante el desarrollo. | 52 |
| 6.2. Monitor empleado durante el desarrollo. | 52 |
| 6.3. Estimación de costes del desarrollo del proyecto. | 53 |
| 6.4. Posibles riesgos durante el desarrollo del proyecto. | 54 |
| 6.5. Plan de acción ante los riesgos. | 55 |
| 6.6. Historia de usuario US-01. | 55 |
| 6.7. Historia de usuario US-02. | 56 |
| 6.8. Historia de usuario US-03. | 56 |
| 6.9. Historia de usuario US-04. | 56 |
| 6.10. Historia de usuario US-05. | 56 |
| 6.11. Historia de usuario US-06. | 57 |
| 6.12. Historia de usuario US-07. | 57 |
| 6.13. Historia de usuario US-08. | 57 |
| 6.14. Historia de usuario US-09. | 57 |
| 6.15. Historia de usuario US-10. | 58 |
| 6.16. Historia de usuario US-11. | 58 |
| 6.17. Historia de usuario US-12. | 58 |
| 6.18. Historia de usuario US-13. | 58 |
| 6.19. Historia de usuario US-14. | 59 |
| 6.20. Historia de usuario US-15. | 59 |
| 6.21. Historia de usuario US-16. | 59 |
| 6.22. Historia de usuario US-17. | 59 |
| 6.23. Historia de usuario US-18. | 60 |
| 6.24. Historia de usuario US-19. | 60 |
| 6.25. Historia de usuario US-20. | 60 |
| 6.26. Historia de usuario US-21. | 61 |
| 6.27. Historia de usuario US-22. | 61 |
| 6.28. Historia de usuario US-23. | 61 |

| | |
|---|----|
| 6.29. Historia de usuario US-24. | 61 |
| 6.30. Historia de usuario US-25. | 62 |
| 6.31. Historia de usuario US-26. | 62 |
| 6.32. Historia de usuario US-27. | 62 |
| 6.33. Historia de usuario US-28. | 62 |
| 6.34. Pruebas realizadas sobre la historia US-01. | 63 |
| 6.35. Pruebas realizadas sobre la historia US-02. | 63 |
| 6.36. Pruebas realizadas sobre la historia US-03. | 64 |
| 6.37. Pruebas realizadas sobre la historia US-04. | 64 |
| 6.38. Pruebas realizadas sobre la historia US-05. | 64 |
| 6.39. Pruebas realizadas sobre la historia US-06. | 64 |
| 6.40. Pruebas realizadas sobre la historia US-08. | 65 |
| 6.41. Pruebas realizadas sobre la historia US-09. | 65 |
| 6.42. Pruebas realizadas sobre la historia US-11. | 69 |
| 6.43. Pruebas realizadas sobre la historia US-12. | 69 |
| 6.44. Pruebas realizadas sobre la historia US-14. | 69 |
| 6.45. Pruebas realizadas sobre la historia US-15. | 69 |
| 6.46. Pruebas realizadas sobre la historia US-16. | 70 |
| 6.47. Pruebas realizadas sobre la historia US-17. | 70 |
| 6.48. Pruebas realizadas sobre la historia US-18. | 70 |
| 6.49. Pruebas realizadas sobre la historia US-19. | 71 |
| 6.50. Pruebas realizadas sobre la historia US-20. | 76 |
| 6.51. Pruebas realizadas sobre la historia US-21. | 76 |
| 6.52. Pruebas realizadas sobre la historia US-22. | 77 |
| 6.53. Pruebas realizadas sobre la historia US-23. | 77 |
| 6.54. Pruebas realizadas sobre la historia US-24. | 77 |
| 6.55. Pruebas realizadas sobre la historia US-25. | 78 |
| 6.56. Pruebas realizadas sobre la historia US-26. | 78 |
| 6.57. Pruebas realizadas sobre la historia US-27. | 79 |
| 6.58. Pruebas realizadas sobre la historia US-28. | 79 |

Capítulo 1

Introducción

1.1. Juego y Gamificación. Conceptos clave

El juego es una actividad inherente al ser humano, que le acompaña durante todas las etapas de su vida, desde que es pequeño, durante su juventud, su vida adulta y hasta la vejez. Sin embargo, no se trata de una actividad exclusiva de las personas, ya que muchos animales juegan como estrategia de aprendizaje en sus primeros meses de vida. Por ello, no resulta fácil datar el inicio del juego, pero es lógico afirmar que se trata de algo anterior a la aparición de nuestra especie [1].

1.1.1. Qué es el juego

La palabra juego, según el Diccionario de la Real Academia de la Lengua Española es un “ejercicio recreativo o de competición sometido a reglas, y en el cual se gana o se pierde”[2]. Otra definición que podemos encontrar fácilmente es la de “actividad que realiza uno o más jugadores, empleando su imaginación o herramientas para crear una situación con un número determinado de reglas, con el fin de proporcionar entretenimiento o diversión” [3].

Muchos autores a lo largo de la historia han establecido distintas definiciones de juego, como las que se enuncian en el párrafo anterior, pero gran parte de ellas recalcan el carácter lúdico y recreativo del juego, dando a entender que el objetivo principal de este es el de divertir y entretener.

Los elementos que conforman un juego están estructurados de forma piramidal (véase figura 1.1). Los elementos más básicos son los componentes. Por encima están las mecánicas del juego y en la cúspide las dinámicas. Por fuera, rodeando la pirámide, se encontraría la experiencia vivida por el jugador.

Estos elementos deben estar debidamente integrados, ya que, de no ser así, no se da lugar a un juego. A continuación se da una descripción más concreta de qué son y qué elementos concretos podemos encontrar en cada categoría.

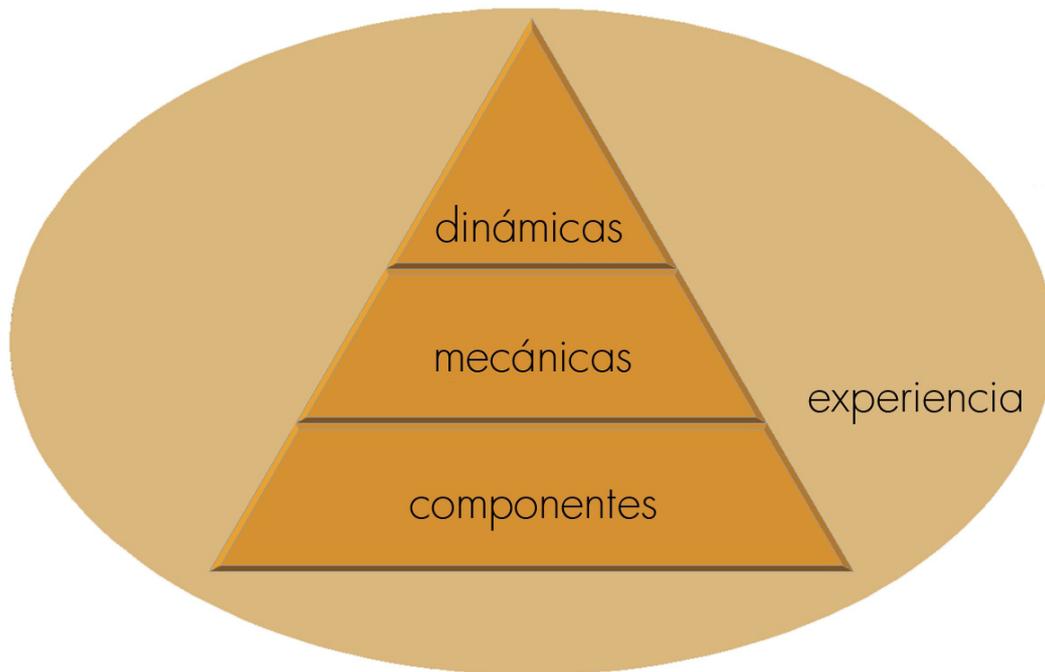


Figura 1.1: Elementos del juego. Fuente: [1].

- Las **dinámicas** son los aspectos generales, que aportan el gran enfoque al juego. Las dinámicas están formadas por:
 - Limitaciones: en muchas ocasiones resulta necesario poner límites a un juego, por ejemplo, para evitar la existencia de una única estrategia ganadora. De esta forma el juego resulta más interesante y fomenta la creatividad del jugador.
 - Emociones: al jugar es habitual apelar a ciertas emociones, no solo positivas, como pueden ser la diversión o la curiosidad. Algunas emociones negativas, como la frustración ante un reto puede resultar ser una invitación a superarse, que después se culmina con la satisfacción por haber logrado superarlo.
 - Narrativa: juega el importante papel de juntar las piezas y proporcionar una historia coherente durante el juego, creando un contexto más allá del de superar objetivos. De esta forma, se aumenta el grado de compromiso del jugador.
 - Progresión: el juego resulta más atractivo si el jugador tiene la impresión de avanzar dentro del juego, pero también fuera.
 - Relaciones: cuando un juego requiere de la participación de varios jugadores, suele ser más divertido. Esto es debido a que el ser humano es un ser social e, incluso, cuando el juego es de un sólo jugador, es habitual ver a éste compartiendo sus logros.
- Las **mecánicas** son los elementos que hacen que el jugador avance en el propio juego. Dentro de las mecánicas podemos encontrar elementos como:
 - Desafíos: tareas que requieren algún esfuerzo para ser resueltas.
 - Suerte: en muchos juegos se incluye un factor de aleatoriedad, que puede equilibrar la partida, dando más oportunidades a ciertos jugadores.

- Cooperación: modo de interacción entre los jugadores en el que buscan un objetivo común que no puede conseguirse de manera individual o es más difícil.
 - Competición: modo de interacción entre los jugadores, en el que uno o un grupo de ellos gana o pierde contra otro. Los juegos no tienen por qué ser cooperativos o competitivos y en muchos de ellos la combinación de ambos da muchas posibilidades de diversión.
 - Feedback: resulta muy importante en muchos juegos que el jugador conozca cómo está siendo su actuación y cuál es el estado del juego. Se puede llevar a cabo a través de un sistema de puntuación, pero no es la única forma.
 - Recompensas: beneficios que obtiene el jugador, como un bonus de puntos por realizar alguna acción. Generalmente gustan al jugador porque activan la dopamina en el organismo. A veces, la obtención de una recompensa determinada es la finalidad del juego en sí mismo.
 - Recursos: objetos, virtuales o no, que un jugador consigue en el juego y que son útiles para el propio juego. También se puede permitir que los jugadores negocien y hagan transacciones con dichos recursos.
 - Turnos: para establecer un orden a la hora de que los jugadores participen, en lugar de que lo hagan todos a la vez.
 - Estados ganadores: en muchos juegos se deben definir las condiciones para las cuales se considera que uno o varios jugadores han ganado.
- Los **componentes** son elementos específicos para materializar lo definido en las mecánicas del juego. Pueden ser:
- Logros: son los objetivos específicos del juego.
 - Búsquedas: similar a los logros, se trata de localizar un objeto concreto del juego que permite realizar algo que no puede realizarse sin haberlo obtenido.
 - Avatares: representaciones de personajes.
 - Puntos: representación numérica de la progresión del jugador, que también pueden considerarse como recompensas en sí mismas.
 - Insignias: representaciones de logros y nivel del jugador.
 - Combates: cualquier modo de confrontación donde se gana o se pierde, normalmente de corta duración.
 - Tablero de marcadores: muestra la puntuación o clasificación del jugador respecto al resto.
 - Gráfico social: especialmente en juegos online, en los que no hay presencia física del jugador en un mismo lugar, muestra la ubicación virtual de ellos para facilitar la interacción. También se emplea para mostrar el estado y la progresión del jugador.
 - Niveles: pasos concretos que definen cómo un jugador progresa en el juego.
 - Equipos: conjunto de jugadores que cooperan para conseguir un objetivo común determinado.
 - Bienes virtuales: es uno de los ejemplos más claro de recursos dentro del juego.

- Por último, la **experiencia** es el hilo conductor que une todo y hace que el jugador viva el juego de forma real. Es la respuesta del jugador al resto de elementos. Podemos enmarcar los siguientes elementos dentro de la experiencia:
 - Sensación: cuando se percibe una sensación agradable al jugar, causada por factores que entran por los sentidos, como que nos guste la estética o el diseño.
 - Fantasía y narrativa: cuando el juego posee una historia creada por el diseñador, o bien son los propios jugadores los que la crean a lo largo del juego. Si el juego se plantea como una realidad paralela puede resultar más interesante.
 - Compañerismo: cuando la experiencia en el juego tiene un alto contenido social, la motivación para jugar se eleva al interactuar con el resto de participantes.

Es importante destacar que los elementos situados en la parte inferior de la pirámide pueden implementar los elementos superiores. Por ejemplo, podemos favorecer la relación entre jugadores, de tal forma que cooperen, planteando limitaciones a la hora de conseguir objetivos individualmente. Además, no es necesario que un juego integre absolutamente todos los elementos descritos.

1.1.2. El juego en la sociedad y su papel en el aprendizaje

El juego aparece multitud de veces a lo largo de la historia. En la antigua Grecia, consistían en distintas competiciones para alzarse con el laurel. De la época de los romanos, además de la afición por las carreras y luchas entre gladiadores, se conoce que jugaban a populares juegos de azar como los dados. Los caballeros de la Edad Media se entrenaban para la guerra en torneos que podían llegar a ser mortales.

No es hasta mediados del Siglo XVIII, con la Ilustración, cuando se empieza a desarrollar la idea de juego como instrumento pedagógico para el futuro del ciudadano. Se utilizan para transmitir nuevas ideas, enaltecer o hacer mofa de la situación política o social.

Johan Huizinga fue un filósofo e historiador holandés que en 1938 publica su obra *Homo Ludens*, dedicada al estudio del juego como fenómeno en la cultura [1, 4]. Dicho libro menciona tres ideas fundamentales:

- El juego es una actividad que el ser humano realiza por naturaleza, libre y sin nadie que le obligue a hacerlo.
- El juego sirve para crear experiencias que, aun siendo fantasía, ayuda al ser humano a tener conciencia sobre su entorno.
- El juego acerca la competencia al ser humano, y ello a la creación del deporte, por lo que el juego es, en sí, un deporte.

El juego entra en nuestra vida desde bien pequeños, y es una herramienta tanto de diversión como de aprendizaje. El hecho de que sea divertido es la mejor motivación para nuestro aprendizaje, si lo

comparamos con otras razones impuestas de alguna manera de forma extrínseca o por factores externos. Por ejemplo, muchos niños juegan a ser papás o mamás y aprenden a reflejar el comportamiento de sus padres.

1.1.3. Gamificación

La palabra gamificación es aquella sobre la que se centra este trabajo y veremos muchas veces a lo largo de este texto. Es importante tener clara cual es la idea a la que se refiere este término.

La gamificación, o ludificación, se define como el empleo de elementos propios de los juegos en entornos que no son de juego, como puede ser la educación, el marketing, la salud y otros, para potenciar la motivación, mejorar la productividad, reforzar la conducta ante un problema y evaluar a individuos concretos.

Podemos considerar el hecho de establecer un sistema de puntos o dar premios como mecánicas de juego. Aplicar este tipo de técnicas se puede considerar Gamificación. Y tiene, como beneficio, el aumento de la retroalimentación, al haber una recompensa continua en forma de puntos a medida que el sujeto progresa; y de la motivación e implicación, al incentivar al usuario.

La gamificación se emplea en multitud de ámbitos. En marketing y publicidad se emplea para motivar y potenciar el interés del consumidor. En salud, se ha usado en pacientes de enfermedades cerebrales degenerativas para prevenir, disminuir o ralentizar el efecto que producen estas enfermedades; o también para la enseñanza o entrenamiento de cirujanos. El ámbito militar es uno de los pioneros a la hora de usar gamificación, por el hecho de otorgar insignias a soldados que han cumplido satisfactoriamente su trabajo; o también por la multitud de videojuegos que existen sobre guerra.

Este trabajo se centra en la aplicación de la gamificación en el ámbito de la educación. Más concretamente, en este trabajo se empleará para la docencia universitaria de la informática.

Hasta hoy, el uso de estas técnicas trae consigo múltiples ventajas para la enseñanza:

- **El protagonista es el alumno.** Es quien tiene que resolver los problemas del juego.
- **Permite crear distintos ritmos** para la adquisición del conocimiento y competencias. En la educación tradicional, cuando el profesor pasa al siguiente nivel, algunos alumnos pueden quedarse atrás y otros incluso aburrirse. En el juego, el alumno pasará al siguiente nivel cuando lo tenga debidamente controlado. La actividad se adapta e individualiza a la velocidad de cada alumno.
- **Fallar no es malo.** En el juego, el alumno puede fallar las veces necesarias hasta que lo logra. Por el contrario, en la educación tradicional, el fallo estigmatiza.
- **Feedback automático.** Permite que el alumno conozca su grado de progreso en la materia en cada momento. Por el contrario, en los sistemas educativos tradicionales, no se conoce el progreso hasta el final, por lo que el alumno no puede reforzar lo bueno y corregir lo malo.

- **El docente también recibe feedback** de la efectividad del juego en todo momento, por lo que puede variar la estrategia de la misma forma que lo hace el alumno.
- **Progresar es divertido.** No se trata de competir contra otros, sino también contra ti mismo.
- **Los juegos favorecen la sociabilización** de los participantes y fomentan el trabajo en equipo.
- **Jugar desarrolla la creatividad.** En los juegos hay que enfrentarse a problemas no convencionales que requieren de soluciones imaginativas, en muchos casos.
- **Aumentan la flexibilidad y autonomía frente al cambio.** Los juegos requieren que el jugador se adapte según la situación.
- **Jugar es divertido.** Muchas personas pasan horas seguidas jugando. Divertirse probablemente es el mayor motivador de nuestro tiempo.

No hay que olvidarse también de que los juegos en el aula pueden traer consigo algunos inconvenientes como los siguientes:

- **Los juegos tienden a cuantificar, reducir y clasificar.** Esto provoca que se puedan perder matices en el proceso de aprendizaje, o que la visión general del tema quede reducida a puntos demasiado concretos.
- **Los juegos pueden focalizar al jugador únicamente a ganar el juego** y se pierda el objetivo real, que es aprender. Se puede jugar mucho y no aprender nada.
- **Los jugadores pueden tener tentación de hacer trampas.** Cuando el deseo de ganar es tan grande que el jugador rompe el marco de juego aceptado por todos los participantes. Sin embargo, hay casos interesantes en el que sólo se puede ganar el juego si se hace trampas. En estos juegos se busca activar el pensamiento lateral, pero son casos muy específicos que deben tener una planificación muy cuidadosa.
- **Un juego mal diseñado puede intensificar la desmotivación** sobre la materia, lo que se convierte en aburrimiento o frustración en lugar de diversión.
- **Si el juego no se adapta a los alumnos,** algunos de ellos pueden quedar fuera del desarrollo de la clase.
- **Los juegos pueden ser adictivos,** y pueden relegar otras formas de aprendizaje y de motivación.
- **A veces se puede consumir un tiempo excesivo en el juego.** Toda actividad requiere de un aprendizaje, y si éste no se calcula bien, puede perderse más tiempo del debido en la actividad.

1.1.4. Juego Serio

Como hemos dicho, el juego tiene inherentemente un carácter recreativo, ya que su objetivo es divertir. Sin embargo, cuando, al usar técnicas de gamificación se da lugar a un juego cuyo objetivo no ha sido el de entretener, resulta necesario buscar un término que le aporte el matiz al qué nos estamos refiriendo.

Así pues, decimos que un Juego Serio es aquel cuyo objetivo no es el de divertir, sino otro distinto, como el de formar, educar o establecer un comportamiento en el jugador. Típicamente, el medio para crear y diseñar un Juego Serio es la Gamificación.

Sin embargo, emplear gamificación no implica que se de lugar a un Juego Serio, ya que un Juego Serio es, ante todo, un juego. Para explicar esto, pongamos como ejemplo un profesor que quiere que sus clases sean muy participativas, hace preguntas a sus alumnos y entrega dulces a quien conteste correctamente. Desde luego, esto no cumple la definición de juego, ya que simplemente es un profesor dando clase a sus alumnos y haciendo preguntas para que respondan. Pero empleando una técnica propia de un juego como es la entrega de una recompensa, ha logrado su propósito, que es aumentar la motivación y participación de sus alumnos.

Los juegos serios existen desde antes de la aparición de los dispositivos informáticos y electrónicos. Sin embargo, no es hasta entonces cuando podemos apreciar su verdadera proliferación. Gracias a la facilidad de distribución a través de internet propia de los videojuegos actuales, los Juegos Serios han adquirido importancia en los últimos años.

Actualmente, con los avances en el ámbito de la realidad aumentada, podemos encontrar videojuegos serios que se asemejan con mucha precisión a situaciones reales y que son capaces de hacer que el jugador viva la experiencia en primera persona. Gracias a lo cual, el Juego Serio en cuestión puede resultar aun más efectivo.

1.2. Motivación

La gamificación es un concepto en creciente auge desde su aparición hace aproximadamente diez años. Sus ámbitos de aplicación pasan por el marketing, la salud, la búsqueda de talento por parte de responsables de recursos humanos en las empresas. Y también puede aplicarse en el ámbito de la educación.

Este Trabajo de Fin de Grado se enmarca dentro del Proyecto de Innovación Docente “Gamificación en la Enseñanza Universitaria de la Informática”. En dicho proyecto se pretende estudiar y medir las ventajas del uso de técnicas de gamificación para enseñar informática en el ámbito universitario.

En cursos anteriores se han desarrollado algunas experiencias previas, centradas en la aplicación de estas metodologías, así como dirección de otros Proyectos Fin de Carrera o Trabajos de Fin de Máster. A través de dichas experiencias, se ha comprobado que el uso de elementos lúdicos hace que

los alumnos sean más participativos y receptivos.

Entre las experiencias realizadas, se encuentra la Yincana de Programación, un juego serio diseñado y creado en la Escuela de Ingeniería Informática por un grupo de profesores encabezado por Alma Pisabarro Marrón. Se trata de un torneo con pruebas prácticas y teóricas que se llevó a cabo con alumnos de la asignatura de Fundamentos de Programación del Grado en Ingeniería Informática. Esta actividad fue integrada en el sistema de evaluación de la asignatura, de modo que contó para nota. Tras la actividad, se percibió un aumento en la motivación de los alumnos, entre otros beneficios.

La actividad requirió de la presencia de los alumnos en el centro de estudios, lo cual no supuso ningún problema, ya que la enseñanza de la asignatura se realiza de modo presencial. Los alumnos acudieron a realizar la actividad de la misma forma que lo hacen habitualmente para ir a clase.

Sin embargo, la aplicación de este juego serio con estudiantes no presenciales, como pueden ser aquellos de cursos de formación online, como los MOOC, y de enseñanzas a distancia, tiene de forma intrínseca un problema: la dificultad de convocar a todos los estudiantes en un mismo lugar y en un determinado momento. Este problema hace prácticamente imposible realizar esta tarea con este tipo de estudiantes

Bien es cierto que los estudios a distancia tienen público y hay claras evidencias de que el e-learning es un modelo de enseñanza que lleva creciendo varios años, gracias a los avances en las nuevas tecnologías [5]. Sin embargo, los porcentajes de abandono en los estudios online son el doble que el de los estudios presenciales, según datos del Ministerio de Educación. Para tratar de reducir esta tasa y aumentar la motivación de los alumnos, en los estudios online se trata de evitar el uso de métodos que son más propios de la modalidad presencial. Se trata de sustituir o mejorar las presentaciones de diapositivas tradicionales por métodos audiovisuales mucho más elaborados, llegando en algunos casos a contar con productoras audiovisuales a nivel profesional e industrial [6].

Como hemos dicho, la Yincana de Programación trajo consigo un aumento de los alumnos que participaron en ella. Pero tiene el inconveniente de no poder realizarse con estudiantes de modalidades a distancia. Sin embargo, sería deseable disponer de una herramienta que permita simular la Yincana, como si de un videojuego se tratara, para poder aplicarla en el ámbito del e-learning. Con ello, podría lograrse superar los dos obstáculos que se han expuesto: la dificultad de aplicar la actividad con alumnos a distancia y aumentar la motivación de esos mismos alumnos, contribuyendo a reducir la tasa de abandono.

1.3. Objetivos

En este Trabajo Fin de Grado se trata de realizar un estudio de viabilidad para la creación de una herramienta que implemente un juego serio multijugador cuyo objetivo sea el de enseñar a programar al usuario.

Por lo tanto, podemos enumerar dos grandes objetivos:

1. Estudiar la posibilidad de la implementación de la yincana de programación de forma virtual. Esto agrupa una serie de metas:
 - Comprender los antecedentes y el contexto de la gamificación para la enseñanza de la informática.
 - Hacer un estudio exhaustivo de las herramientas que permiten crear videojuegos, analizando las ventajas e inconvenientes de cada una.
 - Dado que se trata de un videojuego multijugador, analizar el panorama de los videojuegos actuales para ver qué características pueden resultar interesantes para incorporar a nuestra herramienta.
 - Comprender el funcionamiento de las herramientas elegidas.
2. Realización de un prototipo del videojuego serio que se ha enunciado. No se trata de crear un primer producto final que implemente todas características de la yincana, sino de elaborar un producto mínimo viable con una misión clara: apoyar el aprendizaje de la programación, de manera similar a otras experiencias previas.

Además, estos dos objetivos definen una segregación temporal. La normativa que regula los Trabajos de Fin de Grado indica una duración de 300 horas. El tiempo dedicado al primer objetivo, más analítico y de investigación, es de 100 horas. Mientras que al segundo objetivo, que se trata de realizar un proyecto, se le han dedicado 200 horas.

Capítulo 2

Contexto

2.1. Precedentes - Yincana de Programación

La asignatura de *Fundamentos de Programación* [7] está dentro del plan de estudios del Grado en Ingeniería Informática que se imparte en la Escuela de Ingeniería Informática de Valladolid. Se imparte, durante el curso 2017/18, en el primer cuatrimestre del primer curso, y su carácter es de Formación Básica. Sus objetivos se pueden resumir brevemente en el siguiente: dotar al alumno de capacidad para crear programas de calidad, sencillos, eficaces y fáciles de entender y probar, y que, a su vez, empleen estructuras de datos dinámicas.

Durante el curso 2016/17, las profesoras que imparten esta asignatura deciden sustituir uno de los exámenes parciales por un método de evaluación distinto que emplease técnicas de gamificación. Con esta iniciativa, el citado equipo docente crea en la Escuela de Ingeniería Informática la ***yincana de programación***.

La yincana de programación, como su propio nombre indica, se trata de una actividad de tipo yincana. Según el Diccionario Panhispánico de Dudas [8], una yincana es “*un conjunto de pruebas de destreza o ingenio que se realiza por equipos a lo largo de un recorrido, normalmente al aire libre y con finalidad lúdica*”. Sin embargo, la yincana de programación no tiene una finalidad tan lúdica, sino que tiene el objetivo de evaluar la capacidad de programación y trabajo en equipo de los alumnos. En otras palabras, se trata de un Juego Serio.

2.1.1. Descripción del juego

En la Yincana de Programación, se divide a los alumnos en equipos de tres o cuatro personas. A cada equipo se le asigna un color o un número con el fin de separar, distinguir e identificar a los integrantes de cada equipo. Cada grupo, a su vez, elige un capitán que hará las veces de portavoz del equipo en todo momento. Todos los capitanes llevan un elemento distintivo que les identifica como tal.

La actividad se realiza en la Escuela de Ingeniería Informática, es decir, en el Edificio de Tecnologías

de la Información y las Telecomunicaciones del Campus Miguel Delibes de la Universidad de Valladolid.

Como lugar base se emplea la sala Hedy Lamarr, una gran sala polivalente en la que se puede configurar la distribución de mesas en función de las necesidades del evento. Cada equipo ocupa una mesa de las que hay en la sala. Dichas mesas deben estar debidamente separadas para evitar actos poco honestos como la copia o el espionaje entre equipos. Además, cada grupo debe disponer de un equipo informático que les permita desarrollar programas en Java.

Además de la sala Hedy Lamarr, se utilizan otros espacios como los pasillos o las zonas de despachos del Departamento de Informática, aunque en menor medida.

Los retos de la yincana se dividen en tres niveles. Cada equipo debe superar dichos niveles en orden, sin que sea posible saltarse ninguno. Al inicio de cada nivel, cada equipo recibe un sobre que contiene dos pruebas que pueden resolverse simultáneamente:

- **Prueba práctica:** Se plantea un problema cuya solución suele ser numérica o textual. El problema, aunque es sencillo, habitualmente también es suficientemente complejo como para que se precise de la construcción de un programa de ordenador. El equipo debe crear dicho programa de ordenador, que debe devolver o imprimir por pantalla el resultado del problema. De cualquier manera, el resultado debe ser apuntado en un papel.
- **Prueba teórica:** Se trata de un pequeño test de entre tres y cuatro preguntas sencillas sobre el temario que se imparte en la asignatura de Fundamentos de Programación. Una vez resuelto este test, los alumnos forman, con los identificadores de las respuestas, un código de tantas letras como preguntas haya en la prueba.

Tras la resolución de ambas pruebas, el equipo tiene el resultado de la prueba práctica apuntado en un papel. Con este último, gracias a una tabla que relaciona las distintas combinaciones de códigos con distintas ubicaciones de la Escuela de Ingeniería Informática, el equipo puede averiguar el nombre de un despacho. El capitán del equipo debe acudir a ese despacho, donde entregará el papel con el resultado de la prueba práctica. El profesor o investigador que ocupa dicho despacho validará la respuesta proporcionada por los participantes. En caso de ser correcta, se le entrega el sobre que contiene las pruebas del siguiente nivel.

El equipo puede equivocarse tanto en la prueba práctica como en la teórica. En caso de equivocarse en la prueba práctica, al validar la respuesta no obtendrá el sobre de la siguiente prueba y el capitán deberá volver con sus compañeros de equipo para corregir el programa hasta que devuelva una respuesta correcta. En caso de equivocarse en la prueba teórica, el capitán acudirá a un lugar equivocado en el que simplemente no podrá validar la respuesta que lleva apuntada en el papel. Deberá entonces volver con su equipo a corregir el test hasta resolverlo correctamente para acudir al lugar adecuado.

La Yincana tiene una temática concreta, es decir, la narrativa de los enunciados de las pruebas prácticas sigue una temática desenfadada, relacionada con alguna saga cinematográfica o literaria. También el nombre de la Yincana, los nombres de los equipos y el premio final se escogen en relación con el tema. Hasta la fecha se han diseñado y realizado Yincanas con distintas temáticas: *El Torneo de*

Los 011 Magos tuvo temática relacionada con *Harry Potter*, o *Los Códigos del Hambre* estaba basado en la saga *Los Juegos del Hambre*. Además, otras yincanas celebradas posteriormente han tenido otros hilos argumentales, como la denominada *Juego de Códigos*, basada en la serie *Juego de Tronos*, o *The Programming Dead* basada en *The Walking Dead*.

Por último, la Yincana tiene una duración aproximada de tres horas. Durante este tiempo, los alumnos tienen que superar los retos que se les propone, ya que la duración de la prueba ha sido decidida de forma deliberada y acorde a la necesidad que suponen los retos. No obstante, para aquellos alumnos que no superen todos los retos propuestos, se tiene en cuenta el orden en que van superando los niveles de los que consta la prueba.

2.1.2. Elementos del juego

La primera yincana de programación se celebró el 18 de Octubre de 2016 bajo la denominación de “*Torneo de los 011 magos*” [9] con un grupo de alumnos de la asignatura de Fundamentos de Programación. En el epígrafe 1.1.1 se indica lo que es un juego y los elementos de los que puede componerse. Específicamente, en la yincana de programación se emplearon ciertos elementos de juego que se describen en este apartado.

Las dinámicas del juego fueron:

- Sólo había **limitaciones** temporales (3 horas).
- Los alumnos experimentaron diversas **emociones** durante el juego: curiosidad, creatividad, competitividad, frustración, felicidad...
- La **narrativa** del juego seguía el hilo conductor de las pruebas del libro *Harry Potter y el Cáliz de Fuego* adaptado a las pruebas de la yincana.
- Se establecieron **relaciones** en los alumnos que han perdurado durante el curso.

Respecto a las mecánicas de juego que se utilizan en la yincana, se encuentran:

- **Cooperación**, ya que los jugadores participan en equipo.
- **Competición**, dado que sólo podía ganar un equipo.
- **Feedback**. Los jugadores conocían los fallos que cometían en cada prueba. Además, tras la realización de la actividad, se dedicó una hora de clase a la corrección de los ejercicios propuestos en la yincana.
- **Recompensas**. El equipo ganador obtenía un punto extra en su nota y la copa del torneo. Además, todos los participantes fueron recompensados con una medalla del color y nombre de su equipo.

- **Estados ganadores** definidos antes de empezar el juego.

Los componentes del juego que se usaron son:

- **Logros.** El objetivo del juego era ser el primer equipo en lograr la copa.
- **Avatares.** Cada equipo podía elegir el grupo del universo *Harry Potter* con el que mejor se identificara.
- **Niveles.** Fueron definidas tres pruebas para poder progresar en el juego.
- **Equipos.** Los jugadores participaban en grupos de tres o cuatro personas.
- **Búsquedas.** Los participantes tenían que buscar elementos concretos dentro del juego.
- **Gráfico social.** Para que todos los participantes pudieran conocer en qué punto del juego se encontraba cada equipo, estos debían mantener sus sobres a la vista, ya que cada prueba tenía un color diferente gracias al cuál se podía identificar cada nivel.

2.1.3. Variantes

Durante las distintas ediciones, se han aplicado pequeñas modificaciones a la mecánica de la actividad. Principalmente, podemos distinguir entre las siguientes:

- **Sin penalización:** en esta versión, que es la tradicional, se permite que los participantes se equivoquen las veces que sea preciso hasta que resuelvan correctamente las distintas pruebas.
- **Con penalización:** en esta variante, todos los equipos reciben un número limitado de tarjetas al principio de la actividad. Después, llegado el momento de validar una prueba práctica en alguno de los niveles, si la solución a ésta es errónea, el capitán del equipo deberá entregar una tarjeta. En caso de que un equipo pierda todas sus tarjetas, el equipo no podrá continuar superando pruebas y calificará en último lugar. Dicho de otra forma, se establece un sistema de “vidas” similar al de los videojuegos tradicionales.

Durante el curso 2017/18, se celebró una yincana en la que se separan tres grupos de alumnos. Un primer grupo realiza la yincana tradicional, sin penalizaciones, mientras que otro participa con el sistema de penalización, y el tercer grupo realiza un examen convencional sobre los mismos contenidos con los que se realiza la yincana. El objetivo de ello fue medir empíricamente las diferentes ventajas e inconvenientes de estas experiencias.

En este trabajo nos centramos en la variante tradicional, sin ningún tipo de penalización.

2.1.4. Evaluación de la actividad

Al ser la primera vez que se realizaba una actividad de este tipo, se decidió emplear mecanismos simples de evaluación, como entrevistas a posteriori con participantes y profesores.

La valoración fue muy favorable y positiva en la que los participantes se involucraron en el juego, aprendieron y se divirtieron. Cuando se les preguntó, reflejaron diversas ideas:

- Haber puesto en práctica conocimientos impartidos en clase.
- Haberse divertido mientras han aprendido.
- Agradecimiento por la organización del evento.

Además, tuvo que organizarse una segunda yincana, debido a la petición de los alumnos que no participaron en esta primera edición por no formar parte del grupo de primero seleccionado.

Los alumnos cambiaron de forma positiva su actitud hacia la programación y su motivación se incrementó notablemente, haciéndose patente en las clases posteriores.

Tras la realización de estas ediciones, se decidió llevar este trabajo a las XXIII Jornadas sobre la Enseñanza Universitaria de la Informática celebradas en Cáceres en Julio de 2017 [10]. En dichas jornadas, este trabajo resultó estar entre los seis mejores que se presentaron.

Posteriormente, se elaboró otro trabajo basado en la yincana de programación, que se presentó en la siguiente edición de estas jornadas, las XXIV JENUI, que se celebran en Barcelona en Julio de 2018. De nuevo, fue elegido como uno de los seis mejores trabajos que se presentaron.

2.1.5. Conclusiones sobre la yincana

El juego serio se ha diseñado para la adquisición de conocimientos de un tema concreto, las estructuras de control iterativas. Tras realizarse la actividad, se pudo comprobar que el juego facilitó el proceso de aprendizaje y la predisposición de los estudiantes.

Además, al docente le sirvió para detectar qué errores cometen sus alumnos mientras programan. Por ejemplo, muchos alumnos se encontraron problemas cuando manejaban tipos de datos grandes o con errores de redondeo.

Se puede concluir que el uso de elementos lúdicos en el aula proporciona una serie de ventajas muy útiles. La gamificación aporta a los docentes nuevas herramientas que pueden ser aprovechadas para que los alumnos sean más receptivos y participativos.

2.2. Yincana virtual de programación

Dada la repercusión y el éxito de la yincana de programación que describíamos en el apartado anterior, el Departamento de Informática, a través del Proyecto de Innovación Docente *Gamificación en la Enseñanza Universitaria de la Informática*, desea disponer de un videojuego serio con características similares. El objetivo de esto es poder aplicar técnicas de gamificación que permitan que los participantes no necesiten estar en un mismo lugar.

Se tratará de un videojuego serio que llamaremos “Yincana Virtual de Programación”. Este videojuego permitirá la participación de estudiantes a distancia y online, con los beneficios que ha aportado la yincana de programación a los estudiantes presenciales de Fundamentos de Programación.

2.2.1. Descripción del juego

Se trata de un videojuego que reproducirá la yincana de programación presencial, de forma virtual. Por lo tanto, debe tener unas características similares a lo que ya se ha experimentado. Estas características son clave, y componen una serie de requisitos indispensables:

- **Mundo virtual acorde a la narrativa**

Puede parecer obvio, pero este juego debe ser y tener aspecto de videojuego. Su estética tiene que acompañar a la narrativa y temática de los desafíos, para que el jugador pueda buscar elementos virtuales dentro del mapa de juego. De esta forma se propicia que el jugador se meta el su papel, comprenda adecuadamente los desafíos y progrese con coherencia.

- **Comunicación entre integrantes de un equipo**

El juego debe incorporar una solución que permita que los integrantes se comuniquen de forma instantánea, por ejemplo, a través de algún servicio de voz IP.

- **Desarrollo cooperativo**

El videojuego debe incorporar una herramienta a través de la que los jugadores que integran un equipo puedan editar código de forma conjunta y colaborativa.

- **Comunicación profesor-estudiante**

Se debe incorporar una herramienta que ponga en contacto al profesor con cada equipo. Llegado el caso, los alumnos pueden preguntar dudas al profesor si se sienten atascados porque no entienden algo. También, el profesor puede ayudar a que el juego avance en caso de que la mayor parte de los equipos encuentren algún obstáculo con el que no hayan contado.

- **Gráfico social**

El juego tiene que contar con un elemento que informe al jugador del estado y nivel de sus contrincantes. Esto puede implementarse, por ejemplo, con una tabla de clasificación.

2.2.2. Elementos del juego

En este apartado se describen los elementos de juego involucrados en la yincana virtual de programación.

Las dinámicas del juego son:

- Las **limitaciones** serán únicamente temporales.
- Los alumnos experimentarán **emociones** similares a la yincana convencional: creatividad, curiosidad, competitividad, felicidad, frustración, etc.
- La **narrativa**, junto con la estética del juego, seguirá la temática del universo *Harry Potter*.
- Establecerá **relaciones** entre los participantes. Bien es cierto que se trata de alumnos desubicados geográficamente, por lo que las relaciones entre ellos no son tan profundas como si se vieran a diario en clase. Pero los vínculos que se creen a raíz de este juego pueden llegar a ser suficientes como para que puedan ayudarse durante el resto del curso, mitigando la frustración y aumentando la motivación.

Las mecánicas de juego que se emplearán son:

- **Cooperación**, dado que los participantes participarán en equipo.
- **Competición**, porque sólo podrá ganar un equipo.
- **Feedback**. Los jugadores permanecerán informados de su estado y progresión en el juego continuamente y podrán conocer sus fallos en el momento de cometerlos.
- **Recompensas**. Los jugadores recibirán una insignia en la plataforma de e-learning empleada.
- **Estados ganadores**, que serán conocidos antes de comenzar el juego.

Por último, los componentes del juego incluirán:

- **Logros**: el objetivo del juego será ganar la copa, en este caso, virtual.
- **Avatares**: los jugadores podrán configurar su personaje dentro del juego como quieran.
- **Niveles**: se definirán tres pruebas en las que los jugadores progresarán dentro del juego.
- **Equipos**: los participantes participarán en grupos de tres o cuatro personas.
- **Búsquedas**: los participantes tendrán que encontrar ciertos elementos virtuales escondidos en el juego.
- **Gráfico social**: necesario para que los participantes conozcan el estado de sus oponentes en el juego.

2.2.3. Resultados esperados

Esta herramienta permitirá que alumnos de modalidades a distancia participen en una actividad docente distinta a lo convencional. Esto no excluye que se pueda celebrar con alumnos presenciales si así se desea. En cualquier caso, se esperan las siguientes ventajas:

- **Aumento de la motivación de los alumnos.** De la misma forma que ocurre en la versión presencial, se espera un aumento en la motivación de los participantes, lo cual puede repercutir en un **descenso del abandono** del curso a distancia.
- **Progresión personalizada** para cada alumno, de forma que pase al siguiente nivel una vez que domina el problema actual.
- **Aprendizaje a través del fallo.** Al cometer el fallo dentro de un juego, la sensación de frustración se mitiga si lo comparamos con un fallo cometido en un cuestionario. De esta manera, el alumno vuelve a enfrentarse al problema con más determinación y sabiendo qué errores no debe volver a cometer.
- **Feedback en tiempo real para el alumno.** Los alumnos podrán ver su propio progreso y el de los demás en cada momento, sin necesitar la corrección de ningún profesor.
- **Feedback en tiempo real para el profesor.** El profesor podrá monitorizar a sus alumnos mientras se lleva a cabo la yincana virtual.
- **Aumento de la sociabilización.** Es cierto que en los cursos a distancia es difícil que los alumnos se relacionen entre ellos. Con la yincana, al participar por equipos, se espera que los participantes socialicen entre ellos, de forma que cooperen entre ellos durante el resto del curso. Esto también potenciará el no abandono del curso, ya que pueden motivarse entre ellos.
- **Desarrollo de la creatividad.** Se espera que los alumnos mejoren sus habilidades creativas a la hora de enfrentarse a un problema.

Sin embargo, no podemos olvidar los inconvenientes y problemáticas que podemos esperar, y no sólo deberemos esperarlos. También se debe preparar un plan de contingencia para contrarrestar su efecto. A continuación se enumeran algunos de esos problemas y la solución que se propone.

- **Tentación de hacer trampas:** cuando el deseo de ganar por parte del participante es muy alto, puede llevarle a romper ciertas reglas, como buscar respuestas en internet, o que se produzcan plagios y copias entre distintos equipos. Ante esta problemática es difícil plantear una solución que evite estos actos, puesto que se trata de jugadores que participan desde sus casas y no se les puede controlar. En este sentido, la yincana no deberá tomarse como una actividad calificable y que cuente para nota. Únicamente podrá emplearse como una herramienta más de enseñanza a disposición del docente.

- **Problemas de disponibilidad del alumnado:** al tratarse de alumnos a distancia, es difícil poder convocarles para celebrar la actividad en una misma fecha y hora. Para dar solución a este problema, se pueden realizar distintas convocatorias para realizar la actividad, en distintos horarios: de mañana, tarde y noche. Incluso, se podría plantear la opción de dejar la actividad abierta a la participación para que sea el estudiante el que decida en qué momento hacerlo, y establecer un formato contrarreloj de la prueba.
- **Disponibilidad del sistema informático:** los cursos a distancia suelen tener un número de plazas ilimitado o, en cualquier caso, contar con una cantidad de alumnos muy alta, como es el caso de los MOOC. Si no se piensa bien la infraestructura en la que se despliega el juego, puede caerse si todos los alumnos se conectan a la vez a realizar la actividad. Como hemos visto en el punto anterior, esto es poco probable, pero sigue siendo posible y debe tomarse en cuenta. Una solución a este problema puede ser, como en el punto anterior, abrir diferentes convocatorias para que los alumnos escojan en cuál participar, con un número máximo de participantes por convocatoria.
- **Disponibilidad de conexión a internet por parte del alumno:** para esta actividad es necesario que el alumno disponga de ordenador y conexión a internet. Según la encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares del año 2017, TIC-H 2017 [11], llevada a cabo por el Instituto Nacional de Estadística, el 83,4 % de los hogares españoles tiene acceso a internet. Por grupos de edad, destaca que el uso frecuente de internet (al menos una vez a la semana) es prácticamente universal (96,2 %) y va descendiendo paulatinamente conforme aumenta la edad. A partir de los 55 años este porcentaje desciende notablemente y termina por alcanzar un mínimo en el grupo de edad de 65 a 74 años (38 %). Por otro lado, la edad media de estudiantes a distancia en la UNIR, una de las universidades online a distancia presentes en nuestro país, es de 34 años [6]. Por lo tanto, no parece que la disposición de una conexión a internet y un ordenador por parte del alumno sea un problema. Además, se debe tener en cuenta que para matricularse en una modalidad a distancia es conveniente contar con estas prestaciones. Así pues, se asume que el alumno contará con conexión a internet y ordenador en su casa.

Otro asunto dentro de este mismo punto es la continua disponibilidad de esta conexión, al menos en los momentos en los que se va a realizar la actividad. Se pueden producir cortes por parte del proveedor de servicios de internet de forma inesperada. Ante este problema, podemos recurrir a una solución ya planteada a otros problemas, que es ofrecer convocatorias a diferentes horas para que, el alumno que no pueda participar por este tipo de imprevistos, tenga otras oportunidades. Otra solución podría ser poner a disposición las instalaciones del centro educativo, en este caso, las de la Escuela de Ingeniería Informática de Valladolid.

2.2. Yincana virtual de programación

| | Han utilizado Internet en los tres últimos meses | Usuarios frecuentes de Internet (al menos una vez por semana) |
|-----------------|--|---|
| Total | 84,6 | 80,0 |
| De 16 a 24 años | 98,0 | 96,2 |
| De 25 a 34 años | 96,3 | 93,7 |
| De 35 a 44 años | 95,8 | 91,9 |
| De 45 a 54 años | 90,3 | 84,9 |
| De 55 a 64 años | 73,9 | 66,4 |
| De 65 a 74 años | 43,7 | 38,0 |

Tabla 2.1: Porcentaje de usuarios de internet por grupos de edad. Año 2017. Fuente: [11]

Capítulo 3

Estado del arte

3.1. Panorama actual del videojuegos

Los videojuegos se han convertido en una fuente más de entretenimiento. Tanto es así, que algunos títulos llegan a superar en ingresos a grandes producciones cinematográficas.

Según una encuesta realizada por la Interactive Software Federation of Europe realizada entre europeos de entre 16 y 64 años, el 48 % de las personas de estas edades juega a videojuegos. En España, esta cifra se sitúa en un 40 %. Además, el 55 % de los jugadores son menores de 35 años, por lo que no es una industria reservada a gente joven como quizá fue cuando aparecieron los primeros videojuegos. Por sexos, el 55 % de los jugadores son hombres y el 45 % mujeres [12].

Existen eventos dedicados al videojuego, bien sean exposiciones sobre las novedades del sector, como concentraciones de jugadores verdaderamente multitudinarias. Estas concentraciones son conocidas como LAN Party, donde se reúnen los participantes con sus ordenadores para intercambiar contenido, hablar de tecnología y, sobretodo, jugar. En España una de las más populares es la *Euskal Encounter*, celebrada cada año en el País Vasco y cerca de 10000 participantes. En Castilla y León se celebra cada año la Netwired, en Arroyo de la Encomienda, con algo más de 300 participantes en la última edición.

3.1.1. Videojuegos más jugados

Existen un sinfín de títulos disponibles para jugar, pero siempre hay videojuegos que son más populares. En este epígrafe vamos a repasar algunos de ellos, centrándonos en los juegos disponibles para ordenador [13].

- **Counter Strike: Global Offensive**

Es un videojuego multijugador de disparos en primera persona, desarrollado por Valve y lanzado en 2012. Dispone de varios modos de juego que mezclan el estilo cooperativo y competitivo,

llegando a participar hasta 10 jugadores en una misma partida [14].

- ***PlayerUnknown's Battleground***

Fue publicado en 2017 y fue uno de los más exitosos en ventas. Es un videojuego multijugador, desarrollado por Bluehole, en el que en cada partida participan 100 jugadores en un mapa que se va haciendo cada vez más pequeño. El jugador debe encontrar herramientas con las que defenderse e ir acabando con el resto, ya que el último jugador vivo gana la partida, aunque también permite asociarse en clanes [15].

- ***Fortnite***

También fue lanzado en 2017 y es muy famoso. El modo de juego es el mismo de *PlayerUnknown's Battleground*, aunque con distinto motor y distinto estilo gráfico. Además es gratuito y está desarrollado por Epic Games [16].

- ***Grand Theft Auto Online***

Es un videojuego multijugador de acción y mundo libre desarrollado por Rockstar North. En él, los jugadores deben ir subiendo su nivel mediante puntos de reputación, que se consiguen llevando a cabo misiones que se realizan en equipo [17].

- ***Minecraft***

Es un juego de construcción, de tipo mundo libre, desarrollado por la empresa Mojang AB. Tiene distintos modos de juego en los que los participantes cooperan o compiten en función del modo elegido [18].

- ***Hearthstone: Heroes of Warcraft***

Es un juego de cartas coleccionables online, creado por Blizzard y gratuito. En este juego las partidas son de uno contra uno, por lo que es competitivo [19].

- ***League of Legends***

Es el videojuego más popular del mundo, desarrollado por Riot Games, que además domina la escena de los eSports. Es gratuito y en él se enfrentan dos equipos de cinco jugadores cada uno, en el cada equipo debe conseguir destruir las defensas del otro. Para ello, los jugadores deben cooperar y seguir una estrategia [20].

Como hemos visto, la gran mayoría de estos títulos disponen de opción multijugador, tanto en modo competitivo, como cooperativo, y muchos incorporan una mezcla de ambos modos que hacen que el juego sea más divertido.

No es de extrañar que los videojuegos multijugador sean los más jugados, ya que está en nuestra naturaleza, como ser humano, el vivir en sociedad y compartir las experiencias.

3.1.2. La comunicación en los videojuegos multijugador

Hemos visto cuáles son los videojuegos más jugados y hemos visto que muchos de ellos son multijugador. Resulta interesante ver cómo los jugadores se relacionan dentro del juego y qué herramientas utilizan.

Podemos encontrar, dentro del videojuego, herramientas que favorecen la comunicación. Normalmente podemos encontrar lo siguiente:

- **Chat en el juego**, gracias al cuál permite que los jugadores se comuniquen mediante mensajes. Además, en muchas ocasiones se emplean para manejar opciones más avanzadas, como realizar votaciones para cambiar el mapa o el escenario de juego, como es el caso de *Counter Strike: Global Offensive*. Por otra parte, cuentan con una importante desventaja, y es que hay que dejar de jugar para escribir el mensaje.
- **Servicio de voz**, por el que hablar con todos los jugadores de la partida, o con aquellos que pertenezcan al mismo equipo. También permite restringir la comunicación si el jugador ha sido eliminado de la partida, para evitar que el equipo haga trampas obteniendo pistas del compañero eliminado, que usualmente puede ver el resto de la partida libremente. Usualmente, es necesario presionar una tecla para poder hablar, de tal forma que no se está en constante comunicación y hay que desviar uno de los dedos del teclado para poder hablar.

Sin embargo, muchos jugadores no emplean por completo las herramientas que el videojuego pone a su disposición, y deciden emplear otros servicios externos, como Skype o Discord. Esto ocurre especialmente cuando juegan con amigos o si suelen jugar con las mismas personas. Empleando este tipo de herramientas se consigue una serie de ventajas:

- Se consigue **independencia del juego**, por lo que, si se quiere cambiar a otro juego, la comunicación no se corta.
- **No limita** durante el juego, ya que el programa que se emplee para la llamada grupal queda en un segundo plano. No es necesario presionar ningún botón ni distraerse de la partida para escribir un mensaje.
- **La comunicación es continua.**
- Los servicios externos a los videojuegos ofrecen una **mejor calidad** de comunicación.

Sin embargo, al usar este tipo de herramientas, es posible que uno de los jugadores juegue el rol de espectador en la partida y de pistas a sus compañeros, en lo que se entiende como un acto poco honesto. En este sentido, se delega la responsabilidad en los jugadores.

Por lo tanto, usar un servicio externo, en lugar de incluir un medio de comunicación dentro de nuestro juego para la yincana, puede tener sentido.

3.2. Juegos Serios Similares

En esta sección se aborda el estudio de los Juegos Serios Similares que existen, destacando los más conocidos, sus características y similitudes.

3.2.1. CodinGame

CodinGame es una empresa que administra la plataforma que lleva el mismo nombre. Puede visitarse a través de www.codingame.com. A través de dicha plataforma, CodinGame ofrece a los usuarios multitud de juegos que deben resolverse programando, con el objetivo de que puedan aprender a programar mejor [21].

Su modelo de negocio esta basado en patrocinios que obtienen de otras empresas tecnológicas interesadas en contactar con desarrolladores de software. A cambio del patrocinio, CodinGame ayuda a estas compañías en labores como la búsqueda de talento mediante concursos a nivel mundial y también privados.

En CodingGame se puede jugar programando hasta en 26 lenguajes de programación diferentes, incluyendo Java, Python, C o Kotlin. Cada jugador elige el lenguaje que prefiera.

CodinGame ofrece dos modos distintos de juego en su plataforma:

- Modo práctica**, en el que el jugador debe resolver un puzzle, empleando para ello técnicas de programación concretas, como pueden ser las estructuras condicionales o las estructuras de control iterativas. El puzzle sigue una línea argumental de fantasía, y ofrece una escena de juego que recrea la situación según el código que escribamos.

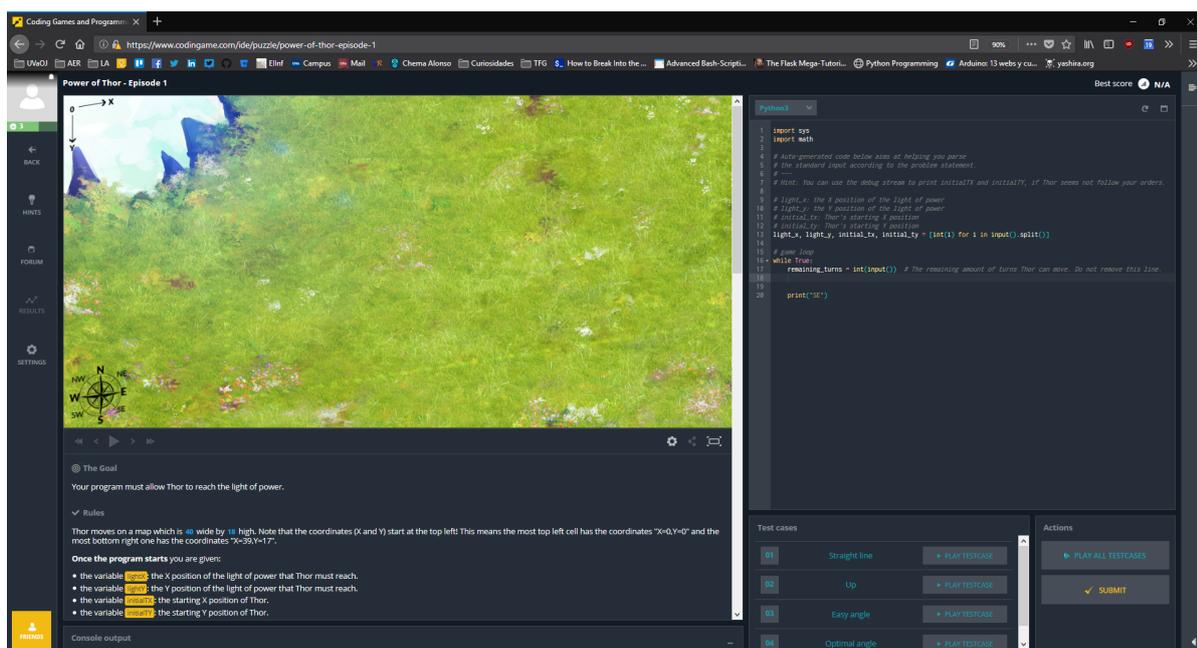


Figura 3.1: Ejemplo de juego en modo práctica en CodinGame.

- **Modo competitivo**, en el que se ofrecen diversos juegos para que los usuarios jueguen entre ellos. El juego más representativo y popular de todos es el *Clash of Code*, en el que varios jugadores se enfrentan, teniendo que resolver un problema con el lenguaje de programación que cada uno escoja. El ganador es aquel que, no sólo lo haga en menos tiempo, sino que emplee la menor cantidad de código para ello.

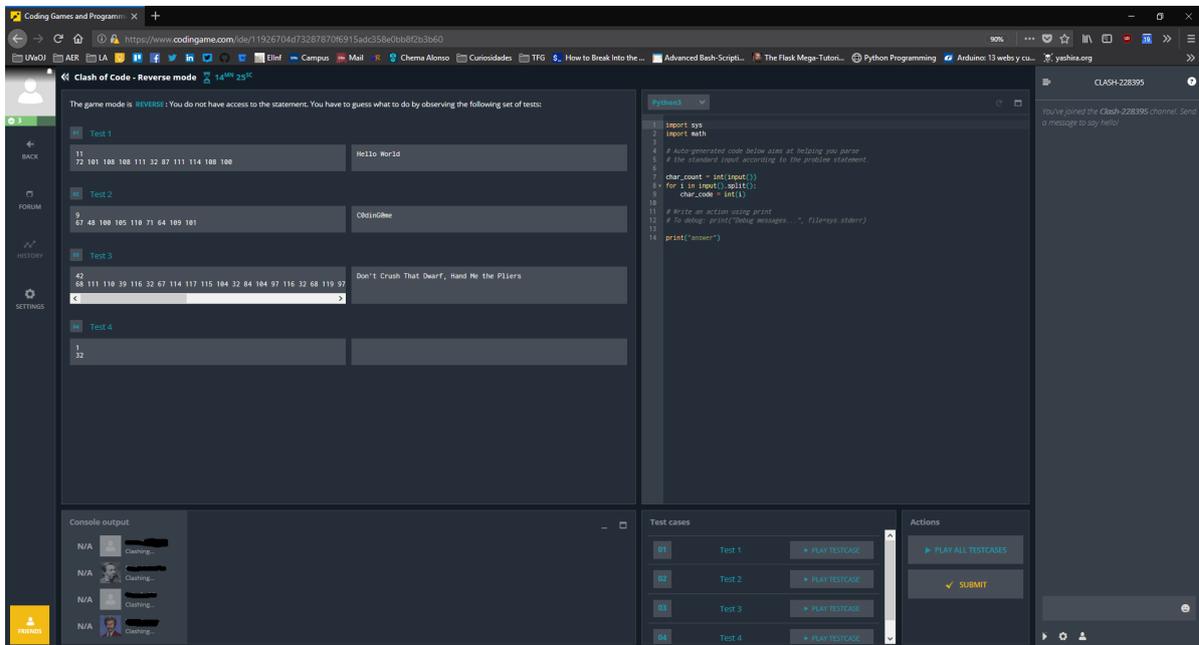


Figura 3.2: Ejemplo de juego *Clash of Code* en CodinGame.

CodinGame ofrece también la posibilidad de poder enviar problemas y desafíos creados por los usuarios. Sin embargo, no existe ningún tipo de juego en la plataforma en el que los usuarios tengan que cooperar para lograr ganar.

3.2.2. CodeCombat

CodeCombat es un juego de rol educacional basado en navegador web cuyo objetivo es enseñar a emplear los lenguajes Python y JavaScript aunque, de forma experimental, van introduciendo nuevos lenguajes [22].

Su modelo de negocio se basa en que los usuarios pagan por obtener ciertas mejoras en el juego o por suscripción para disfrutar de mejoras durante el periodo de suscripción.

Es un juego individual, en el que el jugador debe ir resolviendo retos de dificultad creciente, empezando por un nivel muy básico, que es la sintaxis del lenguaje. Los desafíos van acompañados de una narrativa y una escena de fantasía, en la que deben manejar héroes por el mapa recogiendo gemas y derrotando a oponentes virtuales.

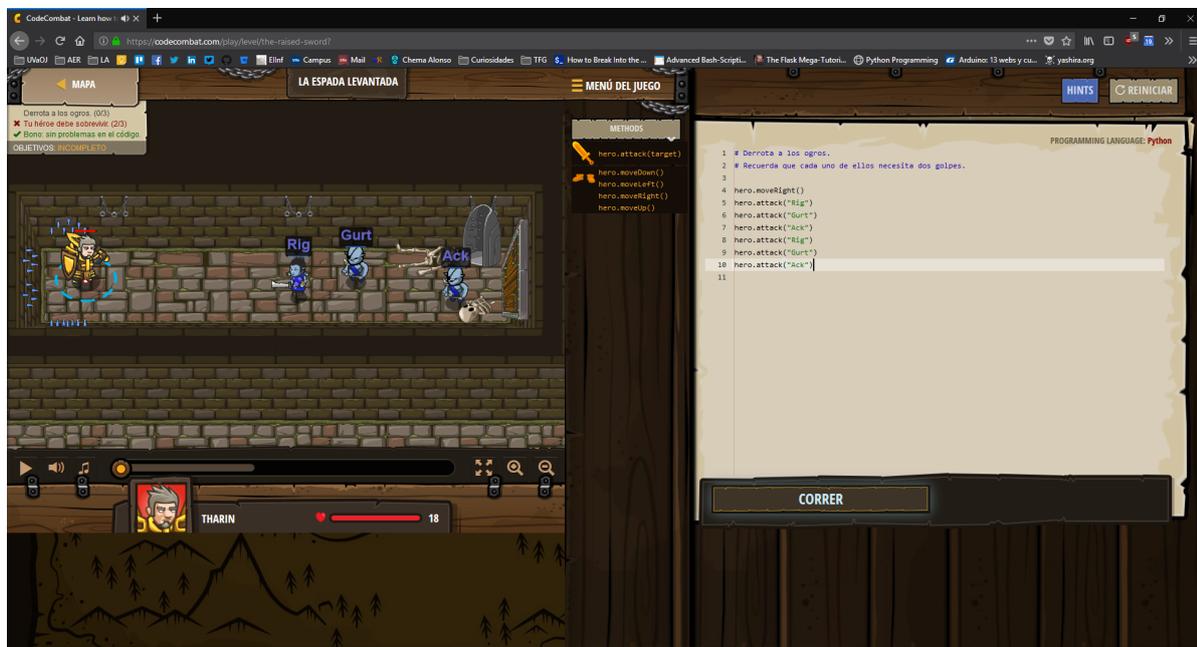


Figura 3.3: Interfaz de juego de CodeCombat.

Esta plataforma está dirigida a un público más infantil y a profesores, ya que ofrece funcionalidades de seguimiento del alumno, y de creación de algunos juegos. Pero tampoco ofrece posibilidad de jugar de forma cooperativa ni competitiva.

3.2.3. Code Hunt

Code Hunt es una plataforma de juegos serios para aprender a programar y participar en concursos de programación desarrollada por Microsoft [23].

Cuenta con una temática de ciencia ficción, en cuanto a lo que a estética se refiere, y soporta Java y C# como lenguajes de programación.

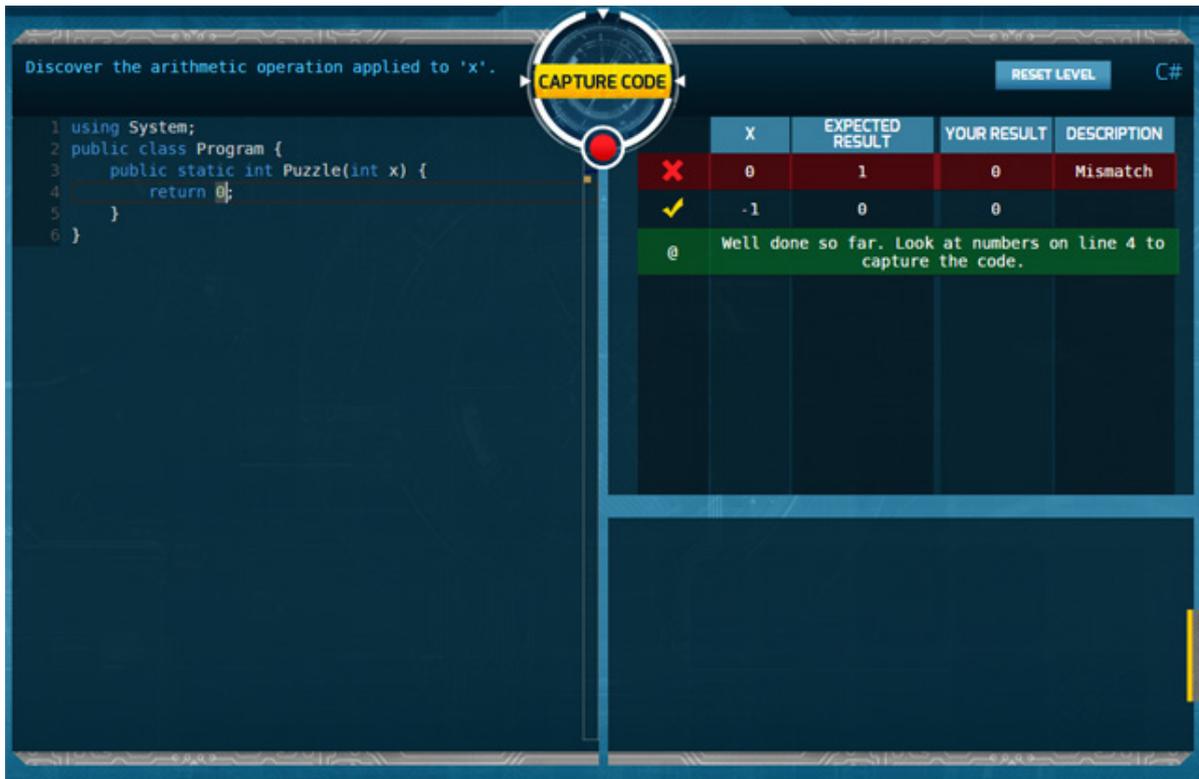


Figura 3.4: Interfaz de juego de Code Hunt. Fuente: [24].

Únicamente soporta partidas uno contra uno o prácticas jugando uno solo, así que no hay modo cooperativo.

Capítulo 4

Análisis de tecnologías

Hemos visto en el capítulo anterior que no existen videojuegos serios relacionados con la programación que tengan los componentes cooperación y competición de forma simultánea. Por lo tanto, no existe nada parecido a la yincana virtual de programación que hemos descrito, así que deberemos crear nosotros mismos el juego que la implemente.

Como se ha indicado en el apartado 2.2.1 de este trabajo, la yincana virtual debe contar con las siguientes características:

- Mundo virtual acorde a la narrativa.
- Comunicación entre integrantes de un equipo.
- Desarrollo cooperativo.
- Comunicación profesor-estudiante.
- Gráfico social.

En este capítulo se van a repasar las distintas tecnologías que permiten crear herramientas que cuentan con cada una de estas características, con el objetivo de estudiarlas, compararlas y decidir cuál es la más adecuada, justificado de manera argumentada la elección de dicha tecnología.

Podemos clasificar las distintas tecnologías a estudiar en cuatro grandes bloques: motores de videojuegos, plataformas para el desarrollo de software colaborativo, servicios de comunicación en los videojuegos.

4.1. Motores de videojuego

Un motor de videojuego es un conjunto de herramientas para diseñar, crear y desarrollar videojuegos. Proveen al juego de una serie de funcionalidades frecuentes en los videojuegos, permitiendo a los creadores y desarrolladores centrarse en los aspectos particulares del juego.

Algunos elementos destacables que proporcionan los motores de videojuegos son motores de renderizado, controles de físicas y colisiones, gestión del sonido, animaciones, gestión de red y conexiones o inteligencia artificial.

En nuestro caso, el motor de videojuego nos permite implementar cuatro de las cinco características que componen la yincana virtual de programación:

- Mundo virtual acorde a la narrativa
- Comunicación entre integrantes de un equipo
- Comunicación entre profesor-estudiante
- Gráfico social

Por ello, el motor de videojuego es, probablemente, el elemento más importante que compondrá la solución que implemente la yincana virtual de programación. A continuación, vamos a revisar algunos de los motores de videojuego más populares y que pueden adecuarse a nuestra solución.

4.1.1. Unity

Unity es un motor multiplataforma creado por la empresa Unity Technologies, y lanzado en 2005. La última versión estable de este motor es la 2018.1 [25].

Con Unity, se pueden crear juegos para multitud de plataformas. Para web: WebGL. Para PC: Windows, SteamOS, OS X, GNU/Linux. Para dispositivos móviles: iOS, Android, Windows Phone. Para Smart TV: tvOS, Samsung Smart TV, Android TV. Para consolas: PlayStation 4 y Vita, Xbox One y 360, Wii U, Nintendo Switch y 3DS. Para dispositivos de realidad virtual: Oculus Rift, Google Cardboard, HTC Vive, Microsoft HoloLens.

Actualmente, Unity se puede obtener mediante tres tipos distintos de licencia:

- Unity Personal: licencia gratuita y privativa, que se puede obtener si el organismo que la emplea no supera los 100.000 USD en ingresos brutos. Esta licencia permite crear videojuegos para todas las plataformas, implementar anuncios y compras dentro de la aplicación. Sin embargo, al crear juegos multijugador, nos limita a que sólo puedan participar hasta 20 usuarios de forma simultánea, si se emplea el servicio Unity Multiplayer. Todos los juegos creados con Unity Personal muestran el mensaje “Made with Unity” al iniciar el juego.
- Unity Plus: esta licencia privativa tiene un coste de 25 USD mensuales, y se puede adquirir siempre que el organismo que la obtiene no supera los 200.000 USD de ingresos brutos anuales. Además de las características de la licencia Personal, ofrece otros beneficios, como poder personalizar la pantalla inicial o descuentos en complementos. Con esta licencia, la limitación para juegos multijugador se amplía hasta 50 jugadores, empleando Unity Multiplayer.

- **Unity Pro:** licencia más enfocada a profesionales, que se puede adquirir por 125 USD al mes, sin límite de ingresos. Ofrece un soporte mejorado y acceso al código fuente, aunque no se tiene información acerca de si permite realizar modificaciones. Respecto a la limitación en juegos multijugador usando Unity Multiplayer, se amplía hasta los 200 jugadores simultáneos.

Unity cuenta con una importante comunidad de desarrolladores que pueden compartir experiencias y problemas a través del foro que Unity pone a su disposición. La documentación oficial se puede encontrar públicamente en la dirección `docs.unity3d.com` y cuenta, además, con multitud de videotutoriales y recursos audiovisuales.

Con Unity se han desarrollado algunos de los juegos más populares en la actualidad, que avalan la versatilidad y potencia de este motor. Por ejemplo, *Hearthstone*, juego de cartas online desarrollado por Blizzard, o el muy conocido *Pokemon GO*, juego de realidad aumentada para dispositivos móviles desarrollado por Niantic.

4.1.2. Blender

Blender es un programa multiplataforma dedicado al modelado, renderizado, y creación de gráficos tridimensionales, que además posee un motor de juegos interno. Fue lanzado por la Fundación Blender en 1995. Actualmente se distribuye bajo licencia GPL [26].

Aunque posee un motor de videojuego, Blender se emplea más frecuentemente con el fin de diseñar y crear modelos en tres dimensiones. Es común usarlo junto con otros motores más potentes como Unity.

4.1.3. Unreal

Unreal Engine es un motor de juego para PC y consolas creado por la empresa Epic Games en 1998. Actualmente, la versión más reciente es la 4.19 [27].

Con Unreal podemos crear juegos para plataformas web, para PC: Windows, macOS, GNU/Linux, SteamOS; para dispositivos móviles: Android, iOS; para consolas: PlayStation 4, Nintendo Switch, Xbox One; y dispositivos de realidad virtual: SteamVR, HTC Vive, Oculus Rift, PlayStation VR o Samsung Gear VR.

Unreal Engine puede adquirirse de forma gratuita y completa, ya que podemos acceder al código fuente en GitHub una vez que hayamos adquirido Unreal Engine. Epic Games no cobra por el uso de Unreal Engine, siempre que no se supere los 3.000 USD brutos de ingreso durante un cuatrimestre por el producto desarrollado con su motor. En caso de superar dicha cantidad, Epic Games obtiene el 5% de los ingresos brutos obtenidos. No obstante se puede negociar los términos de la licencia si se requiere.

Con Unreal Engine también se han desarrollado multitud de juegos exitosos y populares. Reciente-

mente, en el año 2017, se lanzaron al mercado dos videojuegos que crearon sensación: *PlayerUnknown's Battlegrounds* y *Fortnite*.

4.1.4. Source Engine

Es un motor de videojuego lanzado en 2004 y desarrollado por Valve Corporation, la misma empresa que administra Steam, la plataforma digital de distribución de videojuegos más reconocida a nivel mundial [28].

Con Source Engine podemos construir videojuegos para para PC: Windows, osX y GNU/Linux; y para consolas: Xbox One y 360 y PlayStation 3 y 4.

Se puede adquirir este motor de forma gratuita, a través de Steam. Aunque la licencia es privativa.

Existe una larga lista de títulos de videojuegos muy conocidos y desarrollados con este motor, la mayoría distribuidos a través de Steam. El primer juego, que salió a la vez que el propio motor, es *Counter Strike: Source*, pero otros son: *Half-Life*, *Dota 2*, *Portal*, *Counter Strike: Global Offensive* o *Left 4 Dead*.

4.1.5. GameMaker Studio

Game Maker Studio es una herramienta para desarrollar videojuegos orientado a usuarios novatos con pocas nociones de programación. Se publicó en 1999 bajo el nombre de Animo. Actualmente es desarrollado por la empresa YoYo Games [29].

Se pueden crear videojuegos para PC: Windows, Linux, osX; para sistemas móviles: Android, iOS; para consolas: Xbox 360 y One, PlayStation 3, 4, Vita.

Se puede adquirir de manera gratuita, aunque el software es privativo y está sujeto a su propia licencia. También existen opciones comerciales con características avanzadas.

4.1.6. CryEngine

CryEngine es un motor de videojuego creado por Crytek, en 2002. Originalmente era un motor de demostraciones para NVidia. Su última versión estable es CryEngine V y fue lanzada en 2016 [30].

Con este motor se puede desarrollar para ordenadores Windows o Linux, y para consolas PlayStation 4 y Xbox One.

Sin embargo, su licencia restringe expresamente el empleo de CryEngine para juegos serios, de modo que no podemos emplearlo para nuestro propósito.

CryEngine es conocido por ser el motor del videojuego *FarCry*, lanzado en 2004.

4.1.7. LibGDX

LibGDX es un framework para el desarrollo de videojuegos multiplataforma. Permite crear videojuegos para Windows, Linux, macOS, Android, iOS y web. Es una biblioteca bastante simple, que se puede emplear para juegos en dos dimensiones sin demasiadas físicas. Se distribuye mediante licencia Apache2 [31].

4.2. Plataformas para el desarrollo de software colaborativo

Existe un dogma que la mayoría de los que se dedican a enseñar a programar repiten: “para aprender a programar, hay que programar”. Así que, dentro del juego serio que se pretende construir, hay un punto en que los jugadores tienen que programar.

Dado que el juego es cooperativo, por equipos, los alumnos deberán programar juntos, de tal forma que todos puedan contribuir a ello. Es por ello que una de las características clave descritas en el apartado 2.2.1 de este trabajo es el desarrollo cooperativo. Eso sí, se asume que los estudiantes de estos cursos son principiantes en la materia. Por lo que es conveniente que el entorno de desarrollo sea sencillo, es decir, no es necesario que tenga opciones de autocompletado u otras ayudas que proporcionan IDEs como Eclipse, Netbeans o Idea.

Introducir un editor online dentro del propio videojuego, involucrando al motor de videojuego, es complicado, al menos para las primeras versiones. Por ello, vamos a analizar a continuación las herramientas de desarrollo colaborativo que pueden ser más útiles y convenientes. Todos estos editores online funcionan a través del navegador web.

4.2.1. Collabedit

Collabedit es un editor online simple en el que varios programadores pueden colaborar sobre el mismo fichero. Para ello, basta con compartir la URL que la propia plataforma te proporciona [32].

No requiere ningún tipo de registro en la plataforma, por lo que no se tiene ningún control sobre quien accede al fichero compartido, aunque sí permite ver los seudónimos de las personas que colaboran en el momento.

En Collabedit se puede programar hasta en 17 lenguajes diferentes, entre los que podemos encontrar Java, Python, C, Ruby o Perl. Sin embargo, es tan solo un editor. No permite compilar ni ejecutar el código que se escribe en él, por lo que se requiere que uno de los colaboradores lo copie a un fichero local para que sea compilado y ejecutado. Tampoco ofrece ayudas al programador, simplemente colorea el código en función del lenguaje elegido

Ofrece una interfaz simple y limpia en la que, además del propio editor, podemos encontrar un chat en el que hablar con el resto de colaboradores. A medida que otro colaborador edita el fichero, lo que va escribiendo aparece en las pantallas de los demás de una forma relativamente fluida.

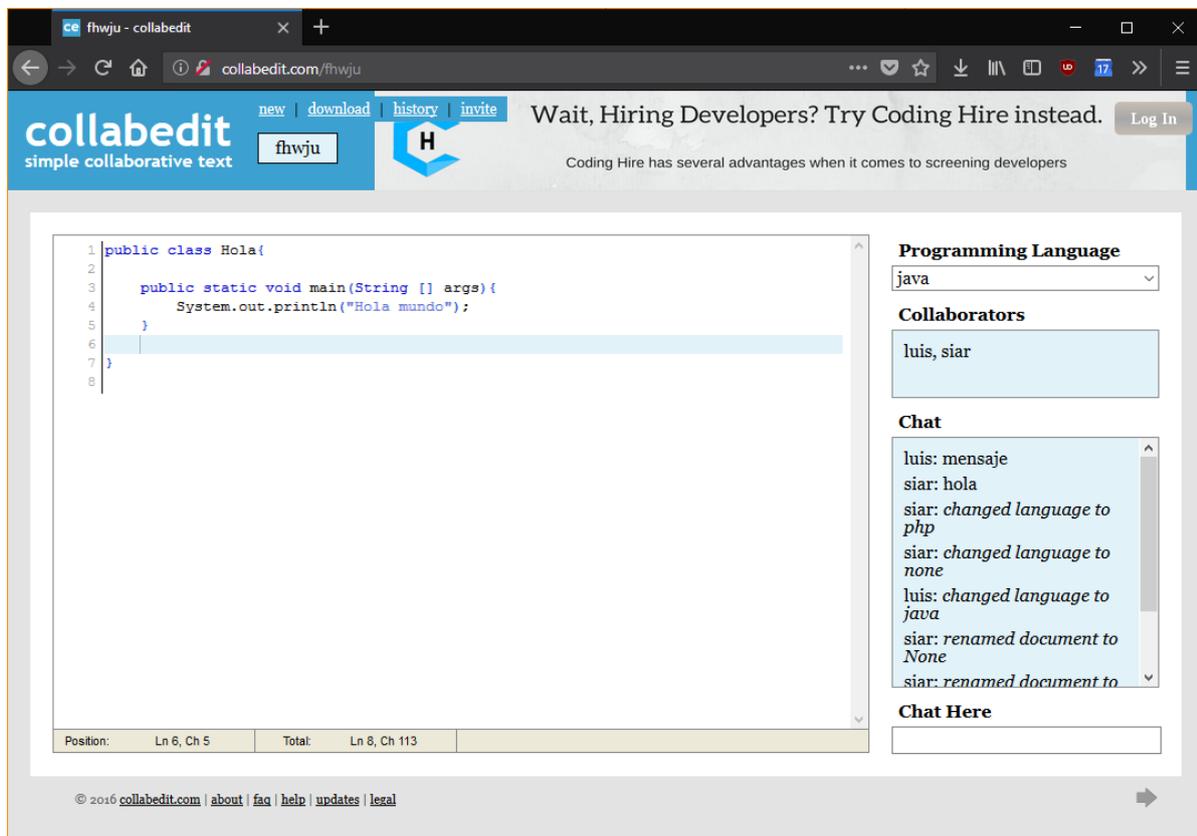


Figura 4.1: Interfaz de Collabedit.

4.2.2. CodeShare

CodeShare es una herramienta similar a Collabedit, que además guarda ciertos parecidos. Para compartir, también basta como compartir la URL del navegador [33].

Tampoco requiere de ningún tipo de registro en la plataforma, por lo tanto, no tenemos control sobre quien accede al fichero compartido. Además, tampoco nos muestra ningún tipo de información sobre los colaboradores.

CodeShare nos ofrece la posibilidad de programar en más de 90 lenguajes diferentes, resaltando el texto en función del que elijamos. Pero tampoco permite compilar o ejecutar, ni ofrece ninguna ayuda.

En lugar de ofrecernos un chat, nos permite realizar videollamadas dentro de la propia interfaz, lo cual resulta muy útil.

El editor se acomoda al modo de trabajo del usuario, ya que permite elegir trabajar como en el editor Vim, Emacs o Sublime. Además, funciona de forma fluida, ya que se puede ver lo que escriben los colaboradores de manera instantánea carácter a carácter.

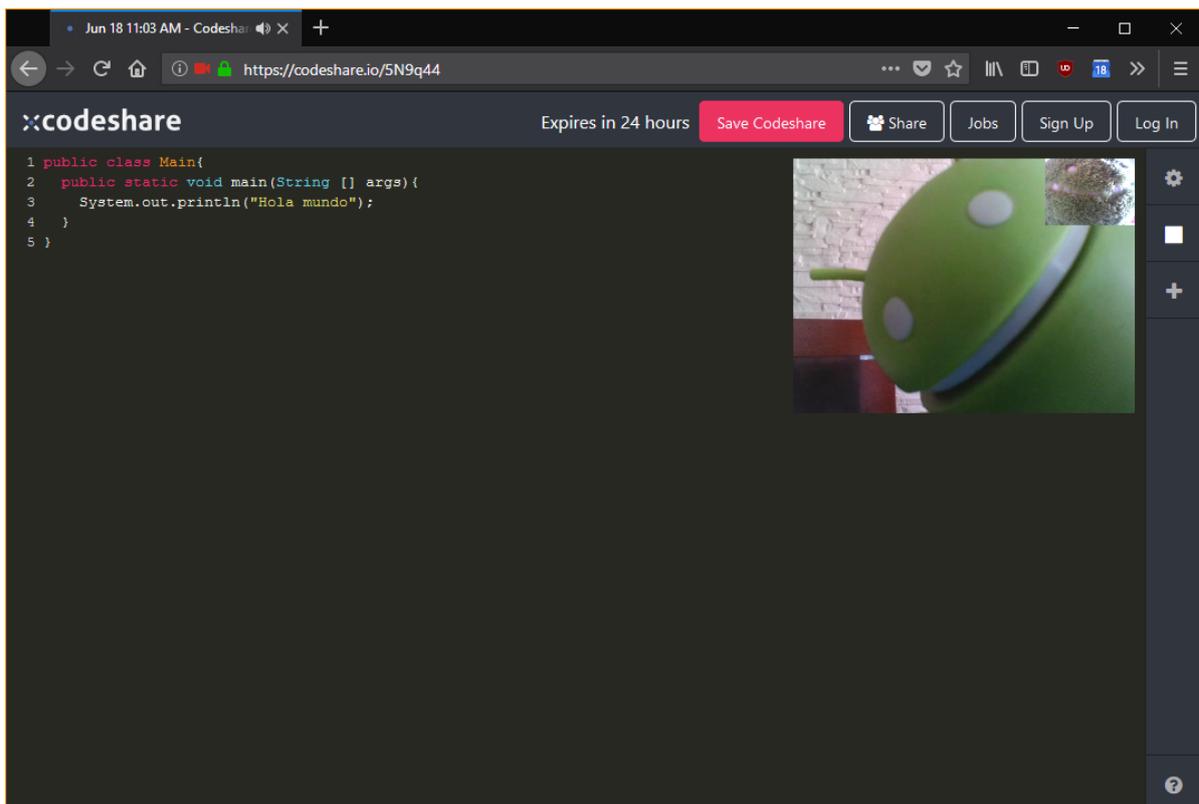


Figura 4.2: Interfaz de CodeShare.

4.2.3. JSFIDDLE

JSFIDDLE es un editor online colaborativo con la restricción de estar orientado a programación frontend. Por esta restricción, permite codificar en HTML, CSS y JavaScript [34].

Tampoco requiere de registro en la plataforma, y para colaborar también hay que compartir una URL. Nos permite ver la lista de colaboradores, dónde está escribiendo cada uno, la ubicación de su cursor en la ventana y dónde hace clic.

Nos ofrece la posibilidad de comunicarnos con el resto de colaboradores a través de un chat y de llamada de voz.

A priori, el hecho de que solo se pueda emplear JavaScript como lenguaje de programación puede ser una limitación. Sin embargo, este editor permite ejecutar el código, y ofrece una serie de ayudas al programador. Su comportamiento es muy fluido al compartir.

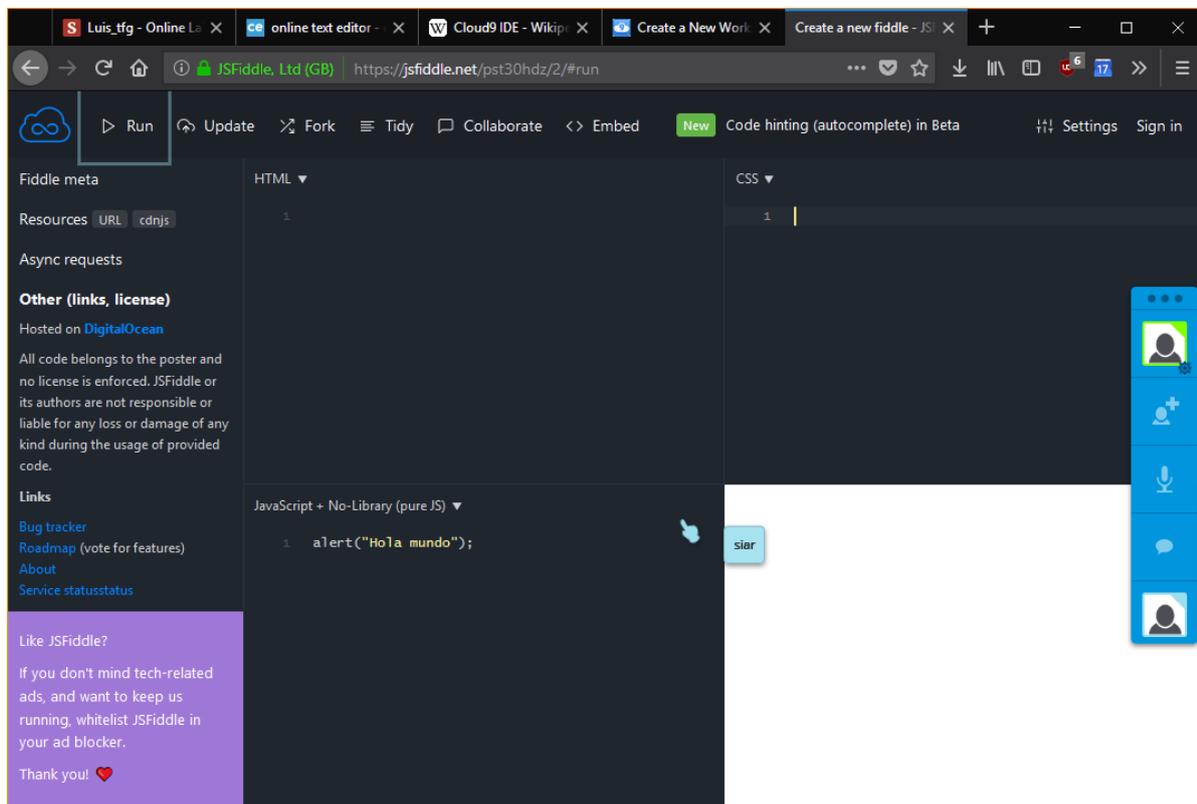


Figura 4.3: Interfaz de JSFIDDLE.

4.2.4. SCRAPfy

SCRAPfy es una herramienta para editar código de forma colaborativa, compartiendo simplemente la URL, sin requerir ningún tipo de registro. Está desarrollado por la compañía Hashdog [35].

Permite programar en 20 lenguajes distintos, coloreando y resaltando el código en función del lenguaje que se escoja. Además, ofrece ayudas durante la programación, como mostrar sugerencias basadas en predicción de lo que se va a escribir y autocompletado. Aunque tampoco permite compilar ni ejecutar.

También nos ofrece la posibilidad de comunicarnos con el resto de colaboradores a través de chat y a través de llamada de voz. Su funcionamiento es fluido, pudiendo ver de una forma prácticamente instantánea lo que escriben los demás colaboradores letra a letra

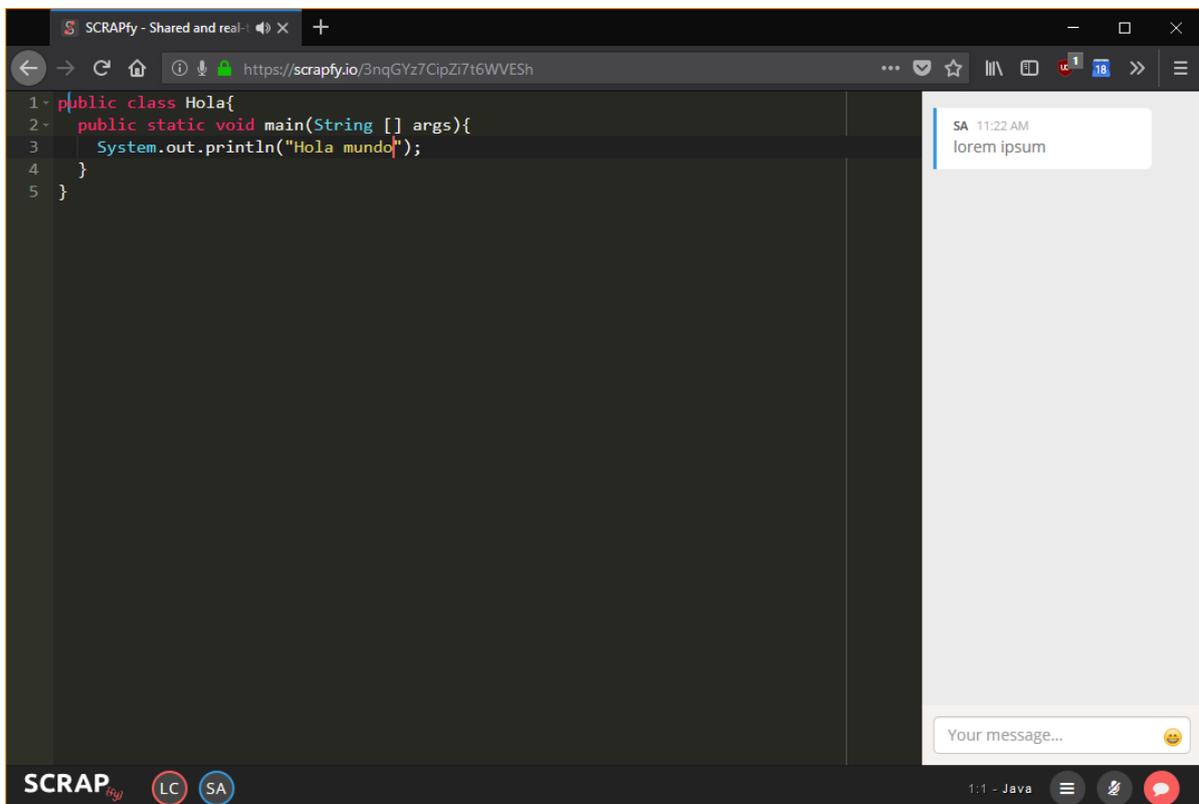


Figura 4.4: Interfaz de SCRAPfy.

4.2.5. Cloud9

Cloud9 es un IDE online que permite trabajar de forma colaborativa. Actualmente es desarrollado por Amazon Web Services y conforma una de sus potentes herramientas [36].

Los usuarios deben registrarse para poder usar la plataforma. Además, se emplea gestión de permisos para compartir los repositorios. Por ello, se tiene un control absoluto de quien accede y modifica un archivo.

Como se ha dicho, se trata de un IDE, por lo que posee tanto ayudas al programador, como la posibilidad de compilar y ejecutar el código. Soporta cientos de lenguajes como C, C++, Java, PHP, Python o JavaScript.

Además, nos ofrece un chat con el que hablar con el resto de programadores mientras se trabaja.

El funcionamiento de esta plataforma es muy adecuado. Sin embargo, se trata de una herramienta con mucha potencia. Tanta, que quizá sea demasiada para el propósito para el que se quiere usar, ya que los alumnos pueden no llegar a entender la cantidad de opciones, pudiendo llegar a sentirse frustrados o abrumados por la herramienta.

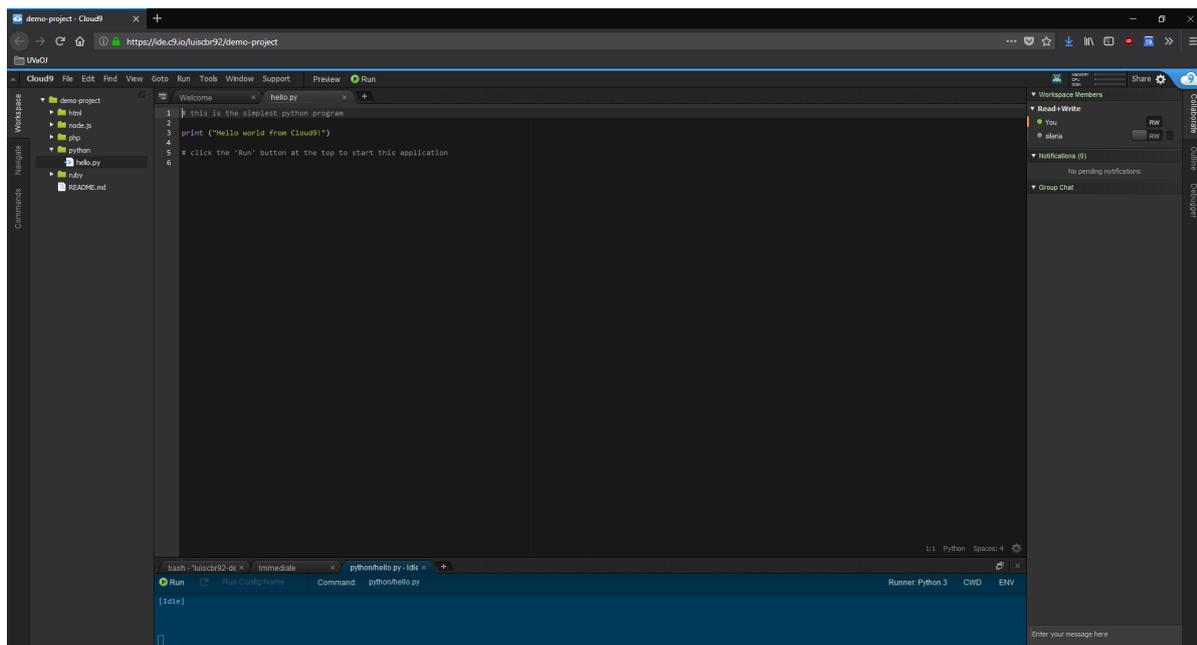


Figura 4.5: Interfaz de Cloud9.

4.3. Comunicación en el videojuego

La comunicación, cuando se trata de videojuegos multijugador, es algo vital. Los jugadores necesitan intercambiar impresiones y estrategias para conseguir la victoria.

En nuestro caso, queremos que la yincana tenga dos características relacionadas con la comunicación: permitir la comunicación entre integrantes de un equipo, y también entre el profesor y los estudiantes

4.3.1. Servicios de chat en videojuegos

A la hora de permitir una comunicación en la yincana de programación, una de las soluciones es emplear un servicio de mensajería o chat en el videojuego que se construya.

Para crear y dar servicio de chat, se debe emplear el motor de videojuego. Muchos motores incluyen dentro de sus funcionalidades la posibilidad de implementar estos servicios de mensajería. En el caso de Unity, podemos encontrar assets que lo implementan. Pero la opción más sencilla es emplear el servicio Unity Multiplayer. Conviene recordar que Unity Multiplayer puede ser usado con un límite de jugadores participando a la vez, que varía en función del tipo de licencia que se adquiera.

Esta opción puede ser más útil para cubrir la comunicación entre el profesor y el equipo de estudiantes.

4.3.2. Servicios de VoIP en videojuegos

Otra de las opciones para cubrir las necesidades de comunicación, es implementar un servicio de VoIP en el videojuego.

De nuevo, esta responsabilidad recaería en el motor de videojuego. Para Unity, podemos encontrar assets, como Dissonance Voice Chat, que nos permiten emplear esta funcionalidad sin que tengamos que hacerlo nosotros, lo cuál sería un esfuerzo importante. Eso sí, estos assets no son gratuitos.

4.3.3. Servicios VoIP externalizados

Como ya hemos visto en el epígrafe 3.1.2, sobre cómo se comunican los jugadores de videojuegos multijugador, la mayoría de ellos opta por emplear servicios externos a los propios videojuegos.

Vamos a repasar las plataformas externas más usadas para este propósito

TeamSpeak

TeamSpeak es una herramienta de VoIP desarrollada por TeamSpeak Systems GmbH y lanzada en 2002 [37].

Con TeamSpeak se puede crear un servidor en el que se pueden añadir usuarios para que se conecten a través del programa cliente. Su licencia gratuita nos permite crear hasta 32 usuarios en un mismo servidor.

Discord

Discord es una aplicación de VoIP gratuita y diseñada para comunidades de videojuegos. Fue lanzada en 2015, y en noviembre de 2017 contaba con más de 87 millones de usuarios [38].

Permite crear servidores, es decir, comunidades de jugadores en el que, a su vez, pueden crear distintas salas de chat de voz. También dispone de un chat general, por lo que el profesor puede estar en este chat resolviendo dudas generales.

Nos permite resolver las necesidades de comunicación de la yincana en una sola herramienta.

Skype

Skype es, también, un servicio de VoIP desarrollado actualmente por Microsoft. Es posiblemente el servicio más popular en el ámbito de los chats de voz [39].

Permite también compartir vídeo, y tiene también características interesantes como la opción de compartir pantalla. Esta funcionalidad nos puede permitir retransmitir la ventana del videojuego de tal manera que el resto de integrantes del equipo puedan ver el juego.

4.4. Conclusiones sobre el análisis de tecnologías

Una vez vistas las tecnologías más importantes para el desarrollo de la yincana virtual de programación, en este epígrafe vamos mostrar una serie de ideas y conclusiones al respecto. Gracias a ellas, podremos tener una aproximación sobre cómo debe diseñarse la solución que compondrá la yincana virtual.

4.4.1. Sobre los motores de videojuego

Como ya se ha mencionado, el motor de videojuego es el elemento más importante que va a formar parte de la yincana virtual. Por ello, elegir el más adecuado es crucial.

Existen un sinnúmero de motores de juegos. En este capítulo hemos estudiado los más importantes del panorama actual. En la siguiente tabla podemos ver sus características de forma muy resumida. De esta tabla se ha excluido el motor CryEngine, debido a que, según su licencia, no se permite su explotación para crear juegos serios, lo que es nuestro objetivo.

| | Unity | Blender | Unreal | Source | GameMaker | libGDX |
|---|-------------------------|----------------|---------------------|---------------|------------------|---------------|
| 2D / 3D | 2D y 3D | 2D y 3D | 3D | 2D y 3D | 2D | 2D y 3D |
| Motor físico apropiado | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Para PC | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Importante comunidad de desarrolladores | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Se han construido importantes juegos | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Lenguajes para scripting | C# JavaScript Boo | Python | C++ UnrealScript | VScript | GML | Java |
| Licencia | Privativa | GPL | Privativa | Privativa | Privativa | Apache2 |
| Coste | Gratuito* | Gratuito | Gratuito* | Gratuito* | Gratuito* | Gratuito |

Tabla 4.1: Comparativa de motores de videojuego

También es importante escoger una plataforma en la que exista una buena cantidad de desarrolladores, ya que es un indicativo de que podremos encontrar buena documentación disponible y será útil cuando necesitemos información durante los problemas que puedan surgir durante el desarrollo. Y si un motor cuenta con una buena cantidad de videojuegos creados con él, que además sean conocidos y exitosos, estos avalan su potencia y utilidad.

Todos los motores que hemos visto se pueden emplear de forma gratuita, pero algunos de ellos han sido marcados con asteriscos. Esto es porque su uso gratuito está sujeto a una serie de condiciones, en las que será necesario pagar cierta cantidad. Unreal, por ejemplo, especifica en su licencia un sistema de regalías, es decir, pago en función de ingresos generados por la explotación del motor. O GameMaker, que cuenta con diversos planes de pago en función de una serie de características opcionales. En el caso de Unity, también cuenta con varios planes de pago que se ven afectados según los ingresos generados, aunque también ofrece características mejoradas en los planes más caros. Dado que nuestro objetivo no es obtener ingresos, no debería preocuparnos este aspecto por ahora.

Tanto si queremos que nuestro juego simule una realidad en dos o en tres dimensiones, es importante que el motor físico sea apropiado y completo. Esto quiere decir, que el motor debe contener una cantidad de funciones que cubran una buena cantidad de necesidades que tendremos a la hora de crear nuestro videojuego. En este sentido, Blender es un programa cuya principal utilidad es el diseño y modelado en tres dimensiones. Incorpora un pequeño motor que puede servir para crear secuencias y animaciones, pero que no contiene otras funcionalidades importantes, como puede ser la gestión multijugador.

Sin embargo, el uso de Blender junto a otros motores más completos es común. Unity permite

4.4. Conclusiones sobre el análisis de tecnologías

importar objetos creados con Blender, por ejemplo. Por lo que no debe descartarse su uso, sobretodo si se quiere dedicar bastante trabajo a crear modelos tridimensionales.

Con todos estos datos, se llega a la conclusión de que el mejor motor para implementar la yincana virtual de programación es Unity, ya que puede usarse de forma gratuita, cuenta con una buena comunidad de desarrolladores y multitud de juegos conocidos que le avalan como un buen motor de videojuego.

4.4.2. Sobre las plataformas de desarrollo de software colaborativo

La herramienta de desarrollo colaborativo es el segundo componente más importante para la yincana virtual de programación. Ya hemos visto las plataformas disponibles más importantes que nos ofrecen crear código de manera colaborativa. En la siguiente tabla se resumen de forma breve.

| | Collabedit | CodeShare | JSFIDDLE | SCRAPfy | Cloud9 |
|---------------------------|------------|-----------|------------|---------|-----------|
| Uso sin registro | ✓ | ✓ | ✓ | ✓ | ✗ |
| Permite ver colaboradores | ✓ | ✗ | ✓ | ✓ | ✓ |
| Lenguajes disponibles | 17 | Más de 90 | JavaScript | 20 | Más de 40 |
| Resaltado de código | ✓ | ✓ | ✓ | ✓ | ✓ |
| Autocompletado | ✗ | ✗ | ✓ | ✓ | ✓ |
| Compila y/o ejecuta | ✗ | ✗ | ✓ | ✗ | ✓ |
| Chat | ✓ | ✗ | ✓ | ✓ | ✓ |
| Llamada de voz | ✗ | ✓ | ✓ | ✓ | ✗ |
| Llamada de vídeo | ✗ | ✓ | ✗ | ✗ | ✗ |

Tabla 4.2: Comparativa de plataformas de desarrollo colaborativo

Como ya hemos dicho anteriormente, no se requiere emplear una herramienta compleja que ofrezca ayudas al programador, por lo que una interfaz sencilla nos sirve para nuestra solución. Tampoco es necesario que la plataforma compile y ejecute el código, porque el jugador puede compilar en su propia máquina tras haber copiado el código a un fichero local.

En este sentido, las herramientas más útiles son Collabedit y CodeShare, ya que cumplen los requisitos que hemos mencionado anteriormente. CodeShare nos ofrece un plus al poder hablar con el resto de colaboradores a través de servicios de voz y vídeo.

Hemos visto además que todos estos editores nos ofrecen algún modo de comunicarnos con el resto de colaboradores, por lo que podríamos decir que el empleo de alguna de estas herramientas nos resuelve dos de las cinco características de la yincana virtual:

- Desarrollo cooperativo
- Comunicación entre integrantes del un equipo

No obstante, se debe tener en cuenta que, al emplear una herramienta de terceros, quien dirige la yincana no tiene control sobre el código empleado para solucionar las distintas pruebas. Ni siquiera

se puede imponer que usen estas u otras herramientas. Por lo tanto, la elección de una de estas plataformas va a ser para ofrecer una sugerencia de entorno de trabajo común. Incluso, puede que algunos se decanten por emplear una de estas herramientas, y otros, otra.

4.4.3. Sobre la comunicación necesaria para la yincana

La comunicación en la yincana es imprescindible. De hecho, dos de las cinco características que hemos enunciado al principio de este capítulo tienen que ver con ello.

Hemos visto que muchos videojuegos multijugador del mercado incorporan algún servicio, ya sea chat o mediante VoIP. Sin embargo, los jugadores prefieren emplear servicios de terceros, por razones que ya hemos explicado en el apartado 3.1.2 de este trabajo.

Para la comunicación entre integrantes de un mismo equipo, lo más práctico es emplear un servicio de VoIP. Hemos visto que muchos editores de código online, como CodeShare, integran servicios así, por lo que puede ser una posible solución. Pero sigue siendo más práctico emplear un servicio externo.

En este trabajo hemos hablado de tres servicios que ofrecen VoIP: TeamSpeak, Discord y Skype. Cada una de estas herramientas tiene sus propias ventajas. Sin embargo, tenemos una situación similar a la que tenemos con las plataformas de desarrollo colaborativo: no tenemos control sobre qué herramienta emplea cada equipo. Por lo que puede dejarse a elección de los propios alumnos emplear una de estas herramientas u otra distinta con la que se sientan cómodos.

Para la comunicación entre el profesor y el equipo, la solución elegida depende del propósito de cada mensaje que se quiera enviar:

- Si el profesor quiere enviar un mensaje a todos sus alumnos, por ejemplo, para dar una pista cuando todos los equipos están atascados en determinada prueba, puede ser buena idea elaborar un sistema de notificaciones dentro del propio juego, lo cual emplearía el motor de videojuegos para su implementación.
- Si la comunicación es más individual, por ejemplo, un equipo tiene una duda y quiere preguntársela al profesor, lo mejor es emplear una herramienta externa que garantice la privacidad, como puede ser un correo electrónico.

4.4.4. Resumen de las tecnologías necesarias

Para cerrar este capítulo se muestra esta tabla, que resume las tecnologías elegidas para implementar cada una de las cinco características necesarias para implementar la yincana virtual de programación.

4.4. Conclusiones sobre el análisis de tecnologías

| Característica | Componente necesario | Nombre de la herramienta |
|---|---------------------------------------|---------------------------------|
| Mundo virtual acorde a la narrativa | Motor de videojuego | Unity |
| Desarrollo colaborativo | Plataforma de desarrollo colaborativo | CodeShare |
| Comunicación entre integrantes del equipo | Servicio VoIP externo | Skype |
| Comunicación profesor-estudiante | | |
| Gráfico social | Motor de videojuego | Unity |

Tabla 4.3: Resumen de características y herramientas empleadas para su resolución

Esta tabla se ha construido en base a los hechos y argumentos expuestos a lo largo de todo el capítulo. Sin embargo, no es necesario que se interpreten como un dogma irrefutable. La decisión de adoptar estas tecnologías se ha tomado para solucionar las necesidades actuales de la yincana virtual de programación. Dichas necesidades pueden variar a lo largo del tiempo, por lo que el empleo de ciertas tecnologías en lugar de otras deberá replantearse llegado el caso. A su vez, este texto pretende ser una guía ese momento.

Capítulo 5

Unity. Conceptos básicos

En el capítulo anterior, tras realizar el análisis de las diferentes tecnologías necesarias para la implementación de la yincana virtual, se expone cual es mejor en cada ámbito, aportando las razones oportunas para cada una. En el ámbito de los motores de videojuego, se concluyo que el más adecuado que existe en el mercado para nuestra solución es Unity.

Más adelante, trabajaremos con Unity, pero antes debemos entender bien qué es, qué ofrece y cómo funciona. En este capítulo, vamos a ver todos estos temas de una forma breve, ya que no pretende ser un manual de la herramienta. Pero si que es necesario comprender parte de su funcionamiento desde el punto de vista de la ingeniería del software para poder modelar adecuadamente nuestra solución.

5.1. Qué es Unity

Unity es un motor de videojuego desarrollado por Unity Technologies, cuya función es proporcionar de un nivel de abstracción para la programación de videojuegos. Incluye un conjunto de funciones y rutinas habituales en el software de los juegos, como pueden ser físicas, detección de colisiones, ingeniería artificial o sonidos y aporta un lenguaje específico de dominio para escribir el código de nuestros juegos. Unity está implementado en C, C++ y C#.

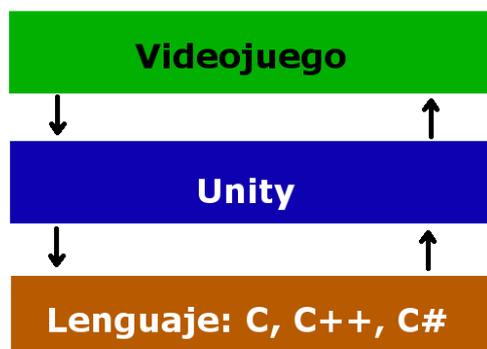


Figura 5.1: Nivel de abstracción que proporciona Unity.

La interfaz del editor de Unity es intuitiva y configurable a gusto del programador. Existen muchos tutoriales, además de la guía oficial, que enseñan a manejar y familiarizarse con el entorno. En adelante nos centraremos en la ingeniería de software que lo compone.

5.2. Estructura de clases

El motor de Unity se identifica por el nombre `UnityEngine`, y en él se implementan todas las clases del motor. La clase básica, de la que hereda el resto, es `Object` [40, 41].

Un `GameObject` es una de las clases más importantes, que además hereda directamente de `Object`. En adelante, nos referiremos a los `GameObject` como *objetos del juego*.

Prácticamente, cualquier elemento que vemos en la pantalla del videojuego es un objeto del juego, como el jugador, el mapa o los enemigos.

Los objetos se componen de instancias de la clase `Component` o de clases que hereden de ella. La clase `Component` también hereda directamente de la clase `Object`, y en adelante nos referiremos a las instancias de esta clase como *componentes*. No se debe confundir con el concepto de componente que se tiene en el ámbito del desarrollo basado en componentes.

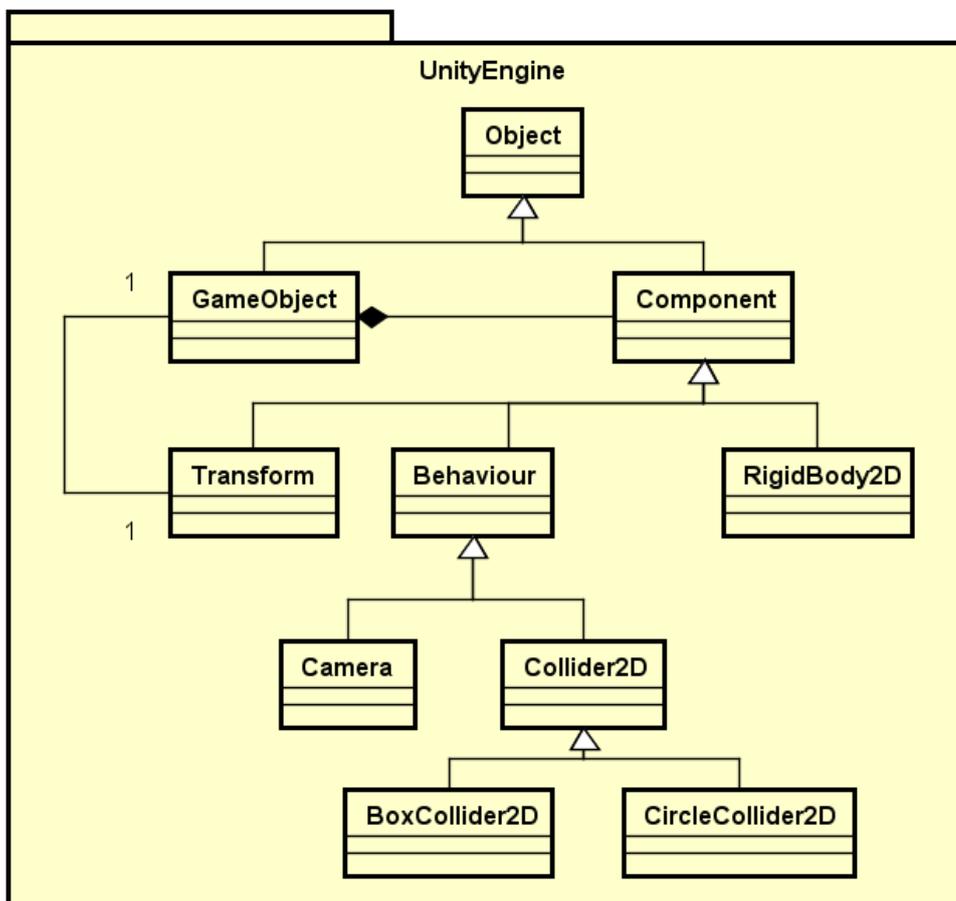


Figura 5.2: Esquema parcial de UnityEngine.

Los componentes definen el comportamiento, el aspecto, o la función del objeto del juego. Por ejemplo, si a un objeto se le añade el componente `Camera`, se le están agregando todas las funciones para que muestren el espacio de juego a través de la pantalla.

Todos los objetos del juego poseen, al menos, un componente `Transform`, que sirve para dar posición, escala y rotación en el espacio tridimensional de la escena. Pero también existen un sinnúmero de componentes, que permiten asignar múltiples funcionalidades necesarias para dar lugar a un videojuego completo.

La figura 5.2 muestra el modelo de `UnityEngine`, con algunas de las clases más importantes de Unity.

5.3. Organización de ficheros en Unity

Debemos entender también como maneja Unity los datos, tanto en memoria como en disco.

Así como los objetos del juego son la clase básica a nivel de memoria, en la parte de persistencia el elemento más importante es el *asset*.

Un *asset* es un fichero en disco, que se guarda en la carpeta “Assets” de un proyecto en Unity. Componen los recursos necesarios, como imágenes, materiales, clips de audio. Los *assets* con formatos propios de Unity son equivalentes a lo que conocemos como archivo JAR en el universo Java.

Un *asset* contiene una colección jerárquica de `Objects`. En Unity, la cache mantiene un mapa entre las instancias de un `Object` en memoria, el archivo *asset* donde se almacena y la ubicación del `Object` dentro del propio *asset*. De esta se permite al motor mantener referencias robustas entre *assets* y `Objects` [42].

Una **escena** de Unity es un tipo de *asset* especial, que contiene todos los `Objects` que la componen. Otro tipo especial de *asset* es un **Prefab**. Los **Prefabs** se emplean cuando necesitamos crear objetos dinámicamente durante el juego. Por ejemplo, cuando en nuestro juego disparamos un proyectil desde una nave espacial, el proyectil se ha instanciado dinámicamente desde el disco. Y, además, cuando el proyectil salga de nuestra pantalla de juego, destruiremos su instancia para evitar que el proyectil se mueva indefinidamente por el espacio tridimensional, optimizando así el juego.

Capítulo 6

Proyecto: Yincana virtual de programación

Tras haber revisado las herramientas existentes para aprender a programar jugando y haber analizado las tecnologías permiten implementar un juego serio como la yincana virtual, en este capítulo se pone en marcha el proyecto para crear dicho juego serio.

Tal y como se ha mencionado, el objetivo de este trabajo no trata de elaborar un producto final que implemente todas las características de la yincana virtual de programación. Sin embargo, se pretende elaborar un prototipo que implemente una de esas características.

El prototipo constará de un videojuego serio de un sólo jugador. El objetivo del juego será que el jugador aprenda a programar.

Como ya se mencionó en los objetivos de este Trabajo, la normativa de TFG indica que la duración debe ser de 300 horas. Este proyecto no compone la totalidad temporal del Trabajo de Fin de Grado, y su duración será de 100 horas.

6.1. Metodología

Para la realización de este proyecto, se ha optado por una metodología de desarrollo ágil. Se ha empleado la metodología Scrum [43], con ligeras modificaciones que permiten la adaptación del trabajo a las características del Trabajo de Fin de Grado.

La descripción de requisitos, en Scrum, se hace a través de lo que se conoce como historias de usuario. Las historias de usuario deben ser:

- Independientes unas de otras.
- Negociables.

- Valoradas por clientes o usuarios.
- Estimables.
- Pequeñas.
- Verificables.

La colección de todas las historias de usuario del proyecto, formando una lista priorizada, componen lo que Scrum denomina Product Backlog.

En Scrum, el proyecto se divide en bloques de trabajo llamados Sprints, típicamente de una duración fija. Sin embargo, para adaptarse al contexto temporal del proyecto y a sus necesidades, emplearemos sprints con duraciones de una y dos semanas.

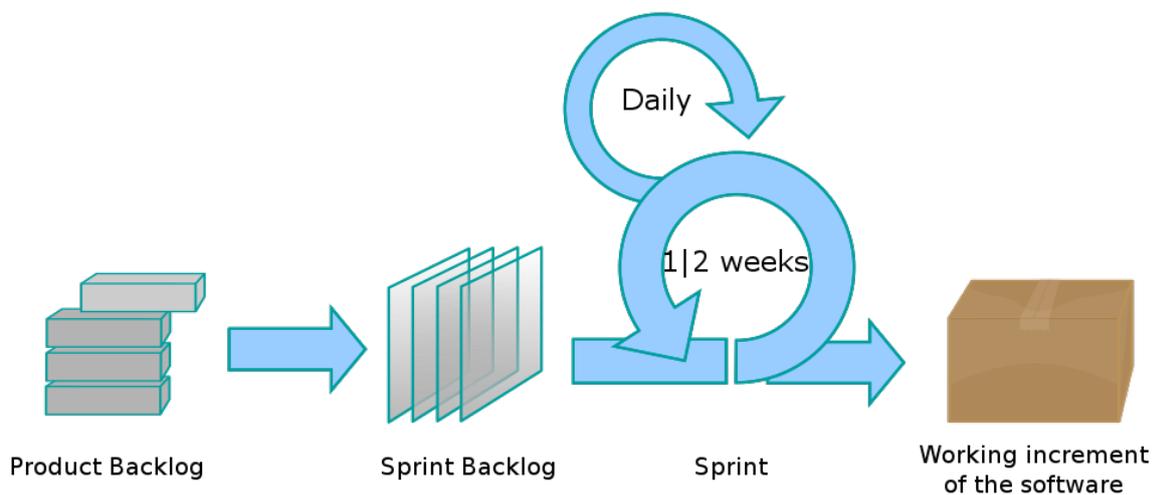


Figura 6.1: Esquema de trabajo de Scrum.

Al principio de cada sprint, se genera el Burn Down Chart, un gráfico que muestra la cantidad de historias pendientes de incorporación al producto. También se elabora el Sprint Backlog, que es un subconjunto de historias que se van a implementar a lo largo del sprint que va a dar comienzo.

Existen una serie de roles en Scrum:

- **Equipo de desarrollo:** grupo de personas que trabajan en la creación del producto. En este caso, el equipo está conformado únicamente por Luis Alberto Centeno Bragado.
- **Product Owner:** persona dedicada a establecer un puente entre el usuario final de la herramienta y el equipo de desarrollo. En este caso, será la tutora Alma Pisabarro.
- **Scrum Master:** persona que apoya y da soporte al equipo de desarrollo, resolviendo conflictos organizativos. Este papel lo desempeña el tutor Carlos Enrique Vivaracho.

Además, al inicio de cada día, el equipo de desarrollo suele reunirse, en la Daily Scrum, para intercambiar impresiones, mostrar dificultades y ofrecer ayuda entre ellos. Como en este caso, el equipo de desarrollo está formado por una única persona, esta reunión no se realiza.

Emplear Scrum aporta una serie de ventajas, entre las cuales podemos encontrar:

- Flexibilidad ante los cambios.
- Mayor calidad del software.
- Mayor productividad.
- Predicción de tiempos.
- Reducción de riesgos.

6.2. Planificación inicial

6.2.1. Distribución temporal

La situación del desarrollador permite su dedicación a tiempo completo. Por lo tanto, se establece un trabajo de 8 horas diarias, comenzando el lunes 7 de mayo y terminando el lunes 11 de junio. De esta manera, son 5 semanas con 40 horas de trabajo cada una, lo que resulta en un total de 200 horas.

El tiempo va a dividirse en un total de tres sprints:

- Sprint 1: empieza el lunes 7 de mayo y termina el lunes 21 de mayo, con una duración de dos semanas. Se contempla que el 14 de mayo es festivo en la localidad de Valladolid.
- Sprint 2: empieza el martes 22 de mayo y termina el lunes 4 de junio, con una duración de dos semanas.
- Sprint 3: empieza el martes 5 de junio y termina el lunes 11 de junio, con una duración de una semana.

6.2.2. Entorno tecnológico

Para la realización del proyecto es necesario emplear una serie de recursos software y hardware.

Recursos software

A continuación se muestra la lista de productos software empleados para la realización del proyecto.

- **Google Mail (versión web)**: para la comunicación con los tutores.
- **ShareLatex (versión web)**: para la redacción de memoria y manuales de instalación y uso.
- **Mozilla Firefox Quantum 60 (64-bit)**: para el acceso a los recursos web.
- **Astah Professional 7.2**: para la realización de diagramas de clases.
- **Unity 2017.3.1f1**: para la implementación del videojuego.
- **MonoDevelop 5.9.6**: para la edición de ficheros fuente en C#.
- **Dropbox 52.4**: para la sincronización de los archivos del proyecto en la nube.
- **GIMP 2.8**: para edición de imagen y creación de recursos visuales.
- **Tiled 1.1.5**: para creación de mapas 2D importables a Unity.
- **Tiled2Unity 1.0.13.2**: para exportar mapas de Tiled a Unity.

Recursos hardware

En las tablas 6.1 y 6.2 se muestran los elementos hardware empleados a lo largo del proyecto.

| Ordenador portátil | |
|---------------------------|---------------------------------|
| ASUS A53SJ | |
| Hardware | |
| Procesador | Intel Core i7-2630QM @ 2.00 GHz |
| Memoria | Crucial SO-DIMM 4GBx2 Total 8GB |
| Placa base | Asus A53SJ |
| Disco duro | Seagate 500GB |
| Tarjeta Gráfica | nVidia GeForce GT520M CUDA 1GB |
| Software | |
| Sistema Operativo | Windows 10 Pro, Version 1803 |
| Arquitectura | 64 bits |

Tabla 6.1: Ordenador portátil empleado durante el desarrollo.

| Monitor | |
|---------------------|-----------|
| ASUS VX238H 23" | |
| Relación de aspecto | 16:9 |
| Resolución | 1920x1080 |

Tabla 6.2: Monitor empleado durante el desarrollo.

6.2.3. Estimación de costes

En este epígrafe se detallan los costes previstos al inicio del proyecto. El salario del trabajador se trata en bruto y todos los costes asociados a recursos incluyen impuestos.

Salario del trabajador

El salario medio de un desarrollador de software en España, en junio de 2018, es de 25.712 € brutos anuales. Lo que se traduce en un coste de 16,98 € por hora [44]. Este es el coste que tendría si se remunerase al desarrollador de software, en este caso, Luis Alberto Centeno Bragado.

Licencias de software

Dado que se ha optado por emplear software libre en ciertos casos, y software con licencia gratuita en el resto de casos, no existen costes asociados a este efecto.

Material

Se tiene en cuenta el coste del material físico que se va a emplear, en el momento de su compra.

Espacio de trabajo y otros

Existen otros recursos asociados al proyecto, como es la necesidad de tener un espacio de trabajo, conexión a internet, luz, etc. Para simplificar, vamos a agrupar esos gastos optando por una opción en un espacio de coworking. Un espacio de coworking es aquel que es compartido por trabajadores de distintas empresas o autónomos. En Valladolid existen estas opciones, que ya incluyen conexión a internet, luz y el propio espacio. El coste estimado es de 150 € al mes.

Costes totales

En la tabla 6.3 se muestra que el precio total de la realización del proyecto es de 4.491 €.

| Concepto | Precio Unitario | Unidades | Subtotal |
|---------------------------|-----------------|-----------|----------------|
| Ordenador portátil | 666 € | 1 | 666 € |
| Monitor | 129 € | 1 | 129 € |
| Espacio de coworking | 150 € | 2 meses | 300 € |
| Salario del desarrollador | 16,98 € | 200 horas | 3.396 € |
| TOTAL | | | 4.491 € |

Tabla 6.3: Estimación de costes del desarrollo del proyecto.

6.2.4. Pruebas a realizar

Las historias de usuario en Scrum no se consideran finalizadas si antes no se han realizado unas pruebas de verificación que la misma historia describe.

Todas las pruebas que se realizarán son manuales, sin implementar ningún mecanismo de pruebas automáticas como en los sistemas de integración continua, ni test unitarios. Esto es debido a la complejidad de estas técnicas en videojuegos hechos con Unity.

6.2.5. Análisis de riesgos

En este punto se detallan una serie de riesgos que pueden implicar retrasos en el desarrollo del proyecto, indicando la probabilidad de que ocurra cada uno, su retraso estimado en días y el plan de acción.

| ID | Riesgo | Probabilidad | Descripción | Retraso estimado |
|----|-----------------------------|--------------|---|------------------|
| 1 | Enfermedad | 0,33 | El desarrollador cae en una enfermedad que le impide trabajar. | 2 |
| 2 | Inexperiencia con Unity | 0,5 | Se ralentiza el desarrollo debido a la falta de conocimientos de Unity. | 7 |
| 3 | Planificación no adecuada | 0,4 | No se planifica debidamente y no se tienen en cuenta los riesgos con el suficiente rigor. | 15 |
| 4 | Fallo en el material físico | 0,05 | Alguno de los dispositivos necesarios para el desarrollo deja de funcionar. | 1 |
| 5 | Perdida de datos por fallo | 0,02 | Alguno de los dispositivos necesarios deja de funcionar provocando pérdida de datos. | 2 |
| 6 | Fallo en suministro | 0,01 | El suministro de conexión a internet o luz falla por parte de los proveedores de servicios. | 2 |

Tabla 6.4: Posibles riesgos durante el desarrollo del proyecto.

| ID | Riesgo | Exposición al riesgo (en días) | Plan de acción |
|---------------|-----------------------------|--------------------------------|---|
| 1 | Enfermedad | 0,66 | Recuperar las horas perdidas fuera de horario. |
| 2 | Inexperiencia con Unity | 3,5 | Dedicar horas fuera de horario para aprender sobre Unity. |
| 3 | Planificación no adecuada | 6 | Replantear inmediatamente la planificación del proyecto. |
| 4 | Fallo en el material físico | 0,05 | Suplir inmediatamente el material estropeado para retomar el trabajo normal cuanto antes. |
| 5 | Perdida de datos por fallo | 0,04 | Realizar copias de seguridad diarias. |
| 6 | Fallo en suministro | 0,02 | Transladarse inmediatamente a otro espacio de trabajo que cuente con suministro. |
| TOTAL: | | 10,27 | |

Tabla 6.5: Plan de acción ante los riesgos.

6.3. Historias de usuario

Los requisitos y objetivos de una aplicación informática, cuando se emplea Scrum, se definen a través de historias de usuario. En este epígrafe vamos a describirlas todas conformando el Product Backlog, es decir, una lista priorizada de las historias de usuario. Cada historia muestra un valor y una estimación en horas.

| US-01 | |
|--|-----------------------|
| Mapa de juego escuela | |
| Valor: 3000 | Estimación: 15 |
| Descripción: Como cliente, quiero que la aplicación tenga un mapa principal e inicial llamado escuela, para que el jugador se meta en la historia del juego. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • Al arrancar el juego se muestra el fragmento del mapa que la cámara sea capaz de abarcar. • El mapa contendrá más de 10 puertas y/o pasadizos. • El mapa tendrá una temática relacionada con el universo Harry Potter. | |

Tabla 6.6: Historia de usuario US-01.

| US-02 | |
|---|-----------------------|
| Objeto jugador | |
| Valor: 3000 | Estimación: 10 |
| Descripción: Como cliente, quiero que la aplicación tenga un objeto, para que el jugador pueda moverse por el mundo. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El objeto se mueve empleando las flechas del teclado o las teclas W, A, S, D, en la dirección correspondiente a cada tecla. • El objeto se mueve en 8 sentidos: arriba, abajo, izquierda, derecha y cuatro diagonales. | |

Tabla 6.7: Historia de usuario US-02.

| US-03 | |
|--|----------------------|
| Cámara sobre el jugador | |
| Valor: 3000 | Estimación: 7 |
| Descripción: Como cliente, quiero que la cámara se sitúe siempre encima del objeto del jugador, para que el jugador no lo pierda de vista y se pierda en el espacio. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • La cámara se mueve sobre el jugador, al mismo ritmo. | |

Tabla 6.8: Historia de usuario US-03.

| US-04 | |
|--|-----------------------|
| Animación del objeto del jugador | |
| Valor: 2900 | Estimación: 12 |
| Descripción: Como cliente, quiero que el objeto tenga una animación que simule que esta andando cuando se esta moviendo por el mundo, para que el jugador sienta el juego como real. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El objeto tiene una animación por cada dirección de movimiento. • El objeto emplea la animación adecuada en cada dirección. | |

Tabla 6.9: Historia de usuario US-04.

| US-05 | |
|--|----------------------|
| Colisiones en mapa escuela | |
| Valor: 3000 | Estimación: 8 |
| Descripción: Como cliente, quiero que el mapa escuela presente colisiones en sus limites y en las paredes, para que el jugador no pueda salir del mapa y perderse en el espacio. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El objeto del jugador no puede salir del mapa por ninguna de sus paredes o límites. | |

Tabla 6.10: Historia de usuario US-05.

| US-06 | |
|---|----------------------|
| Colisiones con objetos en mapa escuela | |
| Valor: 2900 | Estimación: 8 |
| Descripción: Como cliente, quiero que el objeto del jugador choque contra los objetos en el mapa escuela, para que el jugador sienta una experiencia de juego correcta. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador no puede caminar sobre estanterías, arboles, etc. | |

Tabla 6.11: Historia de usuario US-06.

| US-07 | |
|---|-----------------------|
| Mapa de desafío 1 | |
| Valor: 2500 | Estimación: 10 |
| Descripción: Como cliente, quiero que la aplicación tenga un mapa secundario llamado desafío 1, para que el jugador afronte el primer desafío del juego. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El mapa contiene una puerta de salida. • Tendrá una temática relacionada con el universo Harry Potter. | |

Tabla 6.12: Historia de usuario US-07.

| US-08 | |
|---|----------------------|
| Colisiones en mapa desafío1 | |
| Valor: 2500 | Estimación: 5 |
| Descripción: Como cliente, quiero que el mapa desafío1 presente colisiones en sus límites y en las paredes, para que el jugador no pueda salir del mapa y perderse en el espacio. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador no puede salir del mapa por ninguna de sus paredes o límites. | |

Tabla 6.13: Historia de usuario US-08.

| US-09 | |
|--|----------------------|
| Colisiones con objetos en mapa desafío1 | |
| Valor: 2500 | Estimación: 5 |
| Descripción: Como cliente, quiero que el objeto del jugador choque contra los objetos en el mapa desafío1, para que el jugador sienta una experiencia de juego correcta. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador no puede caminar sobre estanterías, arboles, etc. | |

Tabla 6.14: Historia de usuario US-09.

| US-10 | |
|---|-----------------------|
| Mapa de desafío 2 | |
| Valor: 2500 | Estimación: 10 |
| Descripción: Como cliente, quiero que la aplicación tenga un mapa secundario llamado desafío 2, para que el jugador afronte el primer desafío del juego. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El mapa contiene una puerta de salida. • Tendrá una temática relacionada con el universo Harry Potter. | |

Tabla 6.15: Historia de usuario US-10.

| US-11 | |
|---|----------------------|
| Colisiones en mapa desafio2 | |
| Valor: 2500 | Estimación: 5 |
| Descripción: Como cliente, quiero que el mapa desafio2 presente colisiones en sus limites y en las paredes, para que el jugador no pueda salir del mapa y perderse en el espacio. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El objeto del jugador no puede salir del mapa por ninguna de sus paredes o límites. | |

Tabla 6.16: Historia de usuario US-11.

| US-12 | |
|--|----------------------|
| Colisiones con objetos en mapa desafio2 | |
| Valor: 2500 | Estimación: 5 |
| Descripción: Como cliente, quiero que el objeto del jugador choque contra los objetos en el mapa desafio2, para que el jugador sienta una experiencia de juego correcta. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El objeto del jugador no puede caminar sobre estanterías, arboles, etc. | |

Tabla 6.17: Historia de usuario US-12.

| US-13 | |
|---|-----------------------|
| Mapa de desafío 3 | |
| Valor: 2500 | Estimación: 10 |
| Descripción: Como cliente, quiero que la aplicación tenga un mapa secundario llamado desafío 3, para que el jugador afronte el primer desafío del juego. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El mapa contiene una puerta de salida. • Tendrá una temática relacionada con el universo Harry Potter. | |

Tabla 6.18: Historia de usuario US-13.

| US-14 | |
|---|----------------------|
| Colisiones en mapa desafio3 | |
| Valor: 2500 | Estimación: 5 |
| Descripción: Como cliente, quiero que el mapa desafio3 presente colisiones en sus limites y en las paredes, para que el jugador no pueda salir del mapa y perderse en el espacio. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador no puede salir del mapa por ninguna de sus paredes o límites. | |

Tabla 6.19: Historia de usuario US-14.

| US-15 | |
|--|----------------------|
| Colisiones con objetos en mapa desafio3 | |
| Valor: 2500 | Estimación: 5 |
| Descripción: Como cliente, quiero que el objeto del jugador choque contra los objetos en el mapa desafio3, para que el jugador sienta una experiencia de juego correcta. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador no puede caminar sobre estanterías, arboles, etc. | |

Tabla 6.20: Historia de usuario US-15.

| US-16 | |
|--|-----------------------|
| Teletransporte mapa escuela y desafio1 | |
| Valor: 2500 | Estimación: 12 |
| Descripción: Como cliente, quiero que el objeto del jugador pueda teletransportarse entre el mapa escuela y desafio1, para que el jugador pueda moverse en ambos mapas. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador se teletransporta desde el mapa escuela a desafio1 chocando con una de las puertas del primer mapa. • El objeto del jugador se teletransporta desde el mapa desafio1 a escuela chocando con una de las puertas del primero. | |

Tabla 6.21: Historia de usuario US-16.

| US-17 | |
|--|----------------------|
| Teletransporte mapa escuela y desafio2 | |
| Valor: 2500 | Estimación: 8 |
| Descripción: Como cliente, quiero que el objeto del jugador pueda teletransportarse entre el mapa escuela y desafio2, para que el jugador pueda moverse en ambos mapas. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • El objeto del jugador se teletransporta desde el mapa escuela a desafio2 chocando con una de las puertas del primer mapa. • El objeto del jugador se teletransporta desde el mapa desafio2 a escuela chocando con una de las puertas del primero. | |

Tabla 6.22: Historia de usuario US-17.

| US-18 | |
|--|----------------------|
| Teletransporte mapa escuela y desafio3 | |
| Valor: 2500 | Estimación: 8 |
| Descripción: Como cliente, quiero que el objeto del jugador pueda teletransportarse entre el mapa escuela y desafio3, para que el jugador pueda moverse en ambos mapas. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El objeto del jugador se teletransporta desde el mapa escuela a desafio3 chocando con una de las puertas del primer mapa. • El objeto del jugador se teletransporta desde el mapa desafio3 a escuela chocando con una de las puertas del primero. | |

Tabla 6.23: Historia de usuario US-18.

| US-19 | |
|--|-----------------------|
| Formulario validar desafio1 | |
| Valor: 2000 | Estimación: 12 |
| Descripción: Como cliente, quiero que en el mapa desafio1 uno al tocar un objeto aparezca un menú para validar la primera prueba, para que el jugador pueda validar el resultado de la primera prueba. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • Aparece un menú al tocar un objeto. • Valida la respuesta de la prueba. | |

Tabla 6.24: Historia de usuario US-19.

| US-20 | |
|--|-----------------------|
| Formulario validar desafio2 | |
| Valor: 2000 | Estimación: 10 |
| Descripción: Como cliente, quiero que en el mapa desafio2 uno al tocar un objeto aparezca un menú para validar la segunda prueba, para que el jugador pueda validar el resultado de la segunda prueba. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • Aparece un menú al tocar un objeto. • Valida la respuesta de la prueba. | |

Tabla 6.25: Historia de usuario US-20.

| US-21 | |
|--|-----------------------|
| Formulario validar desafio3 | |
| Valor: 2000 | Estimación: 10 |
| Descripción: Como cliente, quiero que en el mapa desafio3 uno al tocar un objeto aparezca un menú para validar la tercera prueba, para que el jugador pueda validar el resultado de la tercera prueba. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • Aparece un menú al tocar un objeto. • Valida la respuesta de la prueba. | |

Tabla 6.26: Historia de usuario US-21.

| US-22 | |
|---|----------------------|
| Mensaje fin juego | |
| Valor: 1500 | Estimación: 2 |
| Descripción: Como cliente, quiero que cuando se valide la ultima prueba se muestre un mensaje de fin de juego, para que el jugador sea informado de que ha terminado. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • Al terminar la tercera prueba se muestra el mensaje de fin de juego. | |

Tabla 6.27: Historia de usuario US-22.

| US-23 | |
|--|----------------------|
| Menu pausa | |
| Valor: 1500 | Estimación: 2 |
| Descripción: Como cliente, quiero que al pulsar la tecla ESC se pause el juego y se muestre el mensaje "PAUSA" que al volver a pulsar ESC se salga de ese menú, para que el jugador pueda pausar el juego. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • Al pulsar ESC se muestra el menú de pausa. • Al volver a pulsar ESC se sale del menú de pausa. | |

Tabla 6.28: Historia de usuario US-23.

| US-24 | |
|---|----------------------|
| Suavizado cámara | |
| Valor: 1500 | Estimación: 6 |
| Descripción: Como cliente, quiero que la cámara tenga un movimiento suavizado respecto al objeto jugador, para que el jugador tenga una mejor experiencia de juego. | |
| Condiciones de satisfacción: | |
| <ul style="list-style-type: none"> • La cámara de mueve de forma suavizada. | |

Tabla 6.29: Historia de usuario US-24.

| US-25 | |
|---|----------------------|
| Orden de desafíos | |
| Valor: 1600 | Estimación: 5 |
| Descripción: Como cliente, quiero que solo se puedan realizar los desafíos en orden, para que el jugador progrese en el juego de forma ordenada. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • Sólo se puede acceder al mapa desafío2 tras haber resuelto la primera prueba. • Sólo se puede acceder al mapa desafío3 tras haber resuelto la primera y la segunda prueba. | |

Tabla 6.30: Historia de usuario US-25.

| US-26 | |
|---|----------------------|
| Mensaje objetivo en pausa | |
| Valor: 1000 | Estimación: 2 |
| Descripción: Como cliente, quiero que el menú de pausa muestre un mensaje sobre el objetivo actual, para que le jugador esté informado de tal objetivo. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El menú de pausa muestra el mensaje sobre el objetivo actual. | |

Tabla 6.31: Historia de usuario US-26.

| US-27 | |
|---|----------------------|
| Botón de salida | |
| Valor: 1000 | Estimación: 2 |
| Descripción: Como cliente, quiero que el menú de pausa tenga un botón que permita cerrar el juego, para que el jugador pueda salir del juego en cualquier momento. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El menú de pausa muestra el botón de salida de juego. • Al pulsar el botón se cierra el juego. | |

Tabla 6.32: Historia de usuario US-27.

| US-28 | |
|---|----------------------|
| Citado de fuentes | |
| Valor: | Estimación: 1 |
| Descripción: Como cliente, quiero que el menú de pausa muestre un mensaje con las fuentes empleadas para sprites y tilesets, para cumplir con la licencia de dichos recursos. | |
| Condiciones de satisfacción: <ul style="list-style-type: none"> • El menú de pausa muestra el mensaje con las fuentes. | |

Tabla 6.33: Historia de usuario US-28.

6.4. Implementación

En este epígrafe se muestra la evolución de la implementación separada por sprints. En cada sprint se desarrolla el sprint backlog, los detalles de implementación, las pruebas realizadas.

6.4.1. Sprint 1

El sprint 1 comienza el lunes 7 de mayo y termina el lunes 21 de mayo, durando dos semanas, con 80 horas de trabajo.

Sprint Backlog

Las historias de usuario que se implementarán en este sprint son:

US-01, US-02, US-03, US-04, US-05, US-06, US-07, US-08 y US-09.

Pruebas realizadas

A continuación se muestra una lista de pruebas realizadas, clasificadas por historia de usuario. No todas las historias de usuario han requerido la realización de una prueba.

| US-01-T01 | |
|--------------------------------|---|
| Título | Mapa escuela |
| Comportamiento esperado | Al arrancar el juego, se muestra el fragmento del mapa escuela que abarque la cámara. |
| Resultado de la prueba | OK |

Tabla 6.34: Pruebas realizadas sobre la historia US-01.

| US-02-T01 | |
|--------------------------------|--|
| Título | Uso de las flechas para mover al jugador |
| Comportamiento esperado | Las flechas del teclado mueven de forma adecuada al objeto del jugador en las 8 direcciones. |
| Resultado de la prueba | OK |

| US-02-T02 | |
|--------------------------------|---|
| Título | Uso de las teclas W, A, S y D para mover al jugador |
| Comportamiento esperado | Las teclas W, A, S y D mueven de forma adecuada al objeto del jugador en las 8 direcciones. |
| Resultado de la prueba | OK |

Tabla 6.35: Pruebas realizadas sobre la historia US-02.

| US-03-T01 | |
|--------------------------------|--|
| Título | Seguimiento de la cámara al jugador |
| Comportamiento esperado | Cuando el jugador se desplaza, la cámara se mantiene moviéndose con él a lo largo del mundo. |
| Resultado de la prueba | OK |

Tabla 6.36: Pruebas realizadas sobre la historia US-03.

| US-04-T01 | |
|--------------------------------|--|
| Título | Animación del objeto jugador |
| Comportamiento esperado | Cuando el objeto jugador se mueve en cada una de las direcciones, muestra la animación adecuada que simula que está andando. |
| Resultado de la prueba | Fallo. Al moverse a la derecha, emplea la animación de debe emplear para moverse a la izquierda. |
| US-04-T02 | |
| Título | Animación del objeto jugador. Segundo intento. |
| Comportamiento esperado | El mismo que en US-04-T1. |
| Resultado de la prueba | OK |

Tabla 6.37: Pruebas realizadas sobre la historia US-04.

| US-05-T01 | |
|--------------------------------|--|
| Título | Colisiones en mapa escuela |
| Comportamiento esperado | Cuando el objeto jugador se choca con una pared o un limite del mapa, se para y no puede traspasarlo. |
| Resultado de la prueba | Fallo. En una pared del mapa no está implementada la colisión y el personaje puede salir del mapa. |
| US-05-T02 | |
| Título | Colisiones en mapa escuela. Segundo intento |
| Comportamiento esperado | El esperado en US-05-T1. |
| Resultado de la prueba | Fallo. En una región del mapa, al chocar con la pared, el objeto del jugador se mueve bruscamente, traspasando la pared. |
| US-05-T03 | |
| Título | Colisiones en mapa escuela. Tercer intento. |
| Comportamiento esperado | El esperado en US-05-T1. |
| Resultado de la prueba | OK |

Tabla 6.38: Pruebas realizadas sobre la historia US-05.

| US-06-T01 | |
|--------------------------------|--|
| Título | Colisiones con objetos en mapa escuela |
| Comportamiento esperado | Al chocar con un objeto del mapa, el jugador se detiene. |
| Resultado de la prueba | OK |

Tabla 6.39: Pruebas realizadas sobre la historia US-06.

| US-08-T01 | |
|--------------------------------|---|
| Título | Colisiones en mapa desafio1 |
| Comportamiento esperado | Cuando el objeto jugador se choca con una pared o un limite del mapa, se para y no puede traspasarlo. |
| Resultado de la prueba | OK |

Tabla 6.40: Pruebas realizadas sobre la historia US-08.

| US-09-T01 | |
|--------------------------------|--|
| Título | Colisiones con objetos en mapa desafio1 |
| Comportamiento esperado | Al chocar con un objeto del mapa, el jugador se detiene. |
| Resultado de la prueba | OK |

Tabla 6.41: Pruebas realizadas sobre la historia US-09.

Detalles de implementación

Para comenzar, se ha creado una escena de Unity que incorporará todos los elementos necesarios para el juego. Tras la implementación de las historias de usuario correspondientes a este primer sprint, se han incorporado los siguientes elementos a la escena. Se indican, como estereotipo, el tipo de elemento de Unity que representa a cada clase.

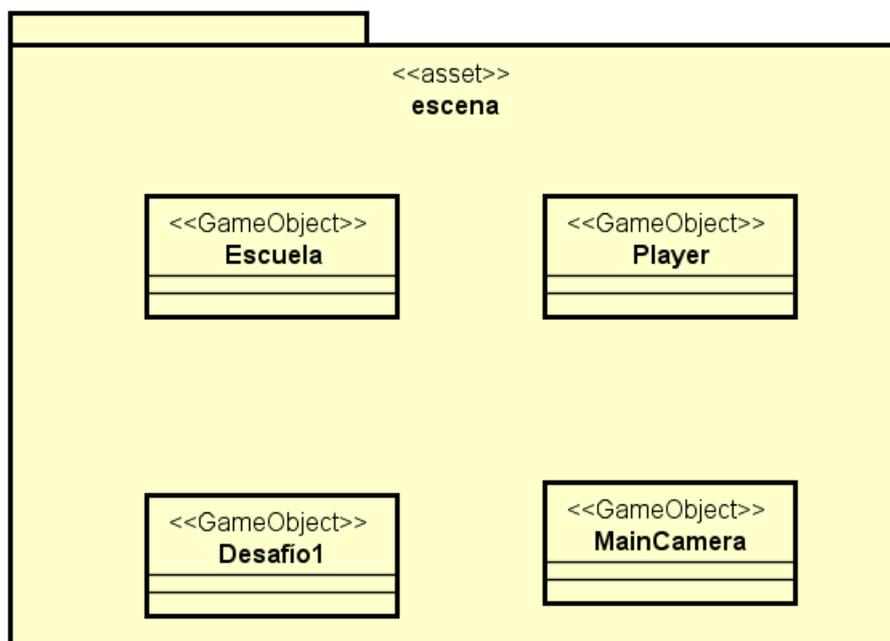


Figura 6.2: Escena del juego tras la implementación del primer sprint.

A su vez, cada uno de los objetos incorporados a la escena tiene vinculación con otros objetos y componentes de Unity.

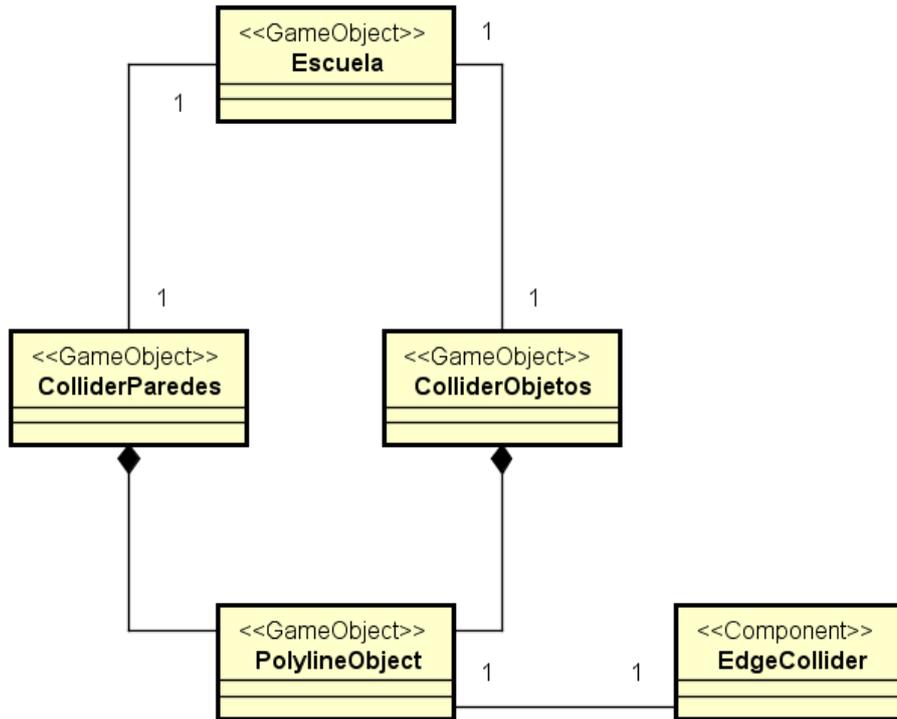


Figura 6.3: Objeto que implementa el mapa escuela tras el primer sprint.

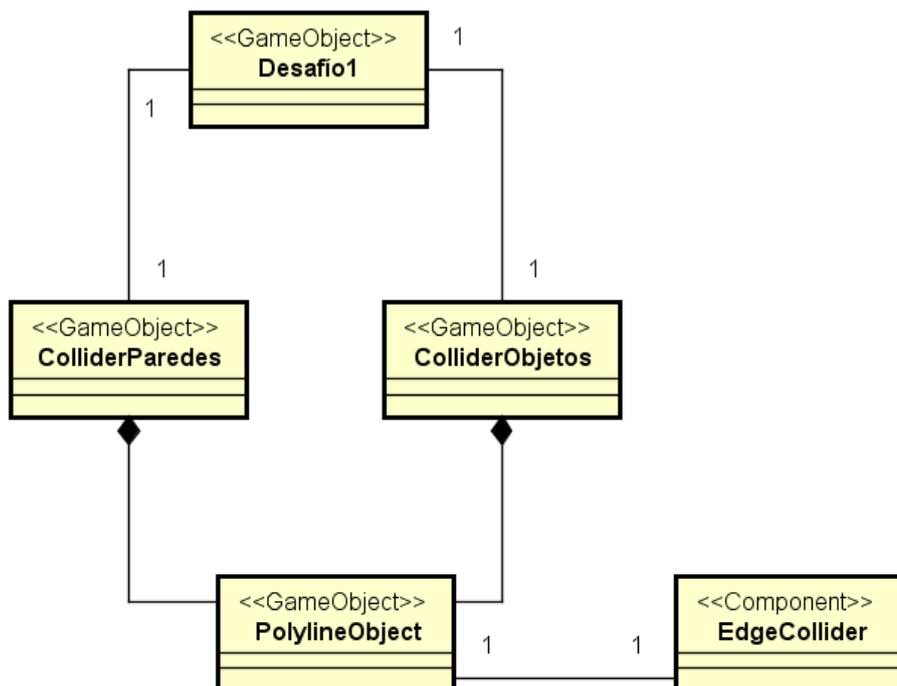


Figura 6.4: Objeto que implementa el mapa desafio1 tras el primer sprint.

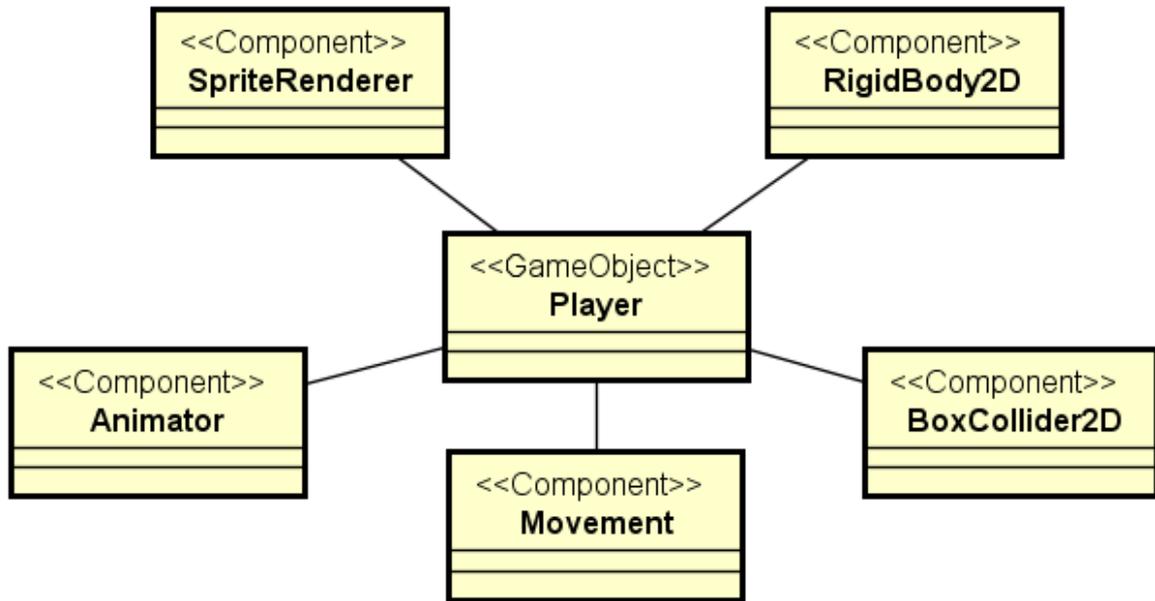


Figura 6.5: Objeto que implementa el objeto del jugador tras el primer sprint.

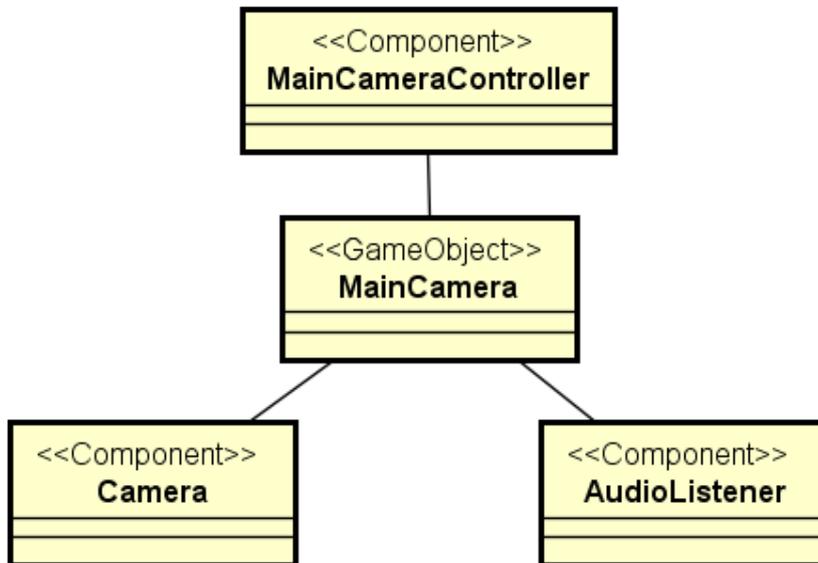


Figura 6.6: Objeto que implementa la cámara tras el primer sprint.

6.4.2. Sprint 2

El sprint 2 comienza el martes 22 de mayo y termina el lunes 4 de junio, durando dos semanas y con 80 horas de trabajo.

Burn Down Chart

El siguiente es el gráfico de burn down, que muestra una relación entre tareas pendientes y tiempo restante de proyecto tras haber realizado el sprint 1.

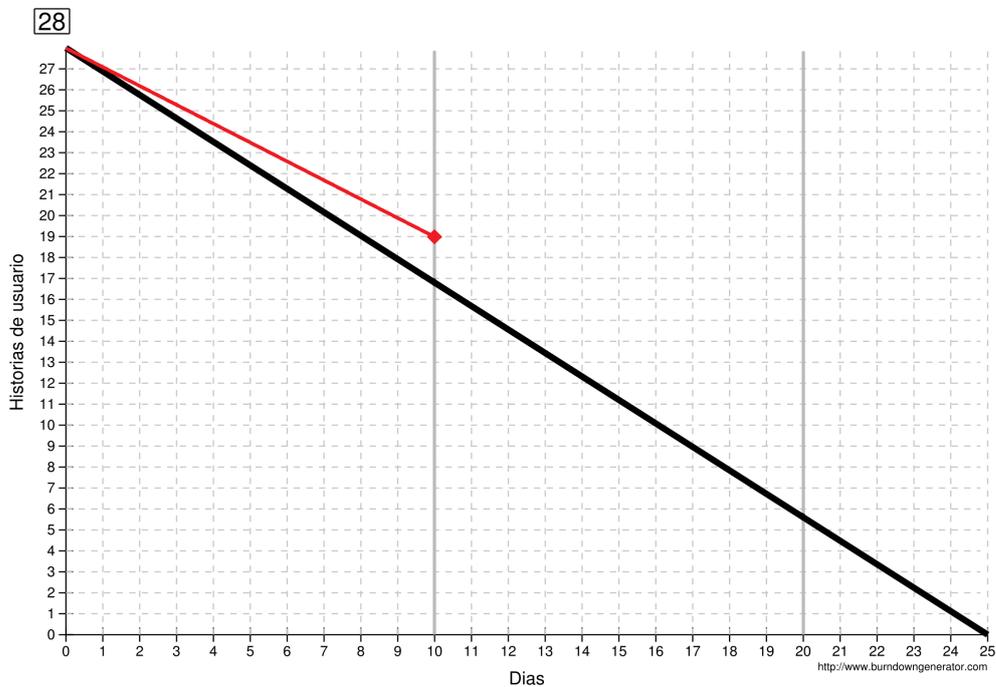


Figura 6.7: Gráfico Burn Down tras el primer sprint.

Aunque se ha cumplido con la planificación, podemos ver que las tareas pendientes son ligeramente superiores a las que deberían quedar. Esto es debido a que la estimación de duración de las primeras tareas es mayor que las restantes.

Sprint Backlog

Las historias de usuario que se implementarán en este sprint son:

US-10, US-11, US-12, US-13, US-14, US-15, US-16, US-17, US-18 y US-19.

Pruebas realizadas

A continuación se muestra una lista de pruebas realizadas, clasificadas por historia de usuario. No todas las historias de usuario han requerido la realización de una prueba.

| US-11-T01 | |
|--------------------------------|---|
| Título | Colisiones en mapa desafio2 |
| Comportamiento esperado | Cuando el objeto jugador se choca con una pared o un limite del mapa, se para y no puede traspasarlo. |
| Resultado de la prueba | OK |

Tabla 6.42: Pruebas realizadas sobre la historia US-11.

| US-12-T01 | |
|--------------------------------|--|
| Título | Colisiones con objetos en mapa desafio2 |
| Comportamiento esperado | Al chocar con un objeto del mapa, el jugador se detiene. |
| Resultado de la prueba | OK |

Tabla 6.43: Pruebas realizadas sobre la historia US-12.

| US-14-T01 | |
|--------------------------------|---|
| Título | Colisiones en mapa desafio3 |
| Comportamiento esperado | Cuando el objeto jugador se choca con una pared o un limite del mapa, se para y no puede traspasarlo. |
| Resultado de la prueba | OK |

Tabla 6.44: Pruebas realizadas sobre la historia US-14.

| US-15-T01 | |
|--------------------------------|--|
| Título | Colisiones con objetos en mapa desafio3 |
| Comportamiento esperado | Al chocar con un objeto del mapa, el jugador se detiene. |
| Resultado de la prueba | OK |

Tabla 6.45: Pruebas realizadas sobre la historia US-15.

| US-16-T01 | |
|--------------------------------|--|
| Título | Teletransporte escuela-desafio1 |
| Comportamiento esperado | Cuando el objeto jugador estando en el mapa escuela se mete en el pasadizo que conduce hacia desafio1, se teletransporta a ese mapa. |
| Resultado de la prueba | Fallo. Se entra en bucle infinito de teletransportes. |
| US-16-T02 | |
| Título | Teletransporte escuela-desafio1. Segundo intento. |
| Comportamiento esperado | El mismo que en US-16-T1. |
| Resultado de la prueba | OK |
| US-16-T03 | |
| Título | Teletransporte desafio1-escuela |
| Comportamiento esperado | Cuando el objeto jugador estando en el mapa desafio1 se mete en el pasadizo que conduce hacia escuela, se teletransporta a ese mapa. |
| Resultado de la prueba | OK |

Tabla 6.46: Pruebas realizadas sobre la historia US-16.

| US-17-T01 | |
|--------------------------------|--|
| Título | Teletransporte escuela-desafio2 |
| Comportamiento esperado | Cuando el objeto jugador estando en el mapa escuela se mete en el pasadizo que conduce hacia desafio2, se teletransporta a ese mapa. |
| Resultado de la prueba | OK |
| US-17-T02 | |
| Título | Teletransporte desafio2-escuela |
| Comportamiento esperado | Cuando el objeto jugador estando en el mapa desafio2 se mete en el pasadizo que conduce hacia escuela, se teletransporta a ese mapa. |
| Resultado de la prueba | OK |

Tabla 6.47: Pruebas realizadas sobre la historia US-17.

| US-18-T01 | |
|--------------------------------|--|
| Título | Teletransporte escuela-desafio3 |
| Comportamiento esperado | Cuando el objeto jugador estando en el mapa escuela se mete en el pasadizo que conduce hacia desafio3, se teletransporta a ese mapa. |
| Resultado de la prueba | OK |
| US-18-T02 | |
| Título | Teletransporte desafio3-escuela |
| Comportamiento esperado | Cuando el objeto jugador estando en el mapa desafio3 se mete en el pasadizo que conduce hacia escuela, se teletransporta a ese mapa. |
| Resultado de la prueba | OK |

Tabla 6.48: Pruebas realizadas sobre la historia US-18.

| US-19-T01 | |
|--------------------------------|--|
| Título | Formulario prueba 1 |
| Comportamiento esperado | Cuando el objeto jugador toca el objeto del primer desafío, aparece un formulario. |
| Resultado de la prueba | Fallo. El formulario no aparece y se produce un error en tiempo de ejecución. |
| US-19-T02 | |
| Título | Formulario prueba 1. Segundo intento |
| Comportamiento esperado | El esperado en US-19-T01. |
| Resultado de la prueba | OK |
| US-19-T03 | |
| Título | Validación correcta prueba 1 |
| Comportamiento esperado | Tras introducir el texto correcto, da por correcta la prueba. |
| Resultado de la prueba | OK |
| US-19-T04 | |
| Título | Validación incorrecta prueba 1 |
| Comportamiento esperado | Tras introducir un texto incorrecto, da por incorrecta la prueba. |
| Resultado de la prueba | OK |
| US-19-T05 | |
| Título | Validación vacía prueba 1 |
| Comportamiento esperado | Tras introducir cadena vacía, da por incorrecta la prueba. |
| Resultado de la prueba | OK |

Tabla 6.49: Pruebas realizadas sobre la historia US-19.

Detalles de implementación

A la escena de Unity creada en el anterior sprint se le han añadido nuevos objetos de juego. Estos objetos se muestran de color verde en el siguiente diagrama.

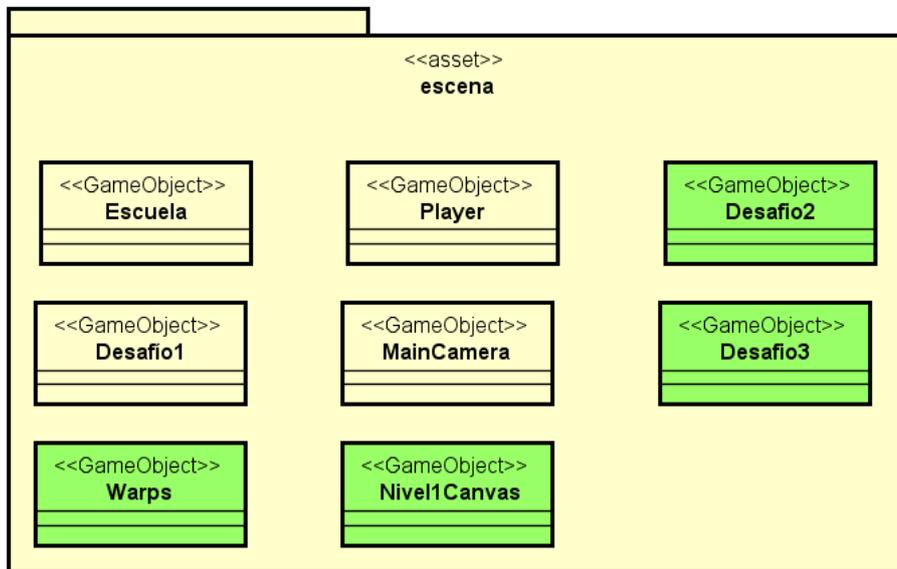


Figura 6.8: Escena del juego tras la implementación del segundo sprint.

Cada uno de los objetos nuevos se vinculan con otros de la forma en que detallan los siguientes esquemas.

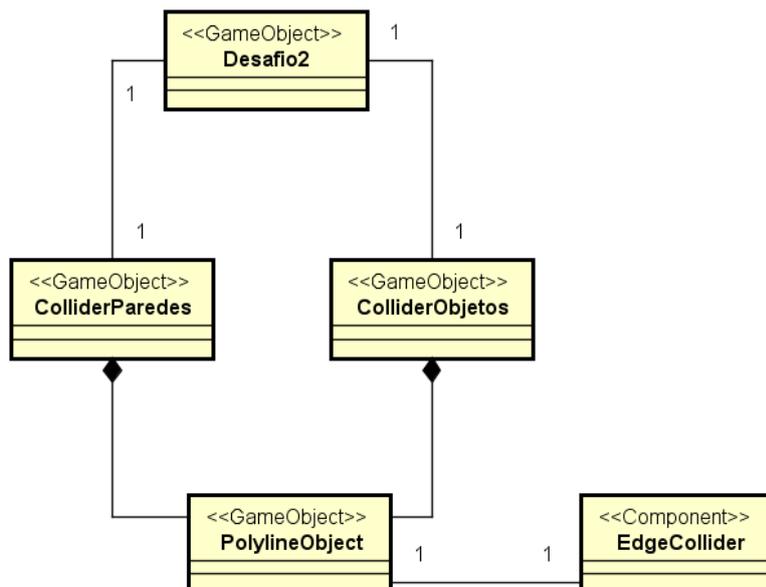


Figura 6.9: Objeto que implementa el mapa desafio2 tras el segundo sprint.

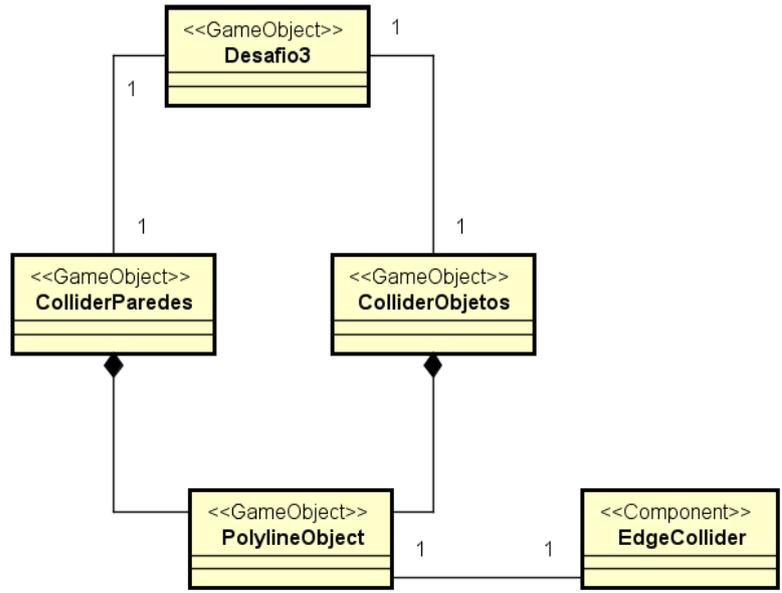


Figura 6.10: Objeto que implementa el mapa desafio3 tras el segundo sprint.

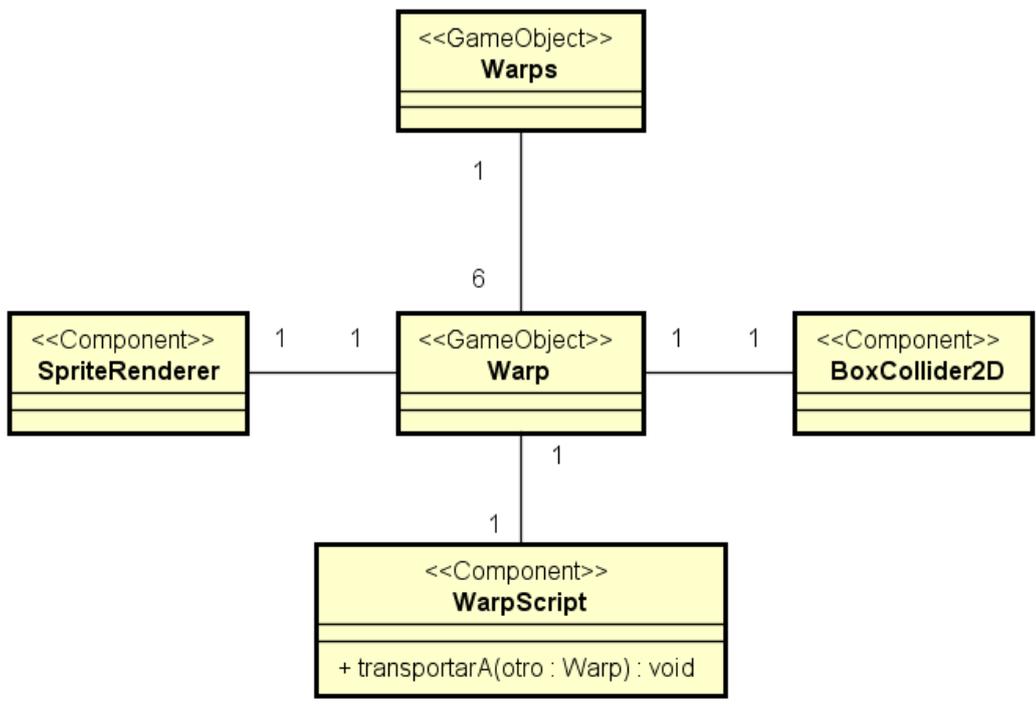


Figura 6.11: Objeto que implementa los puntos de teletransporte entre mapas tras el segundo sprint.

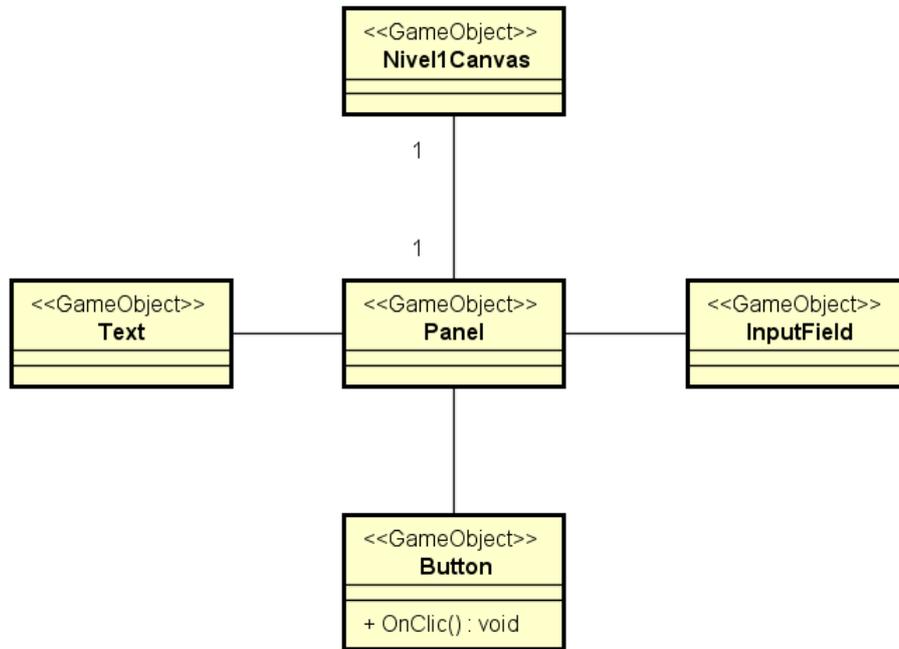


Figura 6.12: Objeto que implementa el formulario de la prueba 1 tras el segundo sprint.

6.4.3. Sprint 3

El sprint 3 comienza el martes 5 de junio y termina el lunes 11 de junio, durando una semana y con 40 horas de trabajo.

Burn Down Chart

El siguiente es el gráfico de burn down, que muestra una relación entre tareas pendientes y tiempo restante de proyecto tras haber realizado el sprint 2.

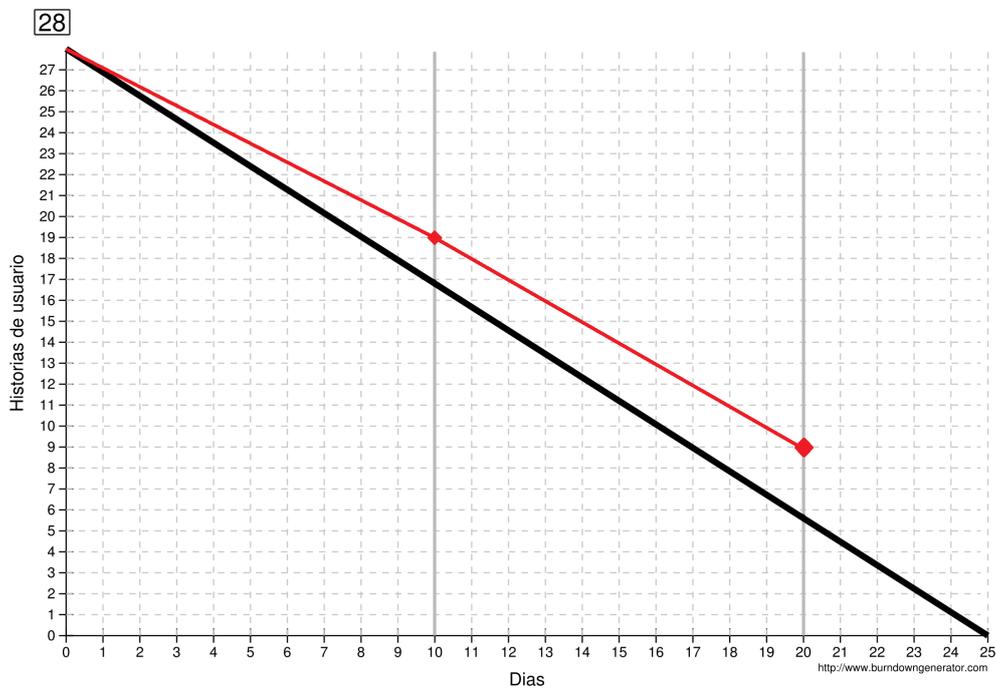


Figura 6.13: Gráfico Burn Down tras el segundo sprint.

De nuevo, se cumple la previsión, pero las tareas pendientes son superiores a lo que debería quedar. Una vez más, las tareas pendientes tienen una duración menor que las que se han realizado.

Sprint Backlog

Las historias de usuario que se implementarán en este sprint son:

US-20, US-21, US-22, US-23, US-24, US-25, US-27, US-27 y US-28.

Pruebas realizadas

A continuación se muestra una lista de pruebas realizadas, clasificadas por historia de usuario.

| US-20-T01 | |
|--------------------------------|---|
| Título | Formulario prueba 2 |
| Comportamiento esperado | Cuando el objeto jugador toca el objeto del segundo desafío, aparece un formulario. |
| Resultado de la prueba | OK |
| US-20-T02 | |
| Título | Validación correcta prueba 2 |
| Comportamiento esperado | Tras introducir el texto correcto, da por correcta la prueba. |
| Resultado de la prueba | OK |
| US-20-T03 | |
| Título | Validación incorrecta prueba 2 |
| Comportamiento esperado | Tras introducir un texto incorrecto, da por incorrecta la prueba. |
| Resultado de la prueba | OK |
| US-20-T04 | |
| Título | Validación vacía prueba 2 |
| Comportamiento esperado | Tras introducir cadena vacía, da por incorrecta la prueba. |
| Resultado de la prueba | OK |

Tabla 6.50: Pruebas realizadas sobre la historia US-20.

| US-21-T01 | |
|--------------------------------|--|
| Título | Formulario prueba 3 |
| Comportamiento esperado | Cuando el objeto jugador toca el objeto del tercer desafío, aparece un formulario. |
| Resultado de la prueba | OK |
| US-21-T02 | |
| Título | Validación correcta prueba 3 |
| Comportamiento esperado | Tras introducir el texto correcto, da por correcta la prueba. |
| Resultado de la prueba | OK |
| US-21-T03 | |
| Título | Validación incorrecta prueba 3 |
| Comportamiento esperado | Tras introducir un texto incorrecto, da por incorrecta la prueba. |
| Resultado de la prueba | OK |
| US-21-T04 | |
| Título | Validación vacía prueba 3 |
| Comportamiento esperado | Tras introducir cadena vacía, da por incorrecta la prueba. |
| Resultado de la prueba | OK |

Tabla 6.51: Pruebas realizadas sobre la historia US-21.

| US-22-T01 | |
|--------------------------------|--|
| Título | Mensaje fin juego |
| Comportamiento esperado | Al terminar la tercera prueba se muestra un mensaje de fin de juego. |
| Resultado de la prueba | Fallo. El mensaje no aparece. |
| US-22-T02 | |
| Título | Mensaje fin juego. Segundo intento |
| Comportamiento esperado | El mismo que US-22-T01. |
| Resultado de la prueba | OK |

Tabla 6.52: Pruebas realizadas sobre la historia US-22.

| US-23-T01 | |
|--------------------------------|--|
| Título | Poner pausa |
| Comportamiento esperado | Al pulsar ESC durante el juego, este se pausa. |
| Resultado de la prueba | OK |
| US-23-T02 | |
| Título | Quitar pausa |
| Comportamiento esperado | Al pulsar ESC mientras se está en pausa, se vuelve al juego. |
| Resultado de la prueba | OK |

Tabla 6.53: Pruebas realizadas sobre la historia US-23.

| US-24-T01 | |
|--------------------------------|--|
| Título | Suavizado cámara |
| Comportamiento esperado | La cámara se mueve después de que el objeto jugador empiece a moverse y termina también cierto tiempo después. Acelera y frena de forma suavizada. |
| Resultado de la prueba | OK |

Tabla 6.54: Pruebas realizadas sobre la historia US-24.

| US-25-T01 | |
|--------------------------------|---|
| Título | Orden de desafíos valido |
| Comportamiento esperado | Se resuelven los retos en orden 1, 2, 3. |
| Resultado de la prueba | OK |
| US-25-T02 | |
| Título | Orden de desafíos incorrecto. Caso A |
| Comportamiento esperado | Al intentar hacer el reto 3 antes que el 1, no se puede acceder al mapa desafio3. |
| Resultado de la prueba | OK |
| US-25-T03 | |
| Título | Orden de desafíos incorrecto. Caso B |
| Comportamiento esperado | Al intentar hacer el reto 2 antes que el 1, no se puede acceder al mapa desafio2. |
| Resultado de la prueba | OK |
| US-25-T04 | |
| Título | Orden de desafíos incorrecto. Caso C |
| Comportamiento esperado | Se realiza el reto 1. Al intentar hacer el reto 3 antes que el 2, no se puede acceder al mapa desafio3. |
| Resultado de la prueba | OK |

Tabla 6.55: Pruebas realizadas sobre la historia US-25.

| US-26-T01 | |
|--------------------------------|---|
| Título | Mensaje objetivo 1 |
| Comportamiento esperado | Antes de completar el primer objetivo, en el menú de pausa se muestra el mensaje sobre el desafío 1. |
| Resultado de la prueba | OK |
| US-26-T02 | |
| Título | Mensaje objetivo 2 |
| Comportamiento esperado | Después de completar el primer objetivo y antes de completar el segundo, en el menú de pausa se muestra el mensaje sobre el desafío 2. |
| Resultado de la prueba | OK |
| US-26-T03 | |
| Título | Mensaje objetivo 3 |
| Comportamiento esperado | Después de completar el segundo objetivo y antes de completar el tercero, en el menú de pausa se muestra el mensaje sobre el desafío 2. |
| Resultado de la prueba | OK |

Tabla 6.56: Pruebas realizadas sobre la historia US-26.

| US-27-T01 | |
|--------------------------------|---|
| Título | Mostrar botón salir |
| Comportamiento esperado | En el menú de pausa se muestra el botón para salir del juego. |
| Resultado de la prueba | OK |

| US-27-T02 | |
|--------------------------------|---|
| Título | Funcionamiento botón salir |
| Comportamiento esperado | Al hacer clic en el botón de salir del juego, el juego se cierra. |
| Resultado de la prueba | OK |

Tabla 6.57: Pruebas realizadas sobre la historia US-27.

| US-28-T01 | |
|--------------------------------|---|
| Título | Mostrar mensaje de citado |
| Comportamiento esperado | En el menú de pausa se muestra un mensaje con información sobre las fuentes de sprites y tiles. |
| Resultado de la prueba | OK |

Tabla 6.58: Pruebas realizadas sobre la historia US-28.

Detalles de implementación

A la escena de Unity se le han añadido nuevos objetos de juego. Estos objetos se muestran de color verde en el siguiente diagrama.

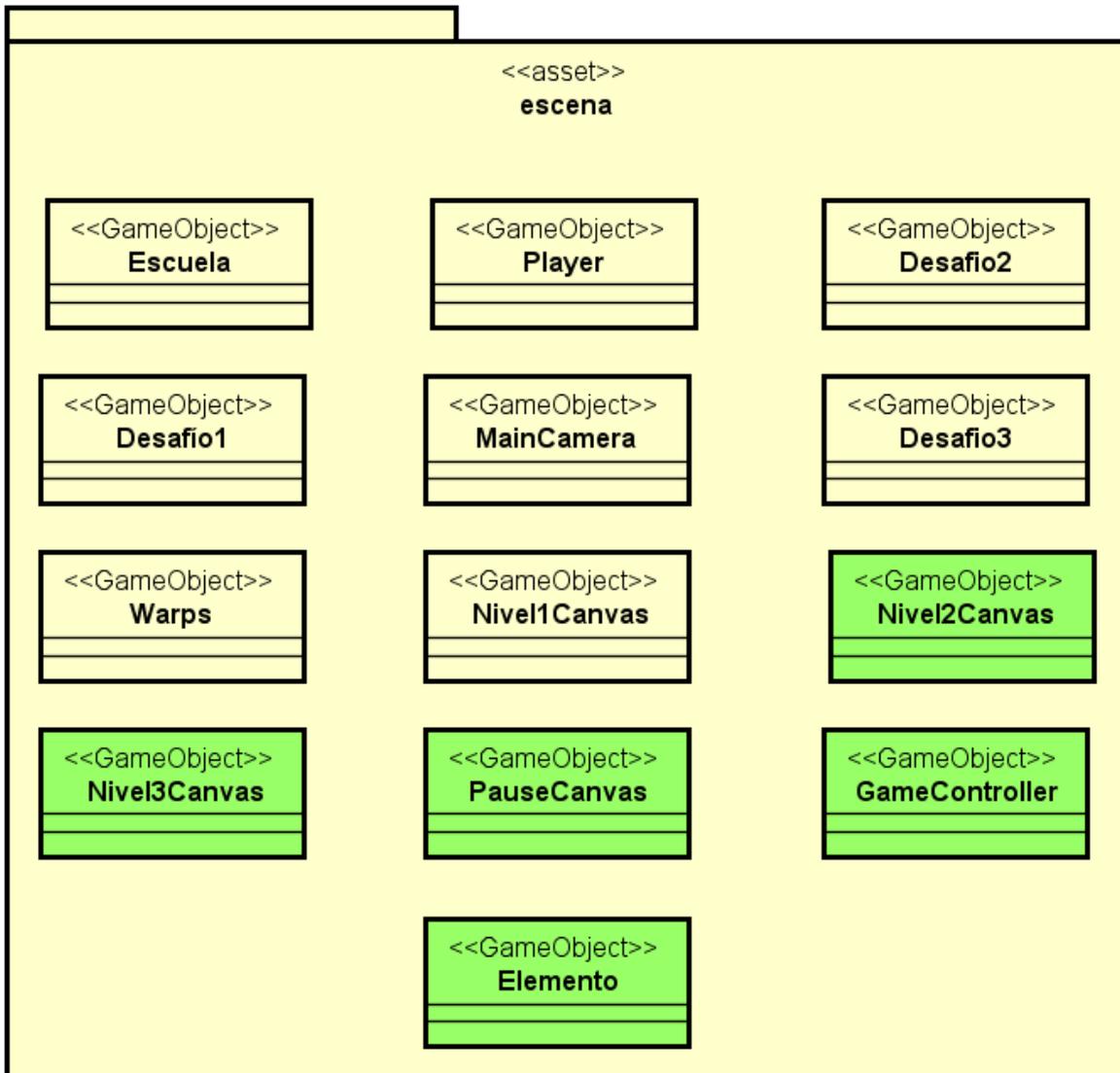


Figura 6.14: Escena del juego tras la implementación del tercer sprint.

Proyecto: Yincana virtual de programación

Cada uno de los objetos nuevos se vinculan con otros de la forma en que detallan los siguientes esquemas.

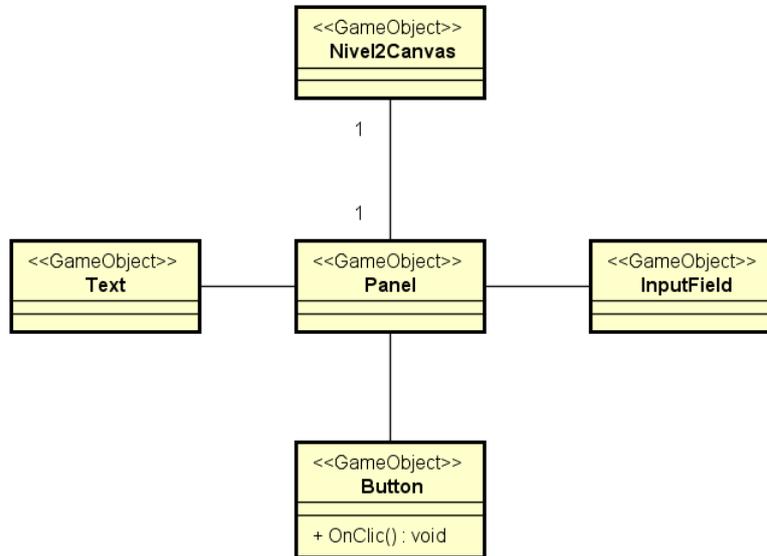


Figura 6.15: Objeto que implementa el formulario de la prueba 2 tras el tercer sprint.

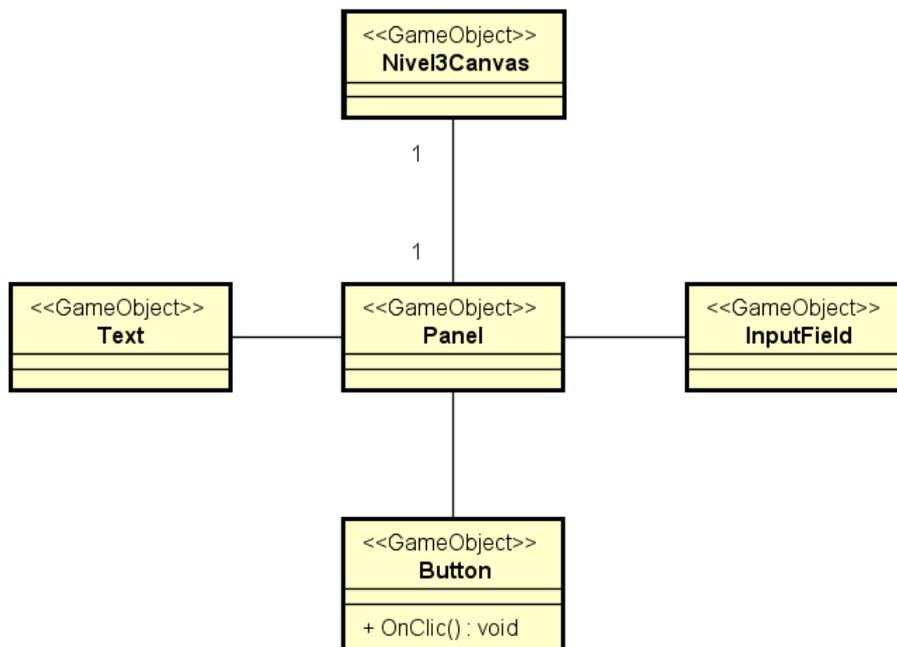


Figura 6.16: Objeto que implementa el formulario de la prueba 3 tras el tercer sprint.

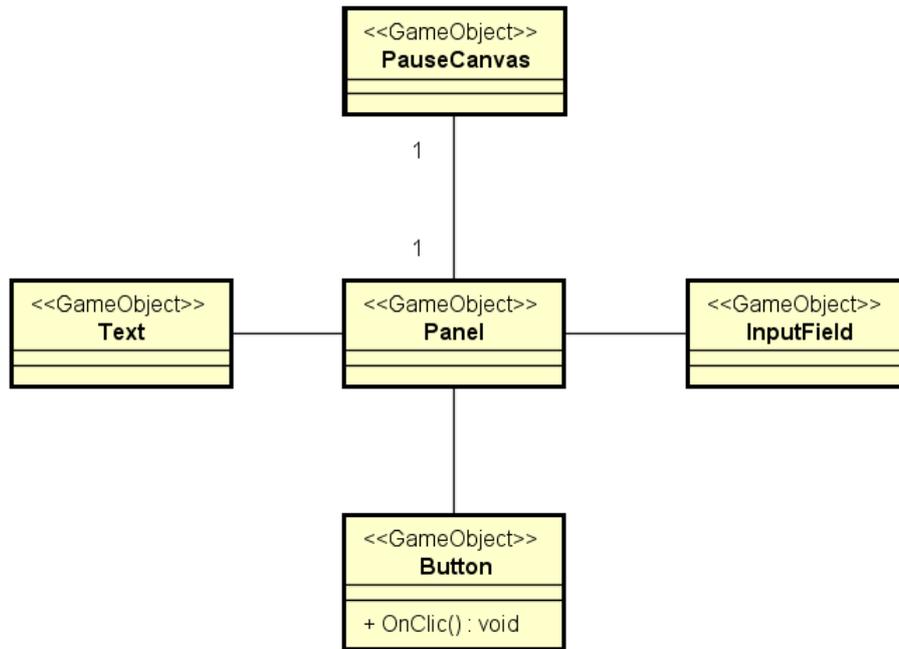


Figura 6.17: Objeto que implementa el menú de pausa tras el tercer sprint.

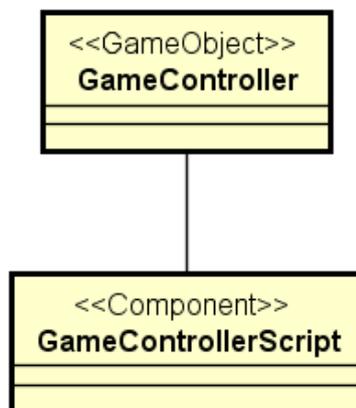


Figura 6.18: Objeto que implementa el controlador del juego tras el tercer sprint.

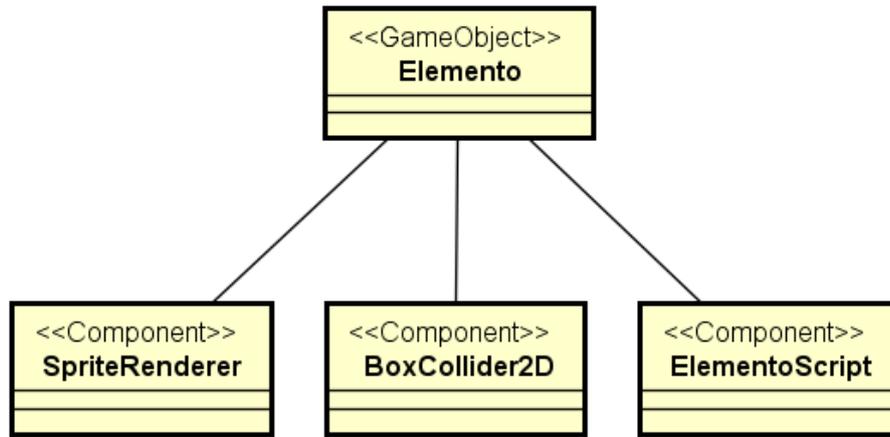


Figura 6.19: Objeto que implementa el menú de pausa tras el tercer sprint.

6.5. Estado final del proyecto

Tras la finalización del tercer y último sprint, en este epígrafe se describe cuál es el estado final del producto.

Se han implementado todas las historias de usuario descritas en el apartado 6.3 de este trabajo. Por lo que puede darse el proyecto por finalizado.

El gráfico burn down, tras el último sprint, queda de la siguiente manera.

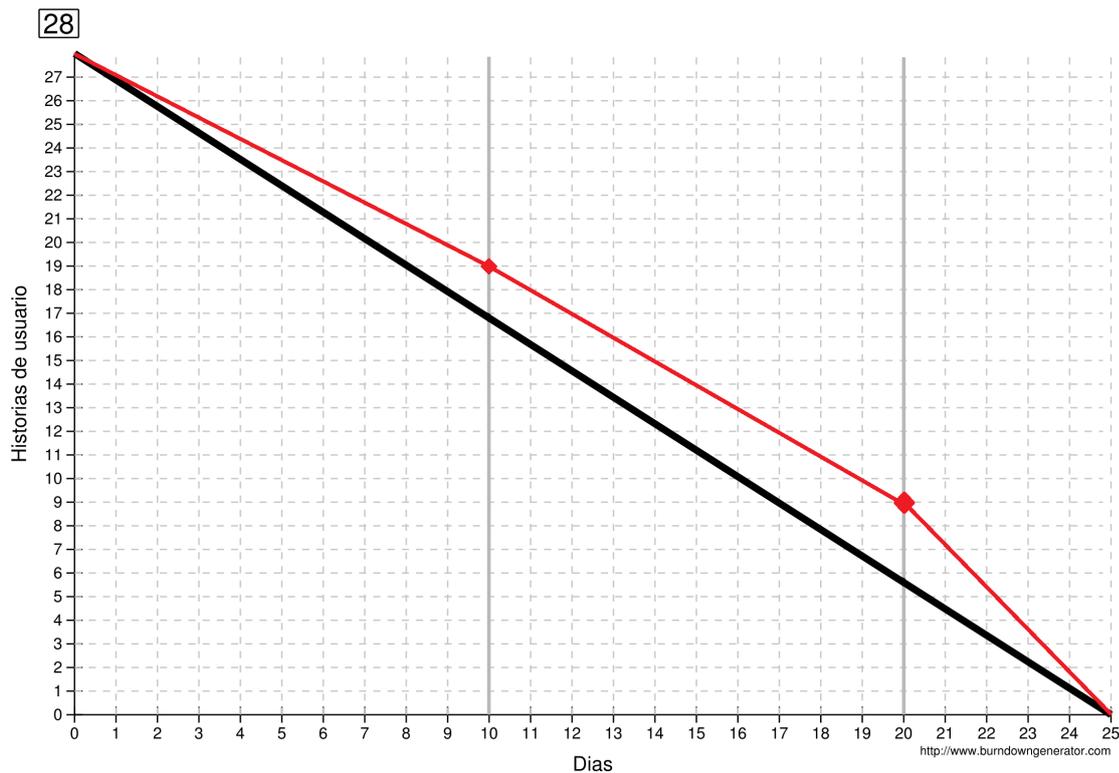


Figura 6.20: Gráfico Burn Down tras el tercer sprint.

Podemos ver que a lo largo de todo el proyecto, nos hemos mantenido por encima de la línea de burn down. Esto es debido a que la estimación de las últimas tareas era notablemente menor que las que se implementaron en los dos primeros sprints.

Respecto a los fallos encontrados en la aplicación una vez que se ha terminado, únicamente podemos encontrar un comportamiento extraño. Tras la realización de la US-24, sobre el suavizado del movimiento de la cámara, podemos encontrar un movimiento no esperado cuando el jugador se teletransporta desde cualquiera de los mapas de desafío hacia el mapa escuela. No obstante, se cumple con las condiciones de satisfacción de esa historia de usuario.

Capítulo 7

Conclusiones

En este capítulo se recogen diversas impresiones que a las que se han llegado tras el desarrollo de este trabajo.

Se ha comprobado que la incorporación de elementos de juego en los sistemas de aprendizaje es una práctica que repercute positivamente tanto en el estudiante como en el profesor. El estudiante se divierte aprendiendo, y el profesor ofrece un método de enseñanza distinto a lo convencional, que además le aporta feedback de forma inmediata.

El proyecto realizado no es el primero que trata de emplear la gamificación para la enseñanza de la programación. Ya existen herramientas online para este propósito. Sin embargo, ninguna de ellas incorpora la posibilidad de que varios jugadores participen de forma cooperativa. Por lo tanto, la herramienta creada puede ser pionera en ese aspecto.

Se ha descubierto como se construyen los videojuegos a partir de motores como Unity, lo cual es un ámbito no impartido a lo largo del plan de estudios y que ha supuesto un reto personal muy interesante. Se han encontrado dificultades durante el proceso de desarrollo, ya que la forma de trabajar con estas herramientas difiere notablemente a que se tiene mientras se crean otro tipo de programas de ordenador.

Además, se han podido aplicar conocimientos adquiridos durante la carrera, como aquellos relacionados con programación, interacción persona-computadora o ingeniería del software.

Finalmente, el desarrollo del proyecto para la realización del prototipo de producto se ha realizado bajo una planificación, poniendo en práctica conocimientos aprendidos dentro y fuera del aula que tienen que ver con el desarrollo ágil.

Capítulo 8

Líneas de trabajo futuro

Podemos destacar dos posibles líneas de trabajo futuro: extensiones de funcionalidad y corrección de comportamientos inesperados.

8.1. Extensiones de funcionalidad

Como ya se ha mencionado, el proyecto realizado en este Trabajo de Fin de Grado se trata de un prototipo que implementa una de las características de la yincana de programación. A partir de este prototipo se puede trabajar para extender su funcionalidad. A continuación se definen algunas propuestas al respecto.

- Con el prototipo, los enunciados de las pruebas se entregan al alumno en PDF. Se podría hacer que el juego muestre dichos enunciados por si mismo.
- Incorporar la característica multijugador, por la cuál los jugadores cooperen para lograr los objetivos del juego.
- De la misma manera, se puede añadir la posibilidad de que dos jugadores compitan para ver quién consigue terminar el juego primero.
- Se puede incorporar un sistema de puntuación al juego, de tal forma que se permita medir el progreso y la calidad con la que se avanza durante el juego. Esas medidas se podrían emplear para la realización de estadísticas que ayuden al docente en la mejora del propio juego.
- Actualmente, solo se permite jugar a un juego en el que los resultados son estáticos, que además están en el propio código del juego. Se podría elaborar una herramienta que permita escoger temática del juego, variando los mapas, y que también pueda cambiar los enunciados de las pruebas.

8.2. Corrección de comportamientos inesperados

Las pruebas realizadas han cubierto la mayor parte de la funcionalidad del juego. Sin embargo, como se ha dicho en el apartado 6.5 de esta memoria sobre el estado final del proyecto, se ha detectado un comportamiento anómalo.

Este comportamiento consiste en que, al volver al mapa *escuela* desde cualquier otro, la cámara del juego realiza un movimiento inesperado. Aunque este fallo no impide jugar, ya que su efecto dura menos de un segundo, afecta a la experiencia de juego y puede ser corregido.

Apéndices

Apéndice A

Acrónimos

- **CD**: Compact Disc, en español, disco compacto.
- **GB**: Gigabytes.
- **GHz**: Giga Hercios.
- **IDE**: Integrated Development Environment, en español, Entorno de Desarrollo Integrado.
- **JAR**: Java Archive, en español, archivo Java.
- **JENUI**: Jornadas sobre la Enseñanza Universitaria de la Informática.
- **LAN**: Local Area Network, en español, Red de Area Local.
- **MB**: Megabytes.
- **MOOC**: Massive Open Online Course, en español, Curso Online Masivo y Abierto.
- **PDF**: Portable Document Format, en español, formato de documento portátil.
- **TFG**: Trabajo Fin de Grado.
- **UNIR**: Universidad Internacional de La Rioja.
- **URL**: Uniform Resource Locator, en español, Localizador de Recurso Uniforme.
- **USD**: United States Dollar, en español, Dolares Estadounidenses.
- **VoIP**: Voice over Internet Protocol, en español, Voz sobre Protocolo de Internet.

Apéndice B

Manual de instalación

Antes de instalar el juego que implementa deberá tener en cuenta los requisitos mínimos de sistema que requiere el programa:

- **Sistema operativo:** Microsoft Windows 10 de 64 bits.
- **Procesador:** Intel Core i3 ó AMD A6 @ 2GHz, ó superior.
- **Memoria RAM:** 1 GB.
- **Espacio libre en disco:** 50 MB.
- **Resolución de pantalla:** 1024x768.

Para instalar el juego, necesitará el CD proporcionado. Los pasos son los siguientes:

1. Copie el fichero "juego.zip" que se encuentra en la carpeta raíz del CD.
2. Pegue el fichero en el directorio que usted desee.
3. Descomprima el fichero "juego.zip".

Una vez seguidos estos pasos, haga doble clic en el fichero "juego.exe" para ejecutar.

Apéndice C

Manual de uso

Para iniciar el juego, haga doble clic en el fichero “juego.exe”. Se le mostrará una ventana como la siguiente:

Escoja los parámetros que más se adecúen a su sistema y haga clic en el botón “Play!”.

Una vez dentro del juego, utilice las flechas del teclado para andar en la dirección que desee. También puede emplear las teclas W, A S y D para el mismo propósito.

Para mostrar el menú de pausa, pulse la tecla escape.

Si desea salir del juego, entre en el menú de pausa y haga clic en el botón “Salir del juego”.

Siga las instrucciones dadas por el profesor para progresar en el juego.

Apéndice D

Contenido del CD

El CD proporcionado contiene los siguientes ficheros:

- **memoria.pdf**: documento con la memoria del Trabajo de Fin de Grado en formato PDF.
- **juego.zip**: fichero comprimido con el juego listo para instalarse. Siga las instrucciones indicadas en el manual de instalación.
- **fuentes.zip**: fichero comprimido con el código fuente realizado para implementar el juego.

Apéndice E

Enunciados de los desafíos de la yincana virtual

Primera prueba

Llegó tu momento. Por fin estás en Hogwarts, lo que siempre habías querido. Te vas a embarcar en una aventura en la que irás descubriendo los misterios que aguarda esta fantástica escuela. Tu primera tarea es hacerte con un mineral capaz de convertir el oro en plomo, y con la que podemos hacer la poción elixir de vida.

Lo has adivinado. Tienes que encontrar la piedra filosofal. Está escondida en una de las salas de este castillo. ¿Cuál? Responde al test que está en el dorso para averiguarlo. Después, con tus respuestas puedes usar el mapa para encontrar el pasadizo que deberás tomar.

Y antes de coger la piedra, deberás...

CALCULAR EL PRODUCTO DE LOS NÚMEROS MÚLTIPLOS DE 3 Y 5 (DE AMBOS A LA VEZ) COMPENDIDOS ENTRE EL 100 Y 200.

1. La notación BNF es...

- a) Un estándar de descripción
- b) Una forma de representación de algoritmos
- c) Una lenguaje de programación de alto nivel

2. Dada la siguiente cabecera de un método Java:

```
public static int pregunta(int a, double b){ ... }
```

Y las siguientes declaraciones de variables:

```
int n; int m = 5; double x = 2.4;
```

Indicar qué sentencia de las siguientes es correcta:

- a) `n = pregunta(m, m);`
- b) `n = pregunta(m, x);`
- c) `pregunta(m, x);`

3. Dados los siguientes fragmentos de código Java:

| Código 1 | Código 2 | Código 3 |
|---|---|---|
| <pre>if (x <= 0) x++; else x-;</pre> | <pre>if (x <= 0) x++; if (x >0) x-;</pre> | <pre>boolean b = (x <= 0); if (b) x++; if (!b) x-;</pre> |

Determinar cuál de las siguientes afirmaciones es correcta

- a) El código 1 es equivalente al código 2
- b) El código 1 es equivalente al código 3
- c) El código 2 es equivalente al código 3

Enunciados de los desafíos de la yincana virtual

Segunda prueba

El que no debe ser nombrado estudió en Hogwarts, y su legado aun perdura. Hace poco te hiciste con el diario de Tom Marvolo Riddle, un antiguo estudiante que conocía bien a Voldemort, pero ¿recuerdas dónde lo has dejado?

Usa uno de los pasadizos para encontrarlo. Resuelve las preguntas al dorso para saber cual debes usar.

Antes de coger el diario deberás...

CALCULAR LA SUMA DE LOS 900 PRIMEROS MÚLTIPLOS DE 7 (SIN CONSIDERAR EL 0)

1. La secuencia de caracteres (`x == 1`) en Java es...

- a) Incorrecta
- b) Un sentencia de asignación
- c) Un expresión de tipo boolean

2. Dada la siguiente cabecera de un método Java:

```
public static void pregunta(int a, int b, double c){ ... }
```

Y las siguientes declaraciones de variables:

```
int n; int m = 5; double x = 2.4;
```

Indicar qué sentencia de las siguientes es correcta:

- a) `pregunta(m, m, m);`
- b) `pregunta(m, m, x);`
- c) `n = pregunta(m, m, x);`

3. Dados los siguientes fragmentos de código Java:

```
int i;
for (i = 1; i<=10; i++)
    if (i<10)
        System.out.print(Math.sqrt(i) + “, ”);
    else
        System.out.println(Math.sqrt(i) + “.”);
-----
int i;
for (i = 1; i<=9; i++)
    System.out.print(Math.sqrt(i) + “, ”);
    System.out.println(Math.sqrt(i) + “.”);
```

Determinar cuál de las siguientes afirmaciones es correcta:

- a) Escriben lo mismo, pero es más eficiente el segundo
- b) Escriben lo mismo, pero es más eficiente el primero
- c) No escriben lo mismo

Enunciados de los desafíos de la yincana virtual

Tercera prueba

Cada vez estás más cerca. Si superas este nivel te harás con la gran victoria. Y con el gran trofeo: la ansiada copa de los tres magos. Tan solo deberás cruzar el laberinto que conduce hasta él.

¿Quieres saber dónde está el laberinto? Responde las preguntas del dorso.

Antes de hacerte con la copa deberás...

CALCULAR EL MÁXIMO COMÚN DIVISOR DE 1322382 Y 739878

-
1. ¿Quién es el creador del lenguaje Python?
 - a) Bill Gates
 - b) Guido Van Rossum
 - c) Steve Jobs

 2. ¿Cuál de los siguientes no es un tipo de lenguaje de programación?
 - a) Compilado
 - b) Interpretado
 - c) Tanteado

 3. ¿Quién está considerada como la primera persona que programó en la historia?
 - a) Ada Byron
 - b) Donald Knuth
 - c) Alan Turing

Enunciados de los desafíos de la yincana virtual

Utiliza este mapa para saber qué pasadizo tomar con tus respuestas.

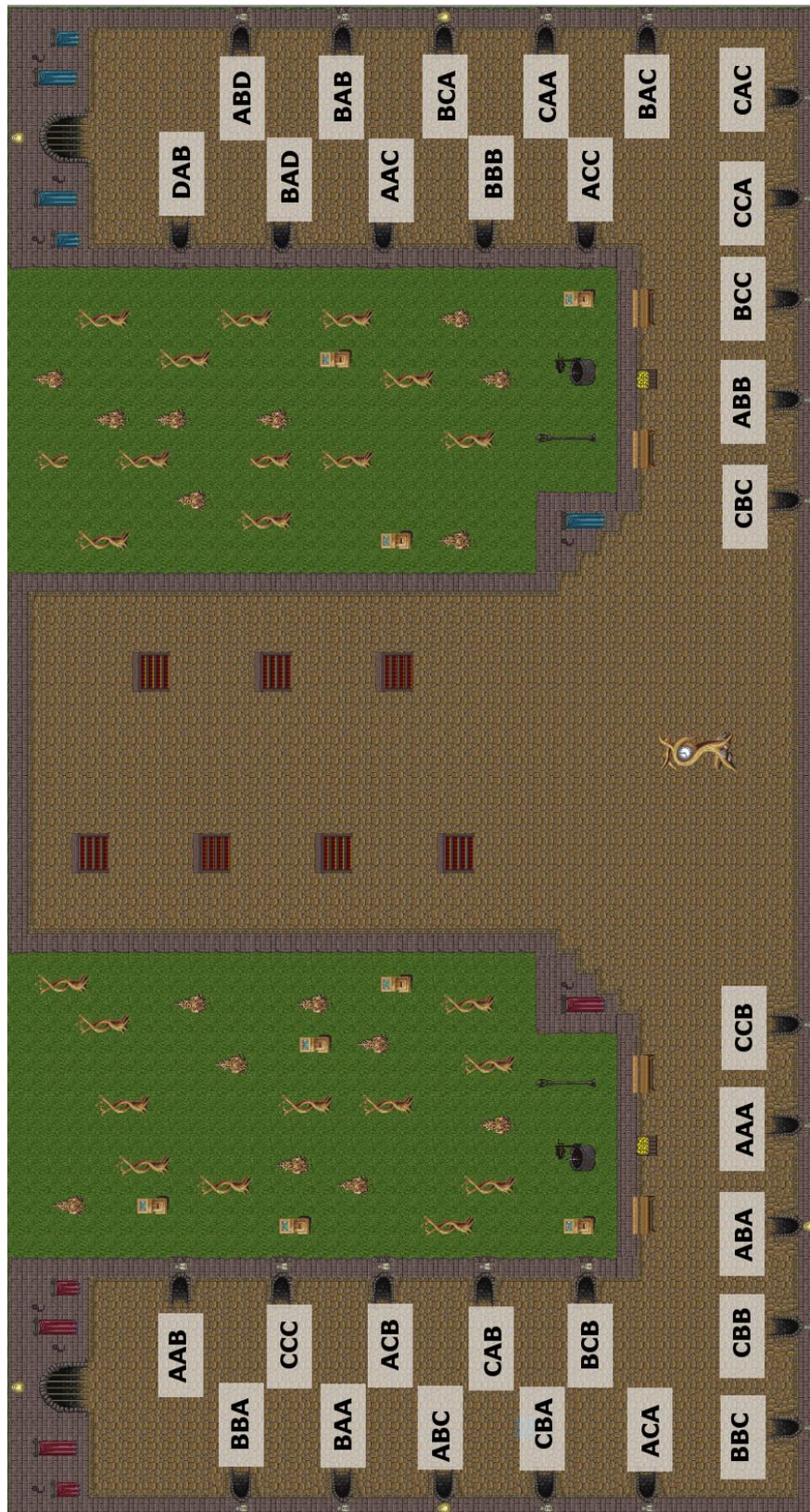


Figura E.1: Mapa de la escuela.

Bibliografía

- [1] D. P. Carracedo and A. P. Marrón, *El juego en los adolescentes como plataforma de aprendizaje de la tecnología en secundaria*. Universidad de Valladolid, 2015. [Online]. Available: <http://uvadoc.uva.es/handle/10324/15223>
- [2] Diccionario de la lengua española: Juego. Visitado: 2018-04-10. [Online]. Available: <http://dle.rae.es/?id=MaS6XPk>
- [3] Juego - wikipedia, la enciclopedia libre. Visitado: 2018-04-10. [Online]. Available: <https://es.wikipedia.org/wiki/Juego>
- [4] Johan huizinga - wikipedia, la enciclopedia libre. Visitado: 2018-06-11. [Online]. Available: https://es.wikipedia.org/wiki/Johan_Huizinga
- [5] El despegue de la educación a distancia. Visitado: 2018-06-07. [Online]. Available: <https://aniversario.elpais.com/el-despegue-de-la-educacion-a-distancia/>
- [6] El 'e-learning', un modelo de formación que gana cada vez más adeptos. Visitado: 2018-06-07. [Online]. Available: <http://www.elmundo.es/extras/formacion-online/2017/09/26/59ca90a1ca4741900e8b4601.html>
- [7] A. Pisabarro, M. L. Fernández, and E. Manso, *Guía Docente de la asignatura Fundamentos de Programación*. Escuela de Ingeniería Informática de Valladolid, 2016/17.
- [8] Yincana - diccionario panhispánico de dudas. Visitado: 2018-02-20. [Online]. Available: <http://lema.rae.es/dpd/?key=yincana>
- [9] Torneo de los 011 magos - escuela de ingeniería informática de valladolid. Visitado: 2018-02-23. [Online]. Available: <https://www.inf.uva.es/2016/10/19/torneo-de-los-0011-magos/>
- [10] A. G. Mancha and R. Rodríguez-Echevarría, *Actas XXIII Jornadas sobre la Enseñanza Universitaria de la Informática*. Universidad de Extremadura, 2017. [Online]. Available: http://jenui2017.unex.es/actas_jenui2017.pdf
- [11] Encuesta sobre equipamiento y uso de tecnologías de la información y comunicación en los hogares. año 2017. instituto nacional de estadística. Visitado: 2018-06-12. [Online]. Available: http://www.ine.es/prensa/tich_2017.pdf
- [12] Videojuego - wikipedia, la enciclopedia libre. Visitado: 2018-06-18. [Online]. Available: <https://es.wikipedia.org/wiki/Videojuego>

- [13] Los títulos más jugados actualmente en pc. Visitado: 2018-06-20. [Online]. Available: <https://www.redbull.com/cl-es/los-t%C3%ADtulos-m%C3%A1s-jugados-actualmente-en-pc>
- [14] Counter-strike: Global offensive. Visitado: 2018-06-20. [Online]. Available: <http://blog.counter-strike.net/>
- [15] Playerunknown's battlegrounds. Visitado: 2018-06-20. [Online]. Available: <https://playbattlegrounds.com/main.pu>
- [16] Fortnite, de epic games. Visitado: 2018-06-20. [Online]. Available: <https://www.epicgames.com/fortnite/es-ES/home?lang=es-ES>
- [17] Rockstar games - grand theft auto online. Visitado: 2018-06-20. [Online]. Available: <https://www.rockstargames.com/GTAOnline/>
- [18] Sitio oficial | minecraft. Visitado: 2018-06-20. [Online]. Available: <https://minecraft.net/es-es/>
- [19] Hearthstone - página oficial. Visitado: 2018-06-20. [Online]. Available: <https://playhearthstone.com/es-es/>
- [20] Web oficial de league of legends. Visitado: 2018-06-20. [Online]. Available: https://play.euw.leagueoflegends.com/es_ES
- [21] Play with programming - codingame. Visitado: 2018-06-20. [Online]. Available: <https://www.codingame.com/home>
- [22] Codecombat - learn how to code by playing a game. Visitado: 2018-06-20. [Online]. Available: <https://codecombat.com/>
- [23] Code hunt - microsoft research. Visitado: 2018-06-20. [Online]. Available: <https://www.microsoft.com/en-us/research/project/code-hunt/>
- [24] 5 games that teach you how to code. Visitado: 2018-06-19. [Online]. Available: <https://www.webpagefx.com/blog/web-design/games-that-teach-how-to-code/>
- [25] Unity. Visitado: 2018-06-20. [Online]. Available: <https://unity3d.com/es>
- [26] blender.org - home of the blender project - free and open 3d creation software. Visitado: 2018-06-20. [Online]. Available: <https://www.blender.org/>
- [27] What is unreal engine 4. Visitado: 2018-06-20. [Online]. Available: <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>
- [28] Source engine features - valve developer community. Visitado: 2018-06-20. [Online]. Available: https://developer.valvesoftware.com/wiki/Source_Engine_Features
- [29] Gamemaker | yoyo games. Visitado: 2018-06-20. [Online]. Available: <https://www.yoyogames.com/gamemaker>
- [30] Cryengine | the complete solution for next generation game development by crytek. Visitado: 2018-06-20. [Online]. Available: <https://www.cryengine.com/>

- [31] libgdx. Visitado: 2018-06-20. [Online]. Available: <https://libgdx.badlogicgames.com/>
- [32] online text editor - collabedit. Visitado: 2018-06-20. [Online]. Available: <http://collabedit.com/>
- [33] Codeshare - share code in real-time with developers in your browser. Visitado: 2018-06-20. [Online]. Available: <https://codeshare.io/>
- [34] JsFiddle. Visitado: 2018-06-20. [Online]. Available: <https://jsfiddle.net/>
- [35] Scrapfy - shared and real-time code editor for ninjas. Visitado: 2018-06-20. [Online]. Available: <https://scrapfy.io/>
- [36] Aws cloud9 - amazon web services. Visitado: 2018-06-20. [Online]. Available: <https://scrapfy.io/>
- [37] Home | teamspeak. Visitado: 2018-06-20. [Online]. Available: <https://www.teamspeak.com/en/>
- [38] Discord - chat de texto y voz gratuito para gamers. Visitado: 2018-06-20. [Online]. Available: <https://discordapp.com/>
- [39] Skype | herramienta de comunicación para chats y llamadas gratis. Visitado: 2018-06-20. [Online]. Available: <https://www.skype.com/es/>
- [40] Unity - manual: Unity user manual. Visitado: 2018-06-21. [Online]. Available: <https://docs.unity3d.com/Manual/index.html>
- [41] Unity - scripting api. Visitado: 2018-06-21. [Online]. Available: <https://docs.unity3d.com/ScriptReference/index.html>
- [42] Assets, objects and serialization - unity. Visitado: 2018-06-21. [Online]. Available: <https://unity3d.com/es/learn/tutorials/topics/best-practices/assets-objects-and-serialization>
- [43] Scrum methodology. Visitado: 2018-06-22. [Online]. Available: <http://scrummethodology.com/>
- [44] Sueldos de desarrollador de software en españa | indeed.es. Visitado: 2018-06-22. [Online]. Available: <https://www.indeed.es/salaries/Desarrollador/a-de-software-Salaries?period=hourly>

