

Trufo's adventure

Autor: Omar Cester Gracia

Tutor: Gustau Marcos Ballester

Profesor: Joan Arnedo Moreno

Grado de Ingeniería Informática

Itinerario en Tecnologías de la información

14/01/2024

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento- NoComercial-SinObraDerivada [3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Trufo's adventure</i>
Nombre del autor:	<i>Omar Cester Gracia</i>
Nombre del colaborador/a docente:	<i>Gustau Marcos Ballester</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>01/2024</i>
Titulación o programa:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>75.640 - TFG - Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Videojuego, aprendizaje, exploración</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo</i>	
<p>El Trabajo de Fin de Grado propuesto tiene como objetivo la creación de un prototipo completamente funcional de un videojuego en 2D denominado 'Trufo's Adventure', desarrollado con el motor de videojuegos multiplataforma Unity, que permite el desarrollo tanto para PC como para Mac fácilmente. Este videojuego, enfocado a niños de siete a once años, fusiona componentes de exploración y recolección con características de gamificación para el aprendizaje de nuevas palabras en inglés a través del juego.</p> <p>El videojuego se centra en la tarea de Trufo de recolectar ingredientes para completar las recetas solicitadas por diversos NPC, que debe consultarlas el recetario disponible. Un aspecto importante es que todas las indicaciones del recetario se encuentran en inglés, proporcionando así una inmersión lingüística para el jugador.</p> <p>Se ha desarrollado siguiendo una metodología ágil debido al tiempo limitado para la implementación y la inexperiencia en el desarrollo de videojuegos. Por otro lado, debido a los escasos conocimientos en diseño gráfico, se han empleado <i>assets</i> y <i>sprites</i> de terceros para garantizar una representación visual efectiva.</p> <p>Github.</p> <p>Itch.io.</p> <p>Vídeo trailer.</p>	

Abstract (in English, 250 words or less):

The proposed Bachelor's Degree Dissertation aims to create a fully functional prototype of a 2D video game titled 'Trufo's Adventure,' developed using the cross-platform game engine Unity, allowing easy development for both PC and Mac. Targeted at children aged seven to eleven, this video game combines exploration and collection components with gamification features to facilitate the learning of new English words through gameplay.

The game revolves around Trufo's task of gathering ingredients to complete recipes requested by various NPCs, which must be consulted in the available recipe book. A significant aspect is that all instructions in the recipe book are in English, providing linguistic immersion for the player.

The development follows an agile methodology due to the limited time for implementation and the lack of experience in game development. Additionally, owing to limited graphic design knowledge, third-party assets and sprites have been used to ensure effective visual representation.

Resumen

El Trabajo de Fin de Grado propuesto tiene como objetivo la creación de un prototipo completamente funcional de un videojuego en 2D denominado 'Trufo's Adventure', desarrollado con el motor de videojuegos multiplataforma Unity, que permite el desarrollo tanto para PC como para Mac. Este videojuego, enfocado a niños de siete a once años, fusiona componentes de exploración y recolección con características de gamificación para el aprendizaje de nuevas palabras en inglés a través del juego.

El videojuego se centra en la tarea de Trufo de recolectar ingredientes para completar las recetas solicitadas por diversos NPC, que debe consultarlas el recetario disponible. Un aspecto importante es que todas las indicaciones del recetario se encuentran en inglés, proporcionando así una inmersión lingüística para el jugador.

Se ha desarrollado siguiendo una metodología ágil debido al tiempo limitado para la implementación y la inexperiencia en el desarrollo de videojuegos. Por otro lado, debido a los escasos conocimientos en diseño gráfico, se han empleado assets y sprites de terceros para garantizar una representación visual efectiva.

Palabras clave

Videojuego, aprendizaje, exploración.

Abstract

The proposed Bachelor's Degree Dissertation aims to create a fully functional prototype of a 2D video game titled 'Trufo's Adventure,' developed using the cross-platform game engine Unity, allowing development for both PC and Mac. Targeted at children aged seven to eleven, this video game combines exploration and collection components with gamification features to facilitate the learning of new English words through gameplay.

The game revolves around Trufo's task of gathering ingredients to complete recipes requested by various NPCs, which must be consulted in the available recipe book. A significant aspect is that all instructions in the recipe book are in English, providing linguistic immersion for the player.

The development follows an agile methodology due to the limited time for implementation and the lack of experience in game development. Additionally, owing to limited graphic design knowledge, third-party assets and sprites have been used to ensure effective visual representation.

Keywords

Video game, learning, exploration.

Índice

1. Introducción.....	11
1.1. Introducción/Prefacio.....	11
1.2. Descripción/Definición	12
1.3. Objetivos generales	16
1.3.1. Objetivos principales.....	16
1.3.2. Objetivos secundarios	16
1.4. Metodología y proceso de trabajo.....	17
1.5. Planificación.....	17
1.5.1. Planificación desglosada.....	18
1.5.2. Cronograma	19
1.6. Presupuesto	20
1.7. Estructura del resto del documento	22
2. Análisis de mercado	23
2.1. Desarrollos <i>Onedev</i>.....	23
2.2. Títulos <i>indie</i> en Unity	28
2.3. Consejos del sector	31
2.4. Inspiraciones.....	33
2.5. Público objetivo	35
3. Propuesta	36
3.1. Definición de especificaciones del producto.....	36
3.2. Historia y objetivo	37
3.3. Objetivos del gameplay	37
3.4. Plataforma publicación.....	40
3.5. Estrategia de Marketing.....	41
4. Diseño.....	42
4.1. Entorno de desarrollo	42
4.2. Herramientas utilizadas	43
4.3. Arquitectura general	44
4.4. Patrones de diseño	46

4.5. Estructura del proyecto	47
4.6. Listado de clases.....	48
4.7. Diagrama de flujo de escenas.....	54
4.8. Diseño gráfico e interfaces	55
4.8.1. Estilos.....	55
4.8.2. Diseño del nivel	57
4.8.3. Accesibilidad	59
4.9. Inventario de Assests y recursos.....	60
4.10. Evolución del diseño	64
4.11. Estructura de escenas.....	66
4.12. Feedback cuestionarios	66
5. Implementación	71
5.1. Requisitos de instalación	71
5.2. Instrucciones de instalación.....	71
6. Demostración.....	72
6.1. Instrucciones de uso.....	72
6.2. Guía de usuario	73
7. Conclusiones y líneas de futuro	77
7.1. Repercusiones Ético-Sociales de los videojuegos en el aula	77
7.2. Conclusiones	78
7.3. Líneas de futuro.....	79
Bibliografía	81
Anexos	84
Anexo A: Glosario	84
Anexo B: Entregables del proyecto	85
Anexo C: Resultados detallados de la encuesta.....	86

Figuras y tablas

Índice de figuras

Figura 1: The Legend of Zelda: A Link to the Past (1991)	15
Figura 2: Stardew Valley (2016)	15
Figura 3: Harvest Moon: Friends of Mineral Town (2003)	15
Figura 4: Diagrama de Gantt	19
Figura 5: Cave Story	24
Figura 6: Papers, please	25
Figura 7: Stardew Valley	26
Figura 8: Sea of Stars	28
Figura 9: Sea of Stars escenario	28
Figura 10: Sea of Stars interiores	28
Figura 11: Blasphemous	29
Figura 12: Blasphemous img. 1	29
Figura 13: Blasphemous img. 2	29
Figura 14: Wizard of Legend	30
Figura 15: Wizard of Legend img. 1	30
Figura 16: Wizard of Legend img. 2	30
Figura 17: Harvest Moon: Friends of Mineral Town	33
Figura 18: Harvest Moon tools	34
Figura 19: Harvest Moon interacción con personajes	34
Figura 20: Stardew Valley	34
Figura 21: Stardew Valley interacción con personajes	35
Figura 22: Stardew Valley tools	35
Figura 23: Trufo por Cupnoodle	37
Figura 24: Requisitos técnicos Unity.	44
Figura 25: Estructura organizativa Assets	47
Figura 26: Diagrama visual de las clases	48
Figura 27: EventController esquema implementación.	53
Figura 28: Diagrama de flujo de escenas	54
Figura 29: Logotipo	55
Figura 30: Paleta de colores	55
Figura 31: Fuente título menú inicial	56
Figura 32: Fuente utilizada en UI, botones y diálogos	56
Figura 33: Assets UI	56
Figura 34: Vista general nivel	57
Figura 35: Imagen puentes	57
Figura 36: Barco navegación	58
Figura 37: Barra de acción	58
Figura 38: Animation Event que controla el movimiento.	58

Figura 39: Activación Collider. Izquierda Dig, Derecha Cut.	58
Figura 40: Controles de accesibilidad.	59
Figura 41: Modos visualización: Protanopia, Deuteranopia y Tritanopia.	59
Figura 42: Modificación con Piskel y Gimp del color, tamaño y flecha punteros	65
Figura 43: Edición NPC	65
Figura 44: Estructura jerárquica StartScene	66
Figura 45: Estructura jerárquica Level1	66
Figura 46: Evidencias de uso de la demo del prototipo	70
Figura 47: Menú inicio (izquierda)	73
Figura 48: Menú Guía (derecha)	73
Figura 49: NPC inicio de misión.	73
Figura 50: Diálogo y detalles de la receta.	74
Figura 51: Misión finalizada.	74
Figura 52: Selección de herramienta correcta vs. incorrecta	75
Figura 53: Herramienta para talar	75
Figura 54: Función de la herramienta Point.	76
Figura 55: Pantalla fin de partida.	76

Índice de tablas

Tabla 1: Mecánicas propuestas	15
Tabla 2: Partida presupuestaria	21
Tabla 3: Acciones gameplay	38
Tabla 4: Ingredientes gameplay	39
Tabla 5: NPC gameplay	40
Tabla 6: Herramientas utilizadas.	43
Tabla 7: Sprout Lands UI recursos gráficos	60
Tabla 8: Sprout Land Asset Pack recursos gráficos	62
Tabla 9: NPC recursos gráficos	62
Tabla 10: Freesound recursos audio	63
Tabla 11: Fuentes de texto	63

1.Introducción

1.1. Introducción/Prefacio

La realización de un trabajo de fin de grado (TFG) en el área de videojuegos no es solo una elección académica, ha sido una pasión que ha crecido conmigo durante mi juventud y este proyecto de fin de grado además de un requisito educativo del Grado, es la primera piedra hacia una meta que ha sido el motor principal de mi matriculación en el Grado de Ingeniería Informática.

Otro aspecto motivador que ha hecho decantarme por esta temática es la convicción de que los videojuegos poseen potencial educativo y creo firmemente que más allá de ser simples formas de entretenimiento, pueden estimular el pensamiento crítico, fomentar la resolución de problemas, la observación, la asociación y ofrecer experiencias educativas innovadoras que pueden inspirar y enseñar a los jugadores de una manera única y atractiva.

Aunque la pasión por los videojuegos es una constante en mi vida, carezco de experiencia práctica en el desarrollo, más allá de la visualización de numerosos tutoriales de motores como Unity [1] y Unreal [2] (entre otros) para comprender las capacidades que ofrecen a la hora de desarrollar desde una perspectiva *indie*. Mediante este proyecto pretendo transformar ese conocimiento teórico y lo aprendido a lo largo del Grado en un videojuego.

En definitiva, con la realización de este TFG se pretendo crear un prototipo de videojuego que represente la fusión de mi formación académica y mi pasión por los videojuegos, con la aspiración de contribuir de una forma modesta al campo de la educación a través de la creatividad y la tecnología.

1.2. Descripción/Definición

Este Trabajo de Fin de Grado surge de la convergencia de dos elementos: la formación recibida en el Grado de Ingeniería Informática y el interés por el desarrollo de videojuegos. La idea central es la creación de un prototipo de videojuego que no solo sirva como ejercicio práctico para aplicar las herramientas y habilidades técnicas, sino que aspire a ofrecer una experiencia educativa básica atractiva.

Este proyecto tiene relevancia en el contexto actual, donde concurren el uso la tecnología y la educación, las cuales se presentan como una tendencia en constante evolución y adopción en distintos ámbitos educativos como herramientas pedagógicas. Con este proyecto se pretende contribuir a la innovación educativa a través del potencial formativo de los videojuegos, explorando los principios de la gamificación.

Para conseguir este objetivo de aprendizaje del idioma inglés, el videojuego se ha concebido como una herramienta para el docente de la asignatura de inglés, apoyándose en las metodologías activas de Aprendizaje Significativo de Ausubel y el Aprendizaje por Descubrimiento de Bruner que tal y como consideran los autores Moreno y Franklin: “las técnicas, el sistema de evaluación y por supuesto los medios más eficaces que nos van a permitir optimizar nuestro tiempo y labor educativa, haciendo un uso acertado de la tecnología en el ámbito educativo.” [3]. Esta herramienta en forma de videojuego se desarrolla para trabajar con ella en talleres y laboratorios propuestos por el docente bajo su criterio, para que el alumno pueda aprender nuevo vocabulario a través de la búsqueda de ingredientes en inglés, consiguiendo un refuerzo de lo impartido en el aula.

Este videojuego se diseña para un público objetivo de niños entre siete y once años, pudiendo variar según el criterio del docente. Este periodo coincide con la tercera etapa del desarrollo cognitivo desarrollado por Jean Piaget denominada “operaciones concretas”; en esta etapa el niño aumenta el grado de alojamiento de información, además de ser capaces de pensar de manera más lógica y abstracta. El lenguaje y las habilidades de los niños se acelera de forma drástica [4].

El producto que se pretende obtener es un prototipo básico de un nivel, donde el jugador pueda comprobar la jugabilidad y explorar la isla a través de diferentes *quests*. Los NPC distribuidos por el nivel solicitan al jugador la recolección de ingredientes para completar ciertas recetas. Estas se hallan en un recetario en inglés que ha encontrado el jugador al inicio del nivel, de manera que el usuario tiene que localizar visualmente los ingredientes solicitados; esta localización será de forma rápida si conoce la palabra, o tardará un poco más consolidando nuevas palabras en inglés en el caso de desconocer el término de acuerdo a las metodologías de aprendizaje anteriormente citadas.

Se ha pretendido buscar una mecánica divertida y sólida para un juego de recolección y exploración del entorno del nivel, para ello la idea concebida se ha inspirado en juegos ya consolidados y reconocidos que basan parte de su mecánica en la recolección de *ítems*, como la saga Atelier [5], prescindiendo de combates y entorno 3D, Harvest Moon [6] y el icónico Stardew Valley [7] con una estética Top-Down similar a la buscada para este prototipo de nivel, pero con mecánicas y desarrollo sustancialmente simplificados debido al tiempo de desarrollo estimado.

El videojuego comienza con su protagonista Trufo en una isla, donde un NPC necesita cocinar una receta y le pide ayuda, pero el NPC no sabe qué ingredientes le faltan para la receta, así que solicita a Trufo que encuentre un recetario perdido. Una vez Trufo encuentra el libro de recetas y se lo entrega al NPC, este le solicita que busque los ingredientes necesarios de la receta, ya que están en inglés y no lo entiende. A partir de este momento comienza la aventura de exploración de Trufo por la isla donde conoce a distintos personajes que necesitan diversas recetas.

La mecánica principal del protagonista, Trufo, es explorar el mapa y recoger estos ingredientes, y para ello el jugador deberá seleccionar la herramienta de recolección adecuada que le permitirá realizar una de estas tres acciones: coger, cortar o cavar. Cada ingrediente u objeto solicitado solo puede ser recolectado con la herramienta adecuada para ello. Una vez seleccionada, el jugador se tiene que posicionar cerca del ingrediente u objeto solicitado por el NPC y ejecutar la acción en uso con el botón izquierdo del ratón.

Asimismo, el jugador puede hablar con los distintos NPC a través de la herramienta 'hablar', que debe estar activa, y así poder iniciar la *quest* correspondiente; las *quests* se activan tras finalizar la *quest* previa de forma automática. El NPC de inicio de cada *quest* está indicado con un símbolo de admiración "!" encima del NPC para que el jugador lo detecte visualmente; una vez terminada la *quest*, se activa la siguiente y el jugador debe encontrar el NPC con el símbolo "!" y se procede a activar la nueva *quest*.

Por otro lado, el cursor es modificado con el icono de la herramienta que esté seleccionada/activa en cada momento (coger, cavar, cortar), y para ayudar al jugador a mejorar la experiencia, y poder identificar qué ingrediente es necesario recolectar, al pasar el cursor por encima de los objetos que no son recolectables en la *quest* activa, se cambian los colores del cursor a gris. Del mismo modo, si el objeto es recolectable, pero no se ha seleccionado la herramienta adecuada, el cursor cambia a escala de grises. Si, por el contrario, el ingrediente es recolectable por la herramienta en uso, el icono del cursor mantiene el color indicando que el objeto que se apunta con el cursor es el correcto y es recolectable, y se permite ejecutar la acción y obtener el ingrediente.

El jugador puede llevar un control de los ingredientes de cada receta activa, las cantidades totales necesarias de cada ingrediente y los que ya se han recolectado en cada momento; todo ello a través de un panel lateral que se mostrará al inicio de cada *quest* con toda esta información. Una vez se hayan cumplido los requisitos de la *quest*, se informará al jugador que ya se han recolectado los objetos necesarios y puede entregarlos al NPC, que tendrá el icono de admiración "!" activado para que sea fácilmente reconocible por el jugador.

Asimismo, se está valorando introducir como mecánica secundaria el desbloqueo de obstáculos para poder acceder a ciertos ingredientes en zonas inicialmente bloqueadas. Es una idea que se está evaluando cuidadosamente para ampliar la jugabilidad más allá de la simple recolección de ingredientes, siempre y cuando se pueda garantizar la viabilidad dentro del alcance del proyecto.

Resumen mecánicas:

	Mecánicas Principales	Mecánicas Secundarias
Trufo	Explorar. Realizar acciones de recolección.	Quitar obstáculos para acceder a ciertas zonas.
NPC	Activar <i>Quest</i> . Información <i>Quest</i> . Finalizar <i>Quest</i> .	
Ingrediente	Modificar avance de la <i>Quest</i> .	
Receta	Indicar ingredientes a recolectar. Indicar cantidades necesarias.	

Tabla 1: Mecánicas propuestas

La plataforma elegida para la implementación del videojuego será para ecosistemas de escritorio PC/Mac, permitiendo tanto a jugadores de sistemas operativos Windows o macOS pueda disfrutar del prototipo final, puesto que depende del centro educativo el hardware que disponen en las aulas puede variar.

Finalmente, la estética de este prototipo será 2D con vista Top-Down inspirada en diferentes juegos como, por ejemplo:



Figura 1: The Legend of Zelda: A Link to the Past (1991)



Figura 2: Stardew Valley (2016)



Figura 3: Harvest Moon: Friends of Mineral Town (2003)

1.3. Objetivos generales

1.3.1. Objetivos principales

- Proporcionar un prototipo inicial del videojuego completamente funcional.
- Adquirir un conocimiento más profundo del motor Unity [8].
- Plasmar una correcta estructura de datos y lógica del juego aplicando contenidos aprendidos en el Grado.
- Realizar una correcta planificación ajustada a la envergadura del proyecto.

1.3.2. Objetivos secundarios

- Adquirir competencias para la creación de juegos en entorno 2D.
- Desarrollar habilidades en el manejo de herramientas de trabajo para entornos 2D.

1.4. Metodología y proceso de trabajo

Debido a la inexperiencia en el desarrollo de videojuegos y con el motor de videojuegos Unity, sumado al desafío de contar únicamente con un plazo de cuatro meses para la realización de este prototipo completamente funcional, la elección de la metodología ágil parece la opción más adecuada.

La flexibilidad de las prácticas ágiles puede ser un recurso necesario ante este escenario que pueda permitir adaptaciones continuas a medida que se avanza en el desarrollo; facilita la corrección de posibles desviaciones y ajustes en tiempo real, optimizando así los recursos limitados de tiempo.

Finalmente, una comunicación y colaboración regular con el tutor serán esenciales para adaptar en todo momento las desviaciones atendiendo a su experiencia e indicaciones, y así asegurar el cumplimiento de los plazos del proyecto dentro del cronograma establecido.

1.5. Planificación

Con el propósito de elaborar una planificación detallada del proyecto de Trabajo de Fin de Grado, se ha desarrollado un cronograma estructurado que abarca cuatro fases clave. La primera fase, denominada "PEC1 - Plan del Proyecto", se centra en la definición clara de los objetivos y alcance del proyecto. A continuación, la fase "PEC2 - Estado del Arte y Primera Versión" implica la creación de pruebas iniciales y la iteración sobre las mecánicas básicas del videojuego. La tercera fase, "PEC3 - Implementación de Versión Jugable", se enfoca en la creación del contenido jugable, desde el menú de inicio hasta la implementación de *quests* y detalles gráficos finales. Finalmente, la fase "PEC4 - Memoria y Productos Finales" comprende la revisión final, la generación de productos finales, y la documentación completa de la memoria del proyecto y entregables.

1.5.1. Planificación desglosada

PEC1 - Plan del Proyecto (27/9/23 al 8/10/23)

- Definición del Proyecto

PEC2 - Estado del Arte y Primera Versión (9/10/23 al 12/11/23)

- Crear escena de pruebas iniciales
- Pruebas iniciales de mecánicas principales
- Pruebas iniciales cambio de cursor
- Pruebas iniciales herramientas UI
- Segunda iteración revisión posibles mejoras
- Pruebas iniciales diseño NPC
- Pruebas iniciales sistema de *quests*
- Tercera iteración revisión posibles mejoras
- Documentar memoria

PEC3 - Implementación de Versión Jugable (13/11/23 al 17/12/23)

- Crear menú de inicio de juego
- Implementar diseño nivel jugable
- Implementar ingredientes recolectables
- Implementar recetas
- Implementar *quests*
- Pruebas de jugabilidad
- Implementar audios
- Implementar detalles gráficos finales
- Documentar memoria

PEC4 - Memoria y Productos Finales (18/12/23 al 14/1/23)

- Revisión final de errores
- Generar ejecutable del producto final
- Documentación final de la memoria
- Generar tráiler producto final
- Generar vídeo defensa producto final

1.5.2. Cronograma

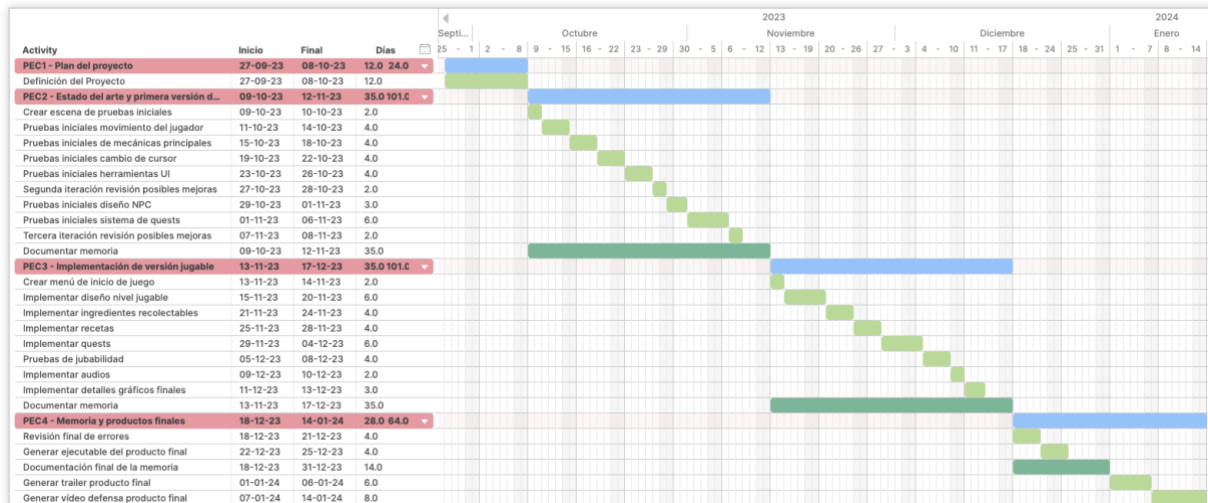


Figura 4: Diagrama de Gantt

Para ver en detalle la duración de cada tarea, consultar la siguiente URL:

<https://plan.tomsplanner.es/public/ocester>

1.6. Presupuesto

En el contexto de este Trabajo de Fin de Grado (TFG), que involucra el desarrollo de un prototipo de videojuego completamente funcional. Cada partida presupuestaria se ha calculado considerando las necesidades específicas del proyecto, desde herramientas y software, hasta equipo humano. Se presenta el siguiente presupuesto provisional detallado:

Partida	Descripción	Coste
Software		
Gimp	Herramienta para edición de imágenes y diseño.	-
Rider	Entorno de desarrollo C# en Unity (licencia estudiante).	-
Unity 2022.3.8f1	Plataforma de desarrollo de videojuegos (licencia personal).	-
Microsoft Office	Suite de oficina para documentación y planificación, (licencia estudiante).	-
Tom's Planner	Herramienta de planificación y gestión de proyectos.	-
PiskelApp	Editor de <i>sprites</i> en línea. Link	-
Github	Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git [9].	-
Github Desktop	Aplicación gratuita de código abierto para trabajar con archivos hospedados en GitHub a través de interfaz gráfica [10].	-
Hardware		
Mac mini 2018	3,2 GHz Intel Core i7 de 6 núcleos, RAM 8 GB 2667 MHz DDR4, Intel UHD Graphics 630 1536 MB.	1.160,93 €
Assests		
Sprout Lands UI	Pack de recursos para la UI de Cup Noodle (versión gratuita). Link	-
Sprout Lands Asset Pack	Pack de Assests para la creación del personaje, ingredientes, tilesets, pointers y animaciones de Cup Noodle. Link	3,99 €
Pixel Adventurer - Free Animated	Pack de personaje animado creado por Florassence. Link	-

Sprite Character		
Simple Top Down Character	Pack de personaje animado creado por BDragon1727. Link	-
Pixel Cute Bear 1	Pack de personaje animado creado por Immunity. Link	-
Cute Mushroom Character Sprite	Pack de personaje animado creado por Caz Creates Games. Link	-
Fantasy RPG townfolk pack	Para de personajes animados creado por Franuka. Link	-
Colorblind Effect	Colorblind Effect es un efecto de cámara de pantalla completa para Unity para adaptar la renderización a jugadores con Discromatropia. Link	-
Sonidos		
Freesound	Web con sonidos de licencia gratuita. Link	-
	Sad Piano.mp3	-
	Happy 8-bit music	-
	Retro Hit Sound	-
	Retro, Coin 04.wav	-
	Digging, Ice, Hammer, A.wav	-
	Huh.amp3	-
	Text-Message or Videogame-Jump	-
	Cheer	-
Fuentes de Texto		
Rainy Hearts	Utilizada para la estética textual del juego. Link	-
Huhg love	Fuente Utilizada para el título del juego en la pantalla principal. Link	-
Equipo humano		
Horas desarrollo	Se han calculado 110 días de trabajo, a una media de 4h. al día, en total 440 h. de desarrollo en una primera aproximación con un coste de 22 €/hora.	9.680 €
Coste Total		10.844,92 €

Tabla 2: Partida presupuestaria

1.7. Estructura del resto del documento

Los siguientes capítulos de la memoria constan de las siguientes secciones:

Capítulo 2 - Estado del arte y análisis de mercado: Se aborda el contexto actual y las tendencias relevantes en la industria, proporcionando una visión global de la posición del proyecto en relación con el mercado.

Capítulo 3 - Propuesta de proyecto y estrategia de comercialización: Se presenta de manera detallada la propuesta de proyecto en el sector, y la estrategia elaborada para la comercialización de este.

Capítulo 4 - Diseño: Se abordan todos los aspectos técnicos del trabajo (Arquitectura, UI, *scripts*, *Assets*...) tal como se han desarrollado durante las semanas.

Capítulo 5 - Implementación: Se abordan los requisitos de instalación e instrucciones de instalación.

Capítulo 6 - Demostración: Esta sección detalla los prototipos generados durante el desarrollo, ofreciendo una descripción de las pruebas y test realizados para evaluar la funcionalidad del proyecto. Además, proporciona una guía de usuario del videojuego.

Capítulo 7 – Conclusiones y líneas de futuro: Resumen y reflexión sobre el proyecto, evaluando la consecución objetivos planteados y analizando la planificación y metodología empleadas. También se ofrece información, predicciones y sugerencias para posibles expansiones futuras del proyecto, y mejoras a considerar.

2. Análisis de mercado

En esta sección se aborda el contexto actual y las tendencias relevantes en la industria, proporcionando una visión global de la posición del proyecto en relación con el mercado en la industria de desarrollo de videojuegos *onedev*, problemas encontrados en el sector y obtener una visión de éxitos de desarrollos previos y decisiones tomadas a tener en cuenta para el desarrollo del proyecto del TFG que nos ocupa y mejores prácticas del sector.

2.1. Desarrollos *Onedev*

Durante la búsqueda e investigación de proyectos creados por un solo desarrollador, se ha evidenciado que la mayoría de los juegos independientes son desarrollados por pequeños equipos de creadores debido al gran trabajo multidisciplinar que conlleva el desarrollo de un videojuego, sin embargo, se han encontrado videojuegos en los que una sola persona ha desarrollado todo sin ayuda de terceros que, si bien no es una tarea fácil ya que el tiempo de desarrollo para crear algo competitivo es de varios años, con alguna excepción cuyo desarrollo es más breve y que se verá a lo largo del desarrollo de este análisis, queda evidenciado que es posible.

Tras la investigación hay ciertos puntos en común en los videojuegos creados por un solo desarrollador:

- Los autores se inspiran en juegos de su infancia.
- El videojuego nace de una necesidad personal de desarrollo profesional.
- La inspiración surge a partir de otros videojuegos e intentan mejorarlos y resolver los problemas que ellos consideran mejorables.
- Largos tiempos de desarrollo y sacrificio debido al carácter multidisciplinar de este tipo de producto.
- Soledad del desarrollador que hace replantearse su continuidad.
- Retroalimentación a través de las RR.SS. durante el desarrollo del videojuego para pulir ciertos aspectos demandados.

A continuación, se detallan videojuegos creados por un solo desarrollador.

Cave Story



Figura 5: Cave Story

Videojuego desarrollado por Daisuke Amaya, conocido como Pixel, lanzado en 2004 estuvo en desarrollo durante más de 5 años. Ha influido notablemente en la comunidad de desarrolladores independientes. Aunque inicialmente se ofreció de forma gratuita, la popularidad creció tras su traducción al inglés y se lanzó comercialmente en varias plataformas: Wii, Steam, DSi Nintendo DSi y 3DS Nintendo 3DS, Mac, Linux, AmigaOS, PSP, GP2X, Wiz GP2X Wiz, Boy Advance Game Boy Advance, Xbox y DS Nintendo DS [11].

A pesar de sus comienzos como estudiante universitario y trabajar en software no relacionado con videojuegos, Daisuke sigue siendo denominado como un "True Indie" que crea por pasión dejando su trabajo para mejorar y expandir Cave Story, siete años después de su lanzamiento, con nuevas versiones. La obra de Daisuke Amaya es caracterizada como "entrañable", amigable, con enemigos y personajes no demasiado difíciles, mostrando en todo momento un toque humano. Daisuke se inspiró en juegos como Metroid, Mega Man y especialmente Wonder Boy [12].

Papers, Please



Figura 6: Papers, please

Videojuego financiado y creado por completo por Lucas Pope, cuyo desarrollo comenzó en noviembre de 2012 y tardó nueve meses en desarrollarse. El autor compartió sus actualizaciones de desarrollo en TIGSource [13] para recibir retroalimentación que incorporó al juego. Tras ser aprobado en Steam Greenlight en abril de 2013, Pope finalizó "Papers, Please". Para asignar los nombres de los personajes del juego, Pope difundió un formulario público, donde personas podía poner sus nombres que posteriormente se asignaron de forma aleatoria a los inmigrantes del juego, ya que el juego se basa en un agente de aduanas que filtra las personas que pueden o no entrar al país.

El objetivo del videojuego según Pope en la entrevista de ABC News es "conectar más a los jugadores con las difíciles decisiones que un inspector de inmigración debe tomar. El enfoque se centra en la tarea de nivel inferior de tratar de manejar un trabajo y una familia cuando lo correcto no está tan claro." [14]. Este enfoque se distancia de la corriente principal de los videojuegos y va un paso más allá, aportando nuevas ideas narrativas y creativas para la industria del videojuego.

Lanzado en agosto de 2013 recibió críticas positivas, aunque Pope expresó cansancio del juego en 2014, deseando pasar a nuevos proyectos y distanciarse de solicitudes repetidas de secuelas o expansiones de DLC. Fue desarrollado inicialmente en HaxeNME[15], en posteriores lanzamientos se ha utilizado el motor de desarrollo de videojuegos Unity tanto para la versión de smartphones [16] como en un futuro la reedición en Windows, MacOs y Linux [17]. Antes de su lanzamiento en iPad y PlayStation Vita, Pope destacó que para plataformas que dependen de controles táctiles manuales, se presentaban diversos desafíos vinculados a la interfaz de usuario del juego que necesitaban actualizarse [18].

El blog de este autor es una fuente de inspiración e información donde se puede tener una fotografía paso a paso del desarrollo de un videojuego independiente, desde el inicio, idea principal, decisiones tomadas, bocetos, modificaciones, desarrollo técnico, etc. hasta su lanzamiento.

Stardew Valley



Figura 7: Stardew Valley

Videojuego desarrollado por un solo creador, Eric Barone, y lanzado en 2016. De acuerdo a la entrevista en el Cornell Daily Sun [19], Barone, inspirado por la insatisfacción con los títulos recientes de Harvest Moon, buscó crear el juego que siempre quiso jugar. Aunque inicialmente tenía experiencia en juegos más pequeños, su habilidad mejoró durante los cuatro años de desarrollo.

Barone, quien comenzó este proyecto a los 24 años, asumió cada aspecto de la creación del juego, incluyendo arte, gráficos y código, lo que resultó en una estética retro. Expresa una conexión con una época pasada en los videojuegos que encuentra mágica y creativa, reconociendo una sensación de estar atrapado en ese pasado imaginativo debido a dedicar 70 horas a la semana a su creación.

En la entrevista a Game Developer [20], reconoce trabajar horas extensas, inicialmente diez horas al día durante el desarrollo y luego aumentando a alrededor de quince horas al día después del lanzamiento del juego. Comenzó este proyecto a los 24 años, asumió cada aspecto de la creación del juego, incluyendo arte, gráficos y código, lo que resultó en una estética retro. Expresa una conexión con una época pasada en los videojuegos, reconociendo una sensación de estar atrapado en ese pasado imaginativo debido a dedicar 70 horas a la semana a su creación.

A pesar de ser un período intenso autoimpuesto, reflexiona sobre el impacto de Harvest Moon en su vida y tiene como objetivo crear un juego con un poder emocional similar, asegurándose de que siga siendo divertido. Barone se propuso mejorar las deficiencias percibidas en Harvest Moon, introduciendo elementos como la creación y misiones para una experiencia de juego más significativa. Stardew Valley ofrece una experiencia diversa con creación, mazmorras, minas y la caza de monstruos más allá de la agricultura.

La espera para su lanzamiento, inicialmente sin fecha, resultó valer la pena al resolver el problema observado en la serie Harvest Moon que se había quedado completamente estancada, y ofrecer más posibilidades de personalización y de espacio [19]. El juego se siguió desarrollando incluso después del lanzamiento, centrándose en nuevas características y la implementación del modo multijugador cooperativo.

Barone, bajo el alias ConcernedApe, mantuvo su identidad en gran parte en secreto durante el desarrollo. Su enfoque de la comunidad se define como auténtico, actuando en línea como lo haría en la vida real, y le apasiona la libertad de implementar todas las facetas del juego, además, destaca la inspiración de Barone en Miyazaki, su amor por los juegos inmersivos y su deseo de ofrecer algo nuevo [19].

En la entrevista para PCGamer [21], Barone indica que inicialmente pensaba lanzar algo relativamente pequeño para Xbox Live Indie Games, pero poco a poco continuó desarrollando el videojuego y creciendo, por lo que terminó por desechar esta idea. Barone anunció el juego en septiembre de 2012, utilizando el sistema Greenlight de Steam para atraer a potenciales jugadores, y la comunidad demostró un considerable interés en el juego, lo que llevó a Barone a dedicarse por completo al proyecto. Mantuvo una comunicación activa con la comunidad a través de Reddit y Twitter, compartiendo regularmente actualizaciones sobre el progreso del juego y buscando retroalimentación y sugerencias.

Finalmente, el lanzamiento se realizó para las plataformas PC, PS4, y Xbox One en 2016, en Nintendo Switch en 2017 y PS Vita en 2018. El desarrollo del videojuego se programó en C# utilizando el *framework* Microsoft XNA, y se utilizó Paint.NET para la creación de gráficos y Propellerhead Software's Reason para la composición musical y efectos de sonido [19].

2.2. Títulos *indie* en Unity

Para el desarrollo del prototipo, se ha buscado inspiración en juegos de estética **retro y pixel art** específicamente aquellos desarrollados en Unity Engine. A continuación, se enumeran algunos ejemplos de videojuegos que han sido concebidos en Unity Engine y que han influido en la elección de este motor para la creación del prototipo.

Sea of Stars



Figura 8: Sea of Stars

Lanzado el 29 de agosto de 2023, Sea of Stars toma inspiración de clásicos JRPG japoneses como Chrono Trigger y los primeros juegos de Final Fantasy, adoptando un estilo de arte pixelado de la era de Super Nintendo. Escenarios llenos de detalles donde esos mapas tan elaborados contribuyen al atractivo del juego fomentando la exploración y la curiosidad. Exhibe de una banda sonora melódica que realza la experiencia de juego. La atención al detalle, desde variaciones en la música de batalla hasta melodías específicas de áreas, añade encanto al juego. The Guardian lo define como “como un clásico moderno con profundidad y destello” [22]. Lanzado en plataformas Windows, Nintendo Switch, PlayStation 4, PlayStation 5, Xbox One, Xbox Series X/S.



Figura 9: Sea of Stars escenario



Figura 10: Sea of Stars interiores

Blasphemous



Figura 11: Blasphemous

Videojuego Español lanzado en 2019 con un éxito más que notable, inspirado en el folclore español llevado al mundo de la fantasía. Aunque el juego sigue la estructura típica de un *hack and slash* con elementos Metroidvania, se destaca por sus controles, una cautivadora ambientación y un estilo artístico muy personal. Tal y como lo define la revista mercurio es un “culto al **barroco pixelado**” [23]. Este título desarrollado con **Unity Engine** compite con los grandes del sector, y Enrique Colinet, uno de los diseñadores lo tiene claro: “Lo que ha vendido el juego ha sido su aspecto visual: el purismo extremo con el que hemos concebido el *pixel art*” [23]. El juego se lanzó para Microsoft Windows, PlayStation 4, Xbox One y Nintendo Switch el 10 de septiembre de 2019, con Warp Digital a cargo de las adaptaciones para consola. Las versiones para macOS y Linux se lanzaron el 21 de septiembre de 2020. Una versión para Amazon Luna fue lanzada el 20 de octubre de 2020 [24].



Figura 12: Blasphemous img. 1



Figura 13: Blasphemous img. 2

Wizard of Legend



Figura 14: Wizard of Legend

Lanzado en 2018, el concepto Wizard of legend se originó con el deseo de combinar el descubrimiento de objetos y la exploración de los clásicos juegos de acción y aventuras de SNES con el atractivo ritmo de combate de los juegos modernos de *hack and slash* y lucha, con opción de cooperación local para dos jugadores [25]. La estética se caracteriza por su **arte pixelado** y su estilo **retro** donde utiliza una perspectiva en 2D con elementos de pixel art, lo que le da al juego un aspecto retro. Los personajes, hechizos y entornos están diseñados con detalles pixelados. A pesar de su estilo simplificado, la estética logra transmitir la acción dinámica de los combates mágicos en el juego. Dado su enfoque procedural, algunas críticas han surgido respecto a la limitada diversidad en los mapas en 2D, lo que ha llevado a una percepción de depreciación del *pixel art* repetitivo [26]. Wizard of Legend fue lanzado simultáneamente en Microsoft Windows, OS X, Linux, PlayStation 4, Nintendo Switch y Xbox One el 15 de mayo de 2018.



Figura 15: Wizard of Legend img. 1



Figura 16: Wizard of Legend img. 2

2.3. Consejos del sector

Durante esta recopilación de consejos, se ha comparado la información compartida por Eric Johnson de Infinite Monkeys Entertainment, en su charla de 2018 para nuevos desarrolladores de juegos independientes [27] utilizando su experiencia personal de desarrollador con Life Goes On como ejemplo, y por otro lado Rogual, quien reflexiona sobre los desafíos enfrentados durante el desarrollo del videojuego Blackshift [28][29].

En el estudio de estas dos perspectivas, las **dificultades en el desarrollo individual** para Rogual fue un punto clave. Se enfrentó a desafíos relacionados con el desarrollo en solitario, insistió en manejar todos los aspectos del juego (arte, programación, música), lo que llevó a problemas como gráficos poco impresionantes, sugiere subcontratar a un artista que mejore estos aspectos.

Ambos desarrolladores destacan desafíos en **marketing** y obtener visibilidad para sus juegos. Rogual menciona esfuerzos limitados de marketing, confiando en Woovit para claves de transmisión con resultados poco significativos. Eric Johnson, sugiere la importancia de conocer el mercado para obtener mejores ventas, se evidencia una conciencia de los desafíos en el marketing de juegos indie, destacando su impacto en el éxito del juego.

Por otro lado, Jhonson menciona varias **herramientas** y recursos para estudiar el mercado de Steam, como por ejemplo Steam Spy, IsThereAnyDeal, Steam Charts y Steam database, que pueden ser útiles para revisar estadísticas sobre las ventas, reseñas, descuentos y otros aspectos del mercado de juegos en Steam.

Johnson también destaca que la cantidad de **reseñas** no siempre se correlaciona con la calidad percibida del juego. Hay juegos con pocas revisiones que son de alta calidad y juegos con muchas revisiones que son de baja calidad y viceversa. También hay juegos con pocas reseñas, pero un alto porcentaje que son positivas. Esto muestra que la calidad no siempre se traduce directamente en más reseñas.

Otro hallazgo interesante que expone Jhonson es la variabilidad en la relación entre el porcentaje de reseñas positivas y las ventas. En los gráficos mostrados en la conferencia, se observa que algunos géneros tienen una relación bastante directa entre el porcentaje de reseñas positivas y las ventas, mientras que otros tienen una relación menor.

Buscar **nichos viables** en el mercado es otro consejo compartido. Rogual describe Blackshift como un juego de nicho con rompecabezas difíciles que atraen a perfiles de ingeniería. Eric Johnson, aunque no menciona explícitamente el nicho, enfatiza la importancia de conocer el mercado, recomienda dirigirse a nichos específicos en lugar de seguir tendencias populares.

La **duración** de la jugabilidad se expone como otro aspecto crucial. Johnson destaca el tiempo de juego promedio y su capacidad de **re jugabilidad** como determinantes del éxito comercial, juegos con contenido duradero suelen atraer más ventas.

Rogual expone la implementación de un editor de niveles en línea para Blackshift, que resultó ser una **asignación errónea de esfuerzos** ya que a la mayoría de los jugadores no les interesaba. La **participación de la comunidad** se presenta como un consejo unánime, Eric Johnson enfatiza la importancia de esta participación, que podría incluir características como el soporte para modificaciones, tablas de clasificación, etc.

La **diferenciación visual** aparece como otro punto clave. Johnson resalta la importancia de la estética visual para influir en las decisiones de compra que, aunque reconoce que no es necesario contar con gráficos de última generación, subraya la necesidad de que el juego se distinga de forma visual de otros.

Ambas experiencias exponen reflexiones sobre las decisiones tomadas durante el desarrollo. Rogual reflexiona sobre la decisión de manejar todos los aspectos del juego limita el alcance del proyecto, y antes de implementar nuevas características es mejor pulir las existentes. Eric Johnson, reflexiona sobre la comprensión del mercado, la creación de juegos que se vendan y el aprendizaje de experiencias pasadas, sugiriendo un **proceso continuo de mejora y adaptación al mercado**.

2.4. Inspiraciones

A continuación, se detallan juegos que han inspirado las mecánicas de juego, estética y jugabilidad de Trufo's adventure.

Harvest Moon: Friends of Mineral Town

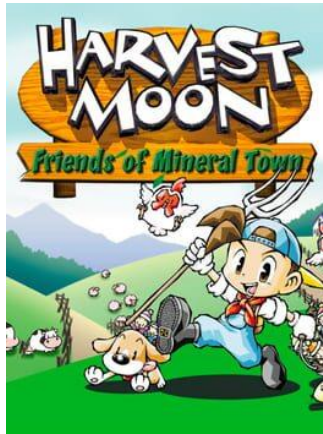


Figura 17: Harvest Moon: Friends of Mineral Town

Harvest Moon: Friends of Mineral Town [30] [31], un juego lanzado en 2003 que sigue la fórmula clásica de la serie de simulación. Se sume el papel de un chico encargado de revitalizar una granja en mal estado después de la muerte de un viejo amigo. La meta es la cría de animales, el cultivo de alimentos en diferentes épocas del año, con una base muy importante de exploración e interacción con los distintos personajes del pueblo y una buena gestión del tiempo, que en el caso del proyecto a desarrollar esta característica se ha descartado.

Estas características han inspirado Trufo's adventure tanto por sus mecánicas con las herramientas que posee para poder realizar todas las tareas de recolección, como por su estética y estilo artístico pixelado. En el caso del proyecto a crear, las mecánicas se han simplificado y desarrollado de forma que están accesibles directamente desde la pantalla principal del juego.

La interacción con los personajes de este proyecto se asemeja a la que se puede observar en en Haverst Moon, sin embargo, se ha adaptado y simplificado a la estética de Trufo's adventure.

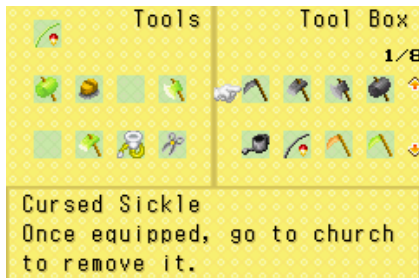


Figura 18: Harvest Moon tools



Figura 19: Harvest Moon interacción con personajes

Stardew Valley



Figura 20: Stardew Valley

Tal y como se ha detallado en la sección 2.1, Stardew Valley es un juego de simulación de granja muy similar al título Harvest Moon, debido a la inspiración del creador en este título. De este título se destaca la progresión a través de las cinco habilidades principales (agricultura, pesca, minería, combate y recolección), que han inspirado a las acciones en Trufo's adventure de una forma más simplificada (coger, cortar o cavar).

Asimismo, el diseño de la barra de herramientas, la atmósfera acogedora y la interacción con los personajes han sido la fuente de inspiración definitiva para el desarrollo de mecánicas de recolección de ingredientes, exploración y estética de este TFG. Como en el caso anterior se ha prescindido de la gestión del tiempo debido al tiempo limitado para el desarrollo de este proyecto ya que su desarrollo complicaría significativamente la creación del proyecto.

3.Propuesta

Este proyecto tiene un enfoque no comercial, va a ser desarrollado sin ánimo de lucro como herramienta y contribución al ámbito docente. Asimismo, el desarrollo de este prototipo como tiene como finalidad personal el desarrollo académico y formar parte del portafolio particular para futuras oportunidades profesionales en el sector de desarrollo de videojuegos.

3.1. Definición de especificaciones del producto

El producto es un prototipo de un nivel de un videojuego del género simulación y exploración básico para un solo jugador bajo la supervisión del docente. El producto estará disponible en español principalmente y en inglés debido a su carácter de aprendizaje del idioma.

Contará con unas mecánicas de juego sencillas basadas en la interacción con los personajes encontrados por el escenario y la recolección de los ingredientes solicitados para avanzar a las siguientes misiones. El jugador tendrá una barra de herramientas en la cual se selecciona la herramienta adecuada para la interacción con el entorno y la recolección de los ítems solicitados.

Asimismo, la interfaz de usuario cuenta con un apartado donde se incluye el texto que en todo momento informará al jugador de las solicitudes de los distintos personajes, así como información en momentos determinados para guiar al jugador, e información de ciertos ítems del escenario.

La ambientación destaca por escenarios coloridos, ambiente encantador, personajes carismáticos, un entorno que va desde zonas misteriosas de exploración, pasando por un prado de cultivo hasta una adorable granja.

Se creará un tráiler del prototipo que evidencie el *gameplay* de forma que los potenciales usuarios docentes puedan comprobar cómo se desarrolla y si puede ser una herramienta válida para sus proyectos que incluyen metodologías activas de aprendizaje.

3.2. Historia y objetivo



Figura 23: Trufo por Cupnoodle

El videojuego transcurre en una isla donde el jugador adopta el papel de Trufo, un inquieto explorador al que los habitantes de isla Kaio solicitan de sus habilidades de exploración y aventura para ayudarles a conseguir los ingredientes necesarios y así poder cocinar los mejores platos estrella que se presentarán en el gran festival de cocina de verano.

Trufo no se lo piensa y se embarca en una aventura de aprendizaje y exploración de la que no es por el momento consciente de todo lo que aprenderá en el transcurso de su viaje por las recetas que sus vecinos de la isla le tienen preparado. Todo se complica cuando el recetario donde tiene que consultar los ingredientes está completamente en inglés, y es aquí cuando realmente comienza su aventura de aprendizaje y diversión.

El prototipo contará con una serie de misiones que se activan secuencialmente según se finalizan hasta llegar a la *quest* final donde todos los personajes han podido cocinar sus platos, y se da por terminado el juego. Al ser un prototipo constará de tan solo seis misiones iniciales.

3.3. Objetivos del gameplay

El videojuego, como se ha mencionado, tiene como objetivo la exploración y recolección de ingredientes que los distintos personajes necesitan para completar la misión. Para ello, el jugador dispone de varias acciones que le ayudarán desde a recoger los ítems, interactuar con los personajes de la isla, hasta interactuar con ciertos objetos.

Se ha procurado un *gameplay* sencillo e intuitivo que permita centrarse en el aprendizaje conjugado con la diversión, por lo tanto, se prescinde de mecánicas muy elaboradas; se podría asemejar al *casual gaming*, ya que el tiempo de docencia es limitado y no puede extenderse demasiado en el tiempo.

A continuación, se detallan las acciones que el jugador podrá realizar para interactuar con todos los objetos del escenario.

Acciones

Acción	Descripción	Icono	Animación
Point	Acción para seleccionar objetos y obtener información de estos.		
Speak	Acción para interactuar con los personajes de la isla.		
Grab	Acción para recoger objetos que puedan ser agarrados.		
Dig	Acción para cavar y recoger objetos que puedan ser cavados.		
Cut	Acción que permite cortar y abrirse paso y desbloquear el camino.		

Tabla 3: Acciones gameplay

Ingredientes

A continuación, se detallan de forma general y agrupados los ingredientes que el jugador puede recolectar para completar las misiones activas (esta tabla es susceptible de ampliación):

























Item	Icono	Acción	Interacciones	
Eggs		Grab		
Milk		Grab		
Fruits		Grab		
Vegetables		Dig		
Mushrooms		Grab		
Sugar		Grab		
Chest		Grab		
Trees		Cut		

Tabla 4: Ingredientes gameplay

NPC

A continuación, se detallan los personajes con los que el jugador puede interactuar para iniciar las misiones):


Nombre	Icono	Acción	Interacciones	
Amina		Speak		
Aisha		Speak		
Li Mei		Speak		
Andrei		Speak		
Malik		Speak		

Tabla 5: NPC gameplay

3.4. Plataforma publicación

Se ha decidido publicar este videojuego en las plataformas de escritorio tanto para Windows como para MacOS y así poder garantizar el desarrollo dentro de los plazos establecidos para la realización de este TFG. Se ha considerado la publicación web, pero debido al alcance del proyecto y tiempo de implementación se ha descartado por el momento.

Para la descarga se ha decidido una descarga directa desde la web principal previo registro como docente. Asimismo, se ha valorado la plataforma de distribución itch.io [32] para su distribución y jugabilidad, pero hasta la finalización del proyecto no se podrá confirmar debido a la limitación en el tiempo del proyecto.

Por la finalidad de contribución al entorno docente, se ha descartado el despliegue en *smartphones* o *tablets*, ya que actualmente hay una gran polémica sobre el uso de estos dispositivos en el aula [33], por lo que se ha decidido no desarrollarlo en estas plataformas. Debido a que este prototipo está concebido como una herramienta para la comunidad educativa, el uso de este videojuego se aconseja siempre sea bajo supervisión del docente en su dispositivo de escritorio del aula.

3.5. Estrategia de Marketing

Este videojuego, como se ha indicado en anteriores apartados, no pretende comercializarse, sino ser una aportación como herramienta para el entorno docente y educativo, sin embargo, son aplicables los mismos consejos del sector que se han mencionado en el apartado 2.3 para una buena acogida del producto. Para ello se van a implementar las siguientes estrategias para llegar al mayor número de centros educativos posibles.

Se comenzará con la creación de la página web donde se pueda presentar y consultar información sobre el videojuego, beneficios educativos para los docentes en las metodologías activas de enseñanza, uso en talleres, aprendizaje por proyectos, etc.

Se gestionará la creación de perfiles en las redes sociales más importantes para llegar al mayor número de docentes posible: Facebook, Instagram, TicTok, Youtube y Twitch, y de esta forma podrán tener información periódica sobre las actualizaciones del videojuego y noticias del sector educativo.

Se buscará recopilar el máximo de testimonios de docentes que lo hayan implementado en sus clases y cómo les ha ayudado positivamente a los alumnos, y de esta forma nutrir la web de estas experiencias positivas.

Otra de las estrategias será mantenerse activo ante las preguntas frecuentes (FAQ) sobre el videojuego y así poder recoger *feedback* de las necesidades que puedan tener los potenciales docentes como usuarios del prototipo.

Presentar el videojuego en redes sociales especializadas en el entorno educativo es otro factor en el que hay que hacer hincapié, como por ejemplo INTEF [34] y TodoELE [35], entre otras, para enseñar como se puede integrar este prototipo en el entorno educativo.

Es vital colaborar con centros educativos a través de talleres de presentación del prototipo y cómo puede complementar su labor formativa, habrá que hacer un gran esfuerzo en este aspecto.

Finalmente, otra medida importante es mostrar en la web imágenes y vídeos del *gameplay* para que puedan tener una idea lo más visual posible de la mecánica del juego. También sería interesante poder grabar talleres donde se pueda observar el desempeño de este videojuego en un entorno real del aula.

4. Diseño

4.1. Entorno de desarrollo

El entorno de desarrollo elegido es **Unity Engine**. La elección de este entorno es debido a varios factores, entre ellos a la gran cantidad de información, manuales y material audiovisual que existe en la web, ser un motor de creación de videojuegos ampliamente reconocido y fiable, así como una comunidad muy activa en foros. Estos factores han sido decisivos para poder llevar a cabo este prototipo dentro del alcance del proyecto teniendo en cuenta la curva de aprendizaje y el tiempo disponible.

Otro de los aspectos fundamentales para la elección de este entorno, son los **requisitos técnicos para el desarrollo**. El prototipo de videojuego a desarrollar no requiere de gráficos 3D de última generación, y el IDE de desarrollo de Unity de acuerdo a los requisitos indicados en su página web [36] los cumplía el equipo de hardware del que se dispone.

La gran cantidad de *assets* disponibles en la Unity Asset Store gratuitos ha sido otro de los puntos a favor de la elección de este entorno, si bien finalmente no se han utilizado ningún recurso de esta plataforma, durante la fase de estudio de los distintos motores ha sido una opción valorada positivamente.

Finalmente, el lenguaje de programación en **C#** ha sido otro de los factores decisivos, ya que estoy familiarizado con este lenguaje y la curva de aprendizaje ha sido más sencilla de lo esperado, si bien es una adaptación de C# a la librería de Unity, es bastante intuitivo si se tiene conocimientos en C# y OOP.

Se han tenido en cuenta otros entornos de desarrollo como, por ejemplo:

- **Unreal Engine:** Se ha descartado por la elevada curva de aprendizaje y por los altos requisitos del sistema, concebido sobre todo para videojuegos en 3D triple A, que no es el prototipo a desarrollar en este caso.
- **Godot:** Se ha descartado por varias razones; la primera de ellas es que la comunidad, si bien esta creciendo exponencialmente, no hay tantos recursos y manuales como otros entornos de desarrollo. La segunda es que que, aunque actualmente es un motor que crece en popularidad, en la industria del videojuego todavía no es un perfil que busquen los estudios dedicados al desarrollo de videojuegos, por lo que a la hora de aumentar los conocimientos y el currículum me parecían más interesantes otros motores con mayor demanda. Finalmente, el uso del lenguaje GDScript con el cual no estoy familiarizado ha sido otro aspecto para descartarlo por temor a no llegar a cumplir objetivos del proyecto.

4.2. Herramientas utilizadas









Rider	Entorno de desarrollo C# en Unity de JetBrains.	
Unity 2022.3.8f1	Plataforma de desarrollo de videojuegos.	
Microsoft Office	Suite de oficina para documentación y planificación.	
Tom's Planner	Herramienta de planificación y gestión de proyectos.	
PiskelApp	Editor de <i>sprites</i> en línea.	
Gimp	Herramienta para edición de imágenes y diseño.	
Github	Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.	
Github Desktop	Aplicación gratuita de código abierto para trabajar con archivos hospedados en GitHub a través de interfaz gráfica.	

Tabla 6: Herramientas utilizadas.

Requisitos Técnicos del entorno de desarrollo

Requisitos técnicos mínimos indicados en la página web de Unity [36]:

Unity Editor system requirements			
This section lists the minimum requirements to run the Unity Editor. Actual performance and rendering quality might vary depending on the complexity of your project.			
Minimum requirements	Windows	macOS	Linux
Operating system version	Windows 7 (SP1+), Windows 10 and Windows 11, 64-bit versions only.	Mojave 10.14+ (Intel editor) Big Sur 11.0 (Apple silicon Editor)	Ubuntu 20.04 and Ubuntu 18.04.
CPU	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support (Intel processors) Apple M1 or above (Apple silicon-based processors)	X64 architecture with SSE2 instruction set support
Graphics API	DX10, DX11, and DX12-capable GPUs	Metal-capable Intel and AMD GPUs	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.
Additional requirements	Hardware vendor officially supported drivers	Apple officially supported drivers (Intel processor) Rosetta 2 is required for Apple silicon devices running on either Apple silicon or Intel versions of the Unity Editor.	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.)
For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer.			

Figura 24: Requisitos técnicos Unity.

4.3. Arquitectura general

La arquitectura del proyecto se basa en la forma que define Unity para estructurar sus proyectos. De acuerdo a los consejos de buenas practicas de Mikael Kalms [37], para este prototipo de videojuego se han utilizado *GameObjects*, *Scripts*, *Prefabs*, *ScribableObjects*, y la clase base de Unity que es *MonoBehaviour*.

GameObjects

Los *GameObjects* son cualquier objeto que se añade a la escena, ya sea luces, personajes, cámara, etc. al cual se le puede añadir una lógica de comportamiento a través de scripts, así como componentes que modificarán su apariencia, funcionalidad, posición en la escena, colisionadores, físicas, etc.

Scripts

Componente que se adjunta a los *GameObjects* para otorgar funcionalidad ampliada, activar eventos del juego, lógica más avanzada y específica de acuerdo a las necesidades particulares del videojuego. Estos scripts por defecto heredan de la clase base *MonoBehaviour*, que se explicará en las siguientes líneas. Asimismo, estos Scripts podrán interactuar con los distintos componentes del mismo *GameObject* así como otros elementos presentes en la escena.

ScriptableObjects

Los *ScriptableObjects* son una clase especial que permite almacenar información a modo de instancia de datos que pueden ser compartidos por cualquier elemento de la escena. Existen muchos autores que defienden el uso de esta clase para sus proyectos, como Ryan Hipple [38] y Richard Fine [39]. A diferencia de los *Scripts*, estos no pueden acoplarse a un *GameObject*. Al ser almacenados como un *Asset* del proyecto, se puede interactuar con ellos a través del inspector de Unity y son útiles para almacenar información estática compartida por cualquier *GameObject*, lo que minimiza drásticamente las instancias necesarias al referenciar todos los *GameObjects* el mismo valor desde un único *Asset*. Asimismo, estos pueden albergar lógica como los scripts, lo que los hace muy versátiles como se mostrará en posteriores apartados.

Prefabs

Los Prefabs son *GameObjects* que hemos configurado de una manera específica, con sus *Scripts*, componentes, etc. y lo guardamos como un *Asset* que podrá ser replicado en la escena sin tener que añadir todos estos componentes cada vez. De esta forma se centraliza la información del *Prefab*, y si es modificada desde el editor se replicará al resto de copias de la escena facilitando el desarrollo. A diferencia de los *ScriptableObjects*, donde los datos se encuentran en un único lugar y son referenciados por los *GameObjects*, los *Prefabs* en escena tienen cada uno una copia de los datos en ejecución, lo que implica el aumento del uso de memoria ya que cada instancia modifica sus valores.

MonoBehaviour

Unity define una clase principal que se denomina *MonoBehaviour* que permite que *GameObjects* puedan interactuar con los *Scripts*, además de ofrecer funcionalidades base para el control del flujo de la escena y el ciclo de vida. Las funcionalidades básicas se pueden consultar en la web de Unity [40].

Todos ellos quedan enmarcados dentro de una escena jerarquizada con todos los *GameObjects* necesarios y sus componentes.

4.4. Patrones de diseño

Para el desarrollo de este prototipo se han aplicado los patrones de diseño más populares como el *Singleton*, *Observer* y el basado en eventos.

El patrón ***Singleton*** proporciona una única instancia del objeto en la escena accesible por otras clases. Se ha utilizado para elementos como los controladores que se necesita sean únicos para que puedan ser accedidos por cualquier *GameObject* o clase de la escena y tener la certeza que no hay otra instancia.

El patrón ***Observer*** nos permite un desacoplamiento entre clases de la escena, es decir, cuando una clase informa de un cambio, el resto de clases que están observando este cambio son informadas y realizan las acciones necesarias en cada caso particular. Se definen las clases *subject* y las *observer*, y ninguna de ellas sabe de la existencia de la otra. Los observadores se registran y des registran según sea necesario.

En este proyecto se ha utilizado este patrón *observer* para la gestión de **eventos**, de forma que una sola clase estática es la encargada de centralizar todos los eventos en lugar de tenerlos dispersos en varias clases. El resto de clases se suscriben a los eventos específicos que necesitan. Esta organización proporciona un sistema más ordenado y desacoplado, facilitando la gestión y mantenimiento de los eventos en el proyecto.

4.5. Estructura del proyecto

Se ha buscado seguir una lógica de buenas prácticas recomendada por Unity.



Figura 25: Estructura organizativa Assets

4.6. Listado de clases

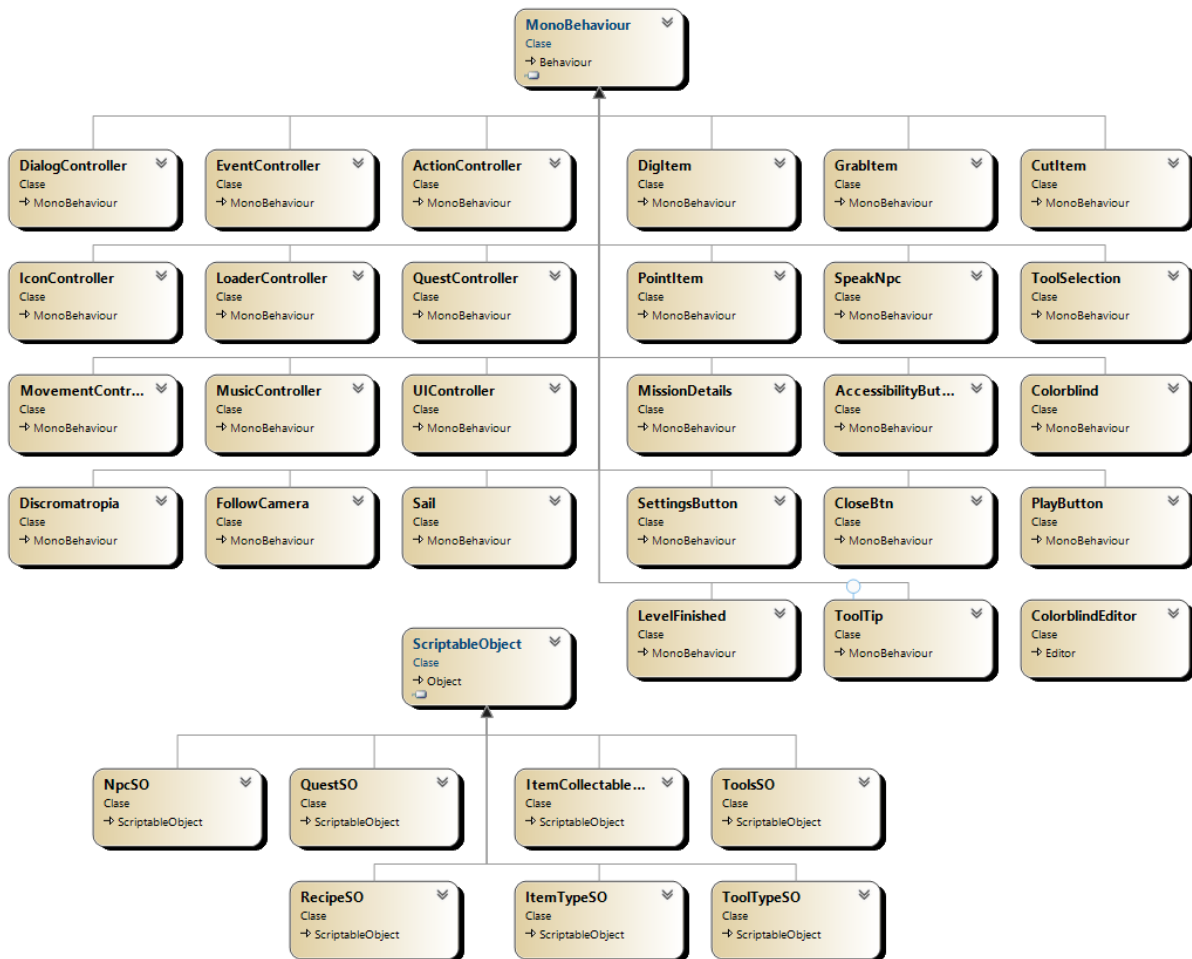


Figura 26: Diagrama visual de las clases

A continuación, definición de las clases por orden alfabético:

ActionController.cs: Clase encargada de ejecutar la lógica y animación dependiendo de la acción seleccionada. Utiliza el patrón *Singleton* para asegurar que solo exista una instancia ya que es necesaria en todos los niveles, aunque en este prototipo solo se ha implementado un nivel.

AccessibilityButton.cs: Script encargado de la lógica de cambio de la UI y elementos del juego para adaptarla a personas que requieran de mejoras de accesibilidad visual.

CloseBtn.cs: Script encargado de la lógica de cierre de los menús.

CutItem.cs: Script encargado de comunicar al ActionController si la acción de cortar sobre el objeto es realizable o no, y en caso de que lo sea activa la lógica correspondiente.

DialogController.cs: Clase encargada de encargada de mostrar los diálogos de las misiones e información de los objetos seleccionados en la UI así como el nombre del NPC que habla.

DigItem.cs: Script encargado de comunicar al ActionController si la acción de cavar sobre el objeto es realizable o no, y en caso de que lo sea activa la lógica correspondiente.

Discromatropia.cs: Script encargado de comunicar al UIController la selección de cambio de modo de visión para personas con Discromatropia y sea activa la lógica correspondiente para cambiar el modo de presentación de los colores en pantalla según el modo seleccionado.

EventController.cs: Clase encargada controlar los eventos con eventos estáticos y proporciona métodos para invocar y notificar eventos, centralizando la gestión de eventos y facilitando la comunicación entre diferentes clases. También utiliza el patrón Singleton ya que es necesaria en todos los niveles y asegurar que solo exista una instancia.

FollowCamera.cs: Script encargado de seguir al jugador cuando se desplaza por la escena.

GrabItem.cs: Script encargado de comunicar al ActionController si la acción de recoger sobre el objeto es realizable o no, y en caso de que lo sea activa la lógica correspondiente.

IconController.cs: Script encargado de gestionar la activación y desactivación del icono indicador de misiones activas que posee cada NPC. Utiliza eventos del EventController para responder a cambios en las misiones.

ItemCollectableSO.cs: ScriptableObject que guarda información del objeto recolectable y del tipo de objeto.

ItemTypeSO.cs: Contiene una enumeración ItemType que enumera varios tipos de objetos del juego, como huevo, fruta, vegetal, árbol, ingrediente y libro. Este scriptable object proporciona una manera estructurada de definir y gestionar los tipos de elementos del ItemCollectableSO.

LevelFinished.cs: Script encargado de la lógica a ejecutar cuando termina el nivel.

LoaderController.cs: Clase encargada de la gestión la transición entre niveles del juego, implementada con un patrón *Singleton* para asegurar que solo exista una instancia. Se utilizan eventos de carga de escena para controlar las animaciones de carga y la activación/desactivación del objeto de carga.

MissionDetails.cs: Script encargado de mostrar y ocultar el panel de detalle de las misiones. Se suscribe a eventos del EventController para escribir y limpiar textos de misiones.

MovementController.cs: Clase encargada de la gestión del movimiento del jugador en la escena. Se Implementa un patrón *Singleton* para garantizar una única instancia y es el encargado de recoger eventos de ratón y de teclado para realizar acciones específicas, y comunicarse con el ActionController para ello. También se encarga de cambiar entre el estado mover y parar a través de un Animation Event creado en la animación del jugador en el momento en el que el *collider* de la animación golpea un elemento y hace que el movimiento del player quede congelado.

MusicController.cs: Clase encargada de la gestión de la reproducción de distintos sonidos en el juego. Se implementa un patrón *Singleton* para garantizar una única instancia y se suscribe a eventos del EventController para activar la reproducción de sonidos específicos en respuesta a acciones del juego.

NpcSO.cs: SriptableObject que contiene propiedades de los NPC del juego como su imagen y la herramienta de interacción asociada de la clase ToolSO.

PlayButton.cs: Script que controla el comportamiento del botón *play* en el juego que se comunica con el LoaderController para iniciar el nivel.

PointItem.cs: Script que controla la interacción con un objeto señalado en el juego. Se encarga de comunicar al ActionController si la acción de señalar sobre el objeto es realizable o no, y en caso de que lo sea activa la lógica correspondiente.

QuestController.cs: Clase encargada de la gestión de las misiones en el juego. Se implementa un patrón *Singleton* para garantizar una única instancia. Contiene una lista de misiones y se suscribe eventos como la activación de iconos de misión y la finalización de misiones. También es el encargado de la lógica de informar sobre la misión actual, inicio, finalización y seguimiento, así como de comprobar si ya se han realizado todas. Asimismo, gestiona la recolección de elementos relacionados con la misión.

QuestSO.cs: ScriptableObject que contiene información de una misión del juego. Contiene propiedades como el nombre de la misión, un identificador único, el NPC que inicia la misión, texto descriptivo de la misión y diálogos relacionados con el NPC. Además, tiene propiedades para verificar el estado de la misión: si ha comenzado, si ha sido completada, y si se han recolectado los elementos necesarios. Asimismo, posee la información sobre la receta de la misión RecipeSO.

RecipeSO.cs: ScriptableObject que contiene información de una receta con propiedades como el nombre de la receta, un identificador único, y una lista de elementos necesarios para completarla. Cada elemento de la receta consiste en un objeto recolectable necesario, la cantidad requerida, y la cantidad recolectada actualmente, que se reinicia a 0 cada vez que el ScriptableObject se activa en la escena.

Sail.cs: Script que controla la interacción del jugador con el barco, cuando el jugador entra en el área de la vela, se desencadena una animación y se ajusta la posición del jugador. Luego, se activan o desactivan los Colliders que rodean el barco para evitar que el jugador se precipite.

SettingsButton.cs: Script está asociado al un botón Guía. Cuando se acciona muestra la guía del juego; se utiliza para mostrar o esconder el menú de guía del juego.

SpeakNpc.cs: Script que controla la interacción con un NPC señalado en el juego. Se encarga de comunicar al ActionController si la acción de hablar sobre el NPC es realizable o no, y en caso de que lo sea activa la lógica correspondiente. Asimismo, administra misiones a través del QuestController, se controla activación/desactivación del NPC basado en misiones no terminadas, y se gestiona la interacción del ratón para iniciar/completar

misiones. Se suscribe a eventos del EventController relacionados con misiones y el final del nivel.

ToolSelection.cs: Script asociado a los botones de las herramientas de la UI, que controla la selección de herramientas, de manera que, al hacer clic en un botón de herramienta, se dispara el evento SelectedToolEvent con la herramienta asociada se realiza la lógica asociada.

ToolsSO.cs: ScriptableObject que contiene información sobre las herramientas del juego. Cada herramienta tiene una acción asociada, imágenes cuando está activa e inactiva, y un texto de información.

ToolTip.cs: Script asociado a objetos para mostrar información adicional. Implementa las interfaces IPointerEnterHandler e IPointerExitHandler que, al detectar el puntero sobre el objeto asociado, activa un cuadro de texto con información sobre la herramienta asociada al objeto y al salir se desactiva.

ToolTypeSO.cs: ScriptableObject que define un enumerador ToolTypeAction con acciones de herramientas: Cut, Dig, Grab, Point y Speak.

UIController.cs: Clase encargada de la gestión de la interfaz de usuario, en concreto la barra de herramientas. Se suscribe al evento del EventController para cambiar la apariencia de los botones de herramientas y el cursor cuando se selecciona una herramienta. También se encarga de la lógica que activa y desactiva la barra de herramientas según sea necesario.

Las clases con más relevancia y que cabe destacar en este proyecto son la QuestController y EventController, sobre las que recae el peso de la funcionalidad del proyecto. EventController facilita la comunicación entre diversas clases mediante la centralización de la gestión de los eventos del juego, consolidando de esta manera el control y la eficiencia en la manipulación de eventos dentro del sistema.

A continuación, un diagrama de cómo se ha diseñado.

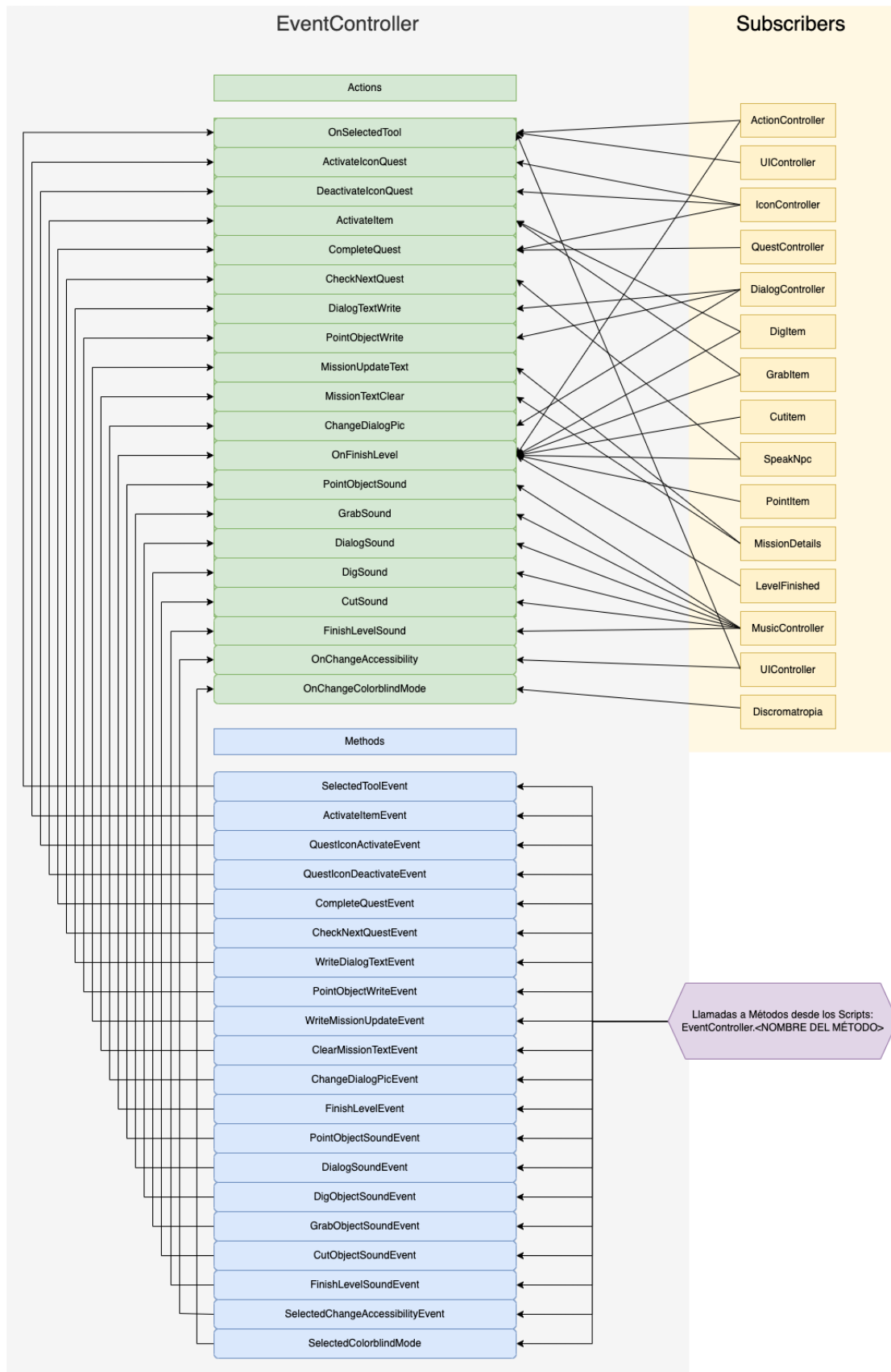


Figura 27: EventController esquema implementación.

4.7. Diagrama de flujo de escenas

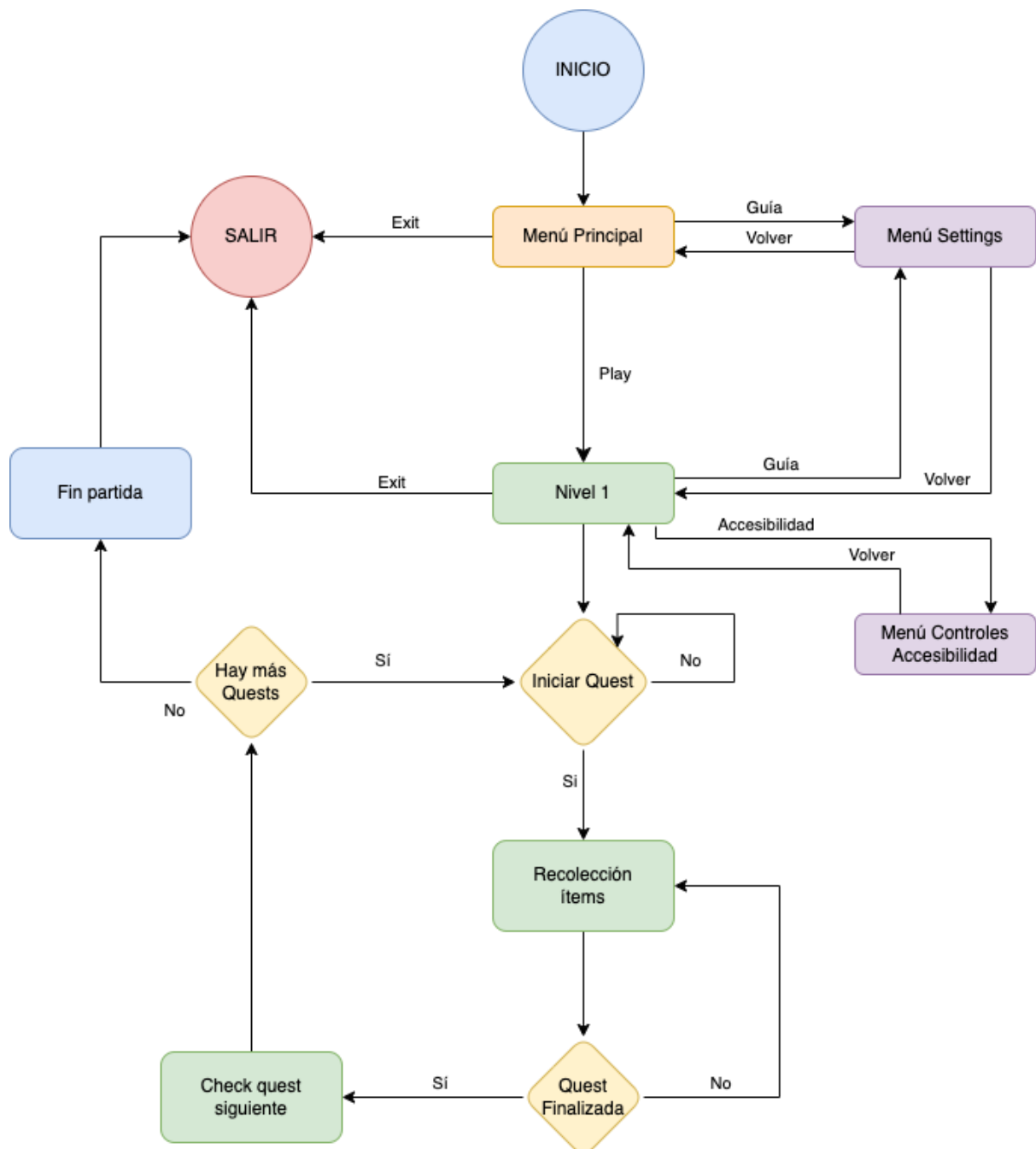


Figura 28: Diagrama de flujo de escenas

Para el flujo de navegación detallado se ha tenido en cuenta el rango de edad del jugador al que va destinado. Se ha buscado sencillez para que los jugadores no tengan dificultad a la hora de entender cómo se desarrolla la jugabilidad de la escena.

4.8. Diseño gráfico e interfaces

4.8.1. Estilos

Logotipo



Figura 29: Logotipo

Paleta de Colores

Paleta de colores del Asset Sprout Lands Asset Pack y Sprout Lands UI Asset Pack.

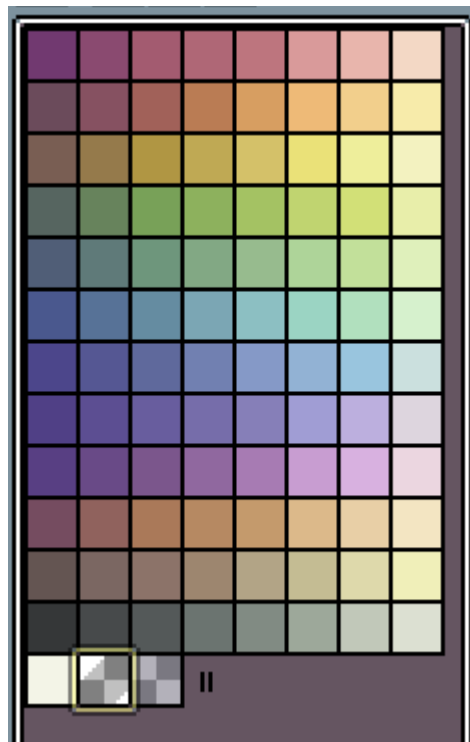


Figura 30: Paleta de colores

Fuentes

Se han utilizado fuentes gratuitas de uso personal y comercial descargadas desde la web Dafont.

Fuente **Hugloves** utilizada para el Título del menú inicial:



Figura 31: Fuente título menú inicial

Fuente **Rainy Hearts** utilizada para los textos de la UI, botones y diálogos:

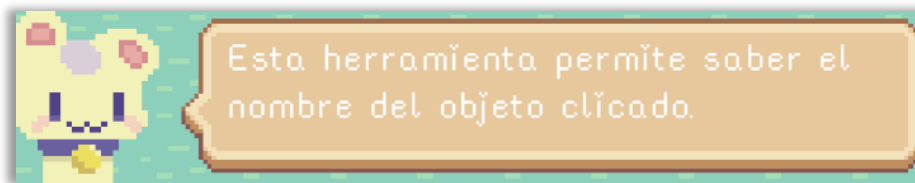


Figura 32: Fuente utilizada en UI, botones y diálogos

UI

Para la interfaz gráfica se ha utilizado el **Sprout Lands UI Asset Pack**.



Figura 33: Assests UI

4.8.2. Diseño del nivel

El mapa del nivel se ha creado buscando una experiencia y sensación al jugador de transición entre entornos distintos, para evitar la monotonía al cambiar de isla.

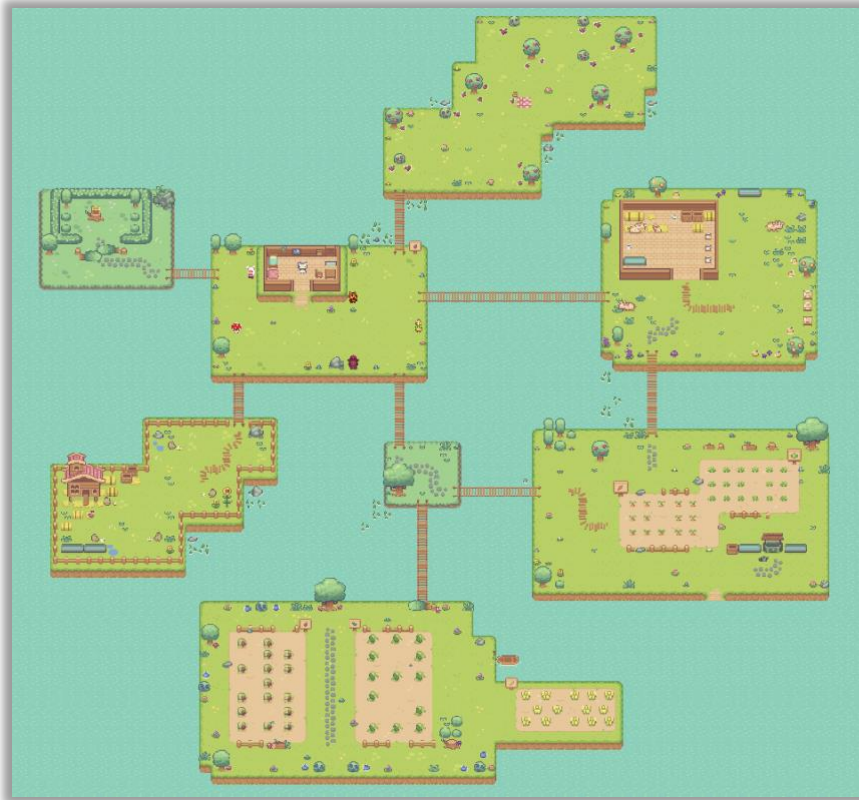


Figura 34: Vista general nivel

Los puentes son el eje central que interconecta las islas y que permite al jugador explorar el nivel de una manera más fluida.



Figura 35: Imagen puentes

Asimismo, para hacerlo más dinámico se ha implementado un barco que permite navegar de una isla a otra y así aportar variedad a la exploración del nivel:



Figura 36: Barco navegación

La barra de acciones es otro de los pilares fundamentales del prototipo, a través de ella el jugador realizará todas las acciones necesarias: Point, Speak, Grab, Dig, Cut. Se han colocado en la parte inferior central de la pantalla.



Figura 37: Barra de acción

Se ha diseñado los Animation Event en las animaciones de las acciones Dig y Cut. Se ha llamado en el último *frame* de ambas animaciones una acción que detiene el movimiento; además se activa un *Collider* que es el que interactúa con el *Collider* del objeto a recolectar.

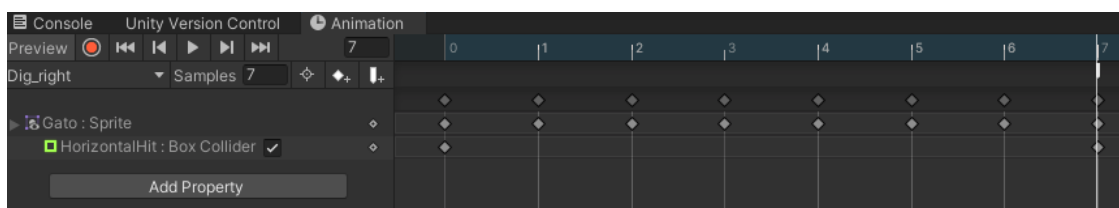


Figura 38: Animation Event que controla el movimiento.



Figura 39: Activación Collider. Izquierda Dig, Derecha Cut.

4.8.3. Accesibilidad

Para la accesibilidad se han implementado varias opciones de personalización dependiendo de las necesidades de accesibilidad del jugador tanto auditivas como visuales.

A continuación, listado de opciones a personalizar:

- **Tamaño de pantalla:** Mediante un botón deslizador se puede cambiar a modo de aumento, donde los personajes y objetos recolectables aumentan de tamaño.
- **Modificar sonido:** A través de un *slider* personalizado se puede aumentar o disminuir el audio del juego.
- **Discromatopsia:** Los jugadores con esta afección pueden modificar el modo de visualización atendiendo a las tres más comunes: **Protanopia** (Rojo), **Deuteranopia** (Verde) y **Tritanopia** (Azul). Para ello se han implementado tres botones deslizadores para elegir el modo deseado.



Figura 40: Controles de accesibilidad.

Para las necesidades visuales, se ha hecho uso de la herramienta gratuita desarrollada por Project Wilberforce, **Colorblind Effect**, que gestiona un *shader* para modificar el renderizado de la pantalla. A continuación, una muestra de cómo se visualiza con cada uno de los modos activados:



Figura 41: Modos visualización: Protanopia, Deuteranopia y Tritanopia.

4.9. Inventario de Assests y recursos

Para el desarrollo de este videojuego, no he creado ningún asset, todos han sido descargados con licencia para poder ser modificados y utilizados en la escena. Se han adaptado algunos assets con las herramientas anteriormente mencionadas para una mejor visualización. A continuación, se detallan un inventario de todos los recursos utilizados y modificaciones realizadas:

Recursos gráficos

Sprout Lands UI	
Pack de recursos para el desarrollo de la interfaz de usuario. Desarrollado por Cup Noodle. Link	
Emoticonos Player	
Jugador	
Panel	
Icono de inicio de Quest	
Zona de diálogo	
Flechas menú guía	
Botones	
Iconos	
Se han modificado los siguientes recursos para ampliar su tamaño a 24 píxeles con la herramienta Gimp, y se ha añadido una flecha para que se vea más claro la zona de acción con la herramienta online Piskel.	
	

Tabla 7: Sprout Lands UI recursos gráficos

Sprout Lands Asset Pack	
Pack de Assests para la creación del nivel. Desarrollado por Cup Noodle. Link	
Ingredientes	
Árboles	
Frutas	
Vegetales	
Objetos	
Herramientas / iconos	
Agua	
Miscelánea	
Tiles	



Animales	
<p>Se ha modificado estos recursos añadiendo una flecha para que se vea más claro la zona de acción con la herramienta online Piskel, y se han aumentado a 24 px. su tamaño con la herramienta Gimp.</p>	
	

Tabla 8: Sprout Land Asset Pack recursos gráficos






Pixel Adventurer - Free Animated Sprite Character	
Pack de personaje animado creado por Florassence. Link	
Simple Top Down Character	
Pack de personaje animado creado por BDragon1727. Link	
Pixel Cute Bear 1	
Pack de personaje animado creado por Immunity. Link	
Cute Mushroom Character Sprite	
Pack de personaje animado creado por Caz Creates Games. Link	
Fantasy RPG townsfolk pack	
Para de personajes animados creado por Franuka. Link	

Tabla 9: NPC recursos gráficos

Recursos Audio

Freesound	
Web Freesound con sonidos de licencia gratuita. Link	
Sad Piano.mp3	Música del menú de inicio.
Happy 8-bit music	Música de fondo durante el nivel.
Retro Hit Sound	Sonido para la acción talar.
Retro, Coin 04.wav	Sonido para la recolección de objetos.
Digging, Ice, Hammer, A.wav	Sonido para la acción cavar.
Huh.amp3	Sonido que inicia el diálogo con los NPC.
Text-Message or Videogame-Jump	Sonido que se ejecuta cuando clicamos sobre un objeto y nos sale la información en la zona de diálogo
Cheer	Sonido que se ejecuta al terminar el nivel.

Tabla 10: Freesound recursos audio



Fuentes de Texto	
Rainy Hearts	Fuente Utilizada para la estética textual del juego. Link
	 <p>The image shows a font preview for 'Rainy Hearts' by Camellina. It displays the text 'rainyhearts.ttf', 'Trufo's adventure', and 'Trufo's adventure' in a pixelated, monospaced font style.</p>
Hugloves	Fuente Utilizada para el título del juego en la pantalla principal. Link
	 <p>The image shows a font preview for 'Huglove' by Khurasan. It displays the text 'Huglove.ttf' and 'Trufo's adventure' in a stylized, handwritten font style.</p>

Tabla 11: Fuentes de texto

4.10. Evolución del diseño

En el concepto inicial de este videojuego, se contempló su desarrollo para tabletas y teléfonos móviles, visualizando las posibilidades interactivas que estos dispositivos ofrecen, movilidad y la integración que tienen en toda la sociedad. Sin embargo, esta modalidad se tuvo que rechazar ya que actualmente en los centros educativos, la presencia de estos dispositivos genera muchos conflictos y distracciones en lugar de facilitar la experiencia de aprendizaje, y hay una tendencia cada vez mayor para su prohibición en el aula y entornos educativos [41]. Esta consideración llevó a descartar la idea original y se replanteó la estrategia de diseño.

Para la elección del control de las mecánicas, inicialmente se había considerado utilizar el teclado en su totalidad para la interacción con el entorno, ideando teclas para diversas funciones como la recogida de ítems y la interacción con NPC, pero se ha reconsiderado la necesidad de hacer el juego más intuitivo para niños de 7 a 11 años, y se ha decidido cambiar a la simplicidad del ratón y apuntar en la pantalla como interfaz principal frente a tener que memorizar las teclas adecuadas para cada acción.

En cuanto al diseño de las mecánicas, se ha propuesto encontrar un equilibrio entre simplicidad y versatilidad. Se han reducido las acciones a cinco mecánicas fundamentales: talar, cavar, agarrar, hablar y señalar. Estas acciones permiten una interacción con el entorno y los NPC, sin caer en la complejidad que podría suponer para el público objetivo si se hubiesen introducido un mayor número y complejidad de las mismas.

Inicialmente se ha planteado la implementación de un inventario de forma que los jugadores visualicen los elementos recolectados en cualquier momento del juego. Sin embargo, al analizar el sistema de *quests* planteado, se hizo evidente que el inventario no aportaba nada significativo a la experiencia de juego más allá de algo meramente visual e incluso distractor, dado que las *quests* están diseñadas para activarse automáticamente cuando ciertos requisitos son cumplidos; por lo tanto, se ha decidido descartar la idea del inventario y centrar el esfuerzo en optimizar la progresión del juego.

La implementación de la acción Grab ha dado ciertos problemas, ya que se comenzó la implementación con una estructura similar a las acciones Cut y Dig a través de *Colliders2D*, pero la jugabilidad no era la esperada y la lógica se complicaba demasiado, por ello se ha optado por una implementación a través de *Raycasts2D* diferente a las otras dos acciones mencionadas que lo simplifica y se obtiene el resultado esperado.

En cuanto al diseño de los iconos de las herramientas y del puntero, se han encontrado problemas a la hora de definir el área del Sprite donde se clica y se realiza la acción. Se ha intentado modificar la zona de acción del Sprite con los distintos métodos que Unity ofrece, pero no daba un resultado satisfactorio. Debido a esto se ha tomado la decisión de editar los *sprites* de los iconos de Point, Speak, Dig y Grab añadiendo una pequeña flecha que indica el punto donde el ratón clica.

Por otro lado, se ha modificado el tamaño de estos iconos incluidos en el *asset* con la herramienta Piskel, ya que su versión por defecto era minúscula para poder jugar de una forma satisfactoria, por lo tanto, se ha tenido que modificar la resolución de los *sprites* a 24px. para poder mejorar la experiencia de juego. Asimismo, se ha modificado el color de los iconos del puntero a escala de grises, necesarios para poder evidenciar que la acción que se desea realizar no es posible.



Figura 42: Modificación con Piskel y Gimp del color, tamaño y flecha punteros

El sistema de *quests* sigue un enfoque simple y genérico que se puede ver en diversos videojuegos: se ofrece una *quest*, se activa con el NPC correspondiente, se recolectan los elementos y se entrega de nuevo al NPC que la ha solicitado. Esta simplicidad es crucial, ya que el videojuego ha sido concebido como una herramienta didáctica que se utilizará en el aula bajo la supervisión del docente.

Para la acción de diálogo, donde se muestra texto por pantalla, la imagen a mostrar es una modificación del *asset* original de cada personaje con Piskel para ajustarlo al marco del cuadro de la imagen que indica qué NPC está hablando en cada momento.

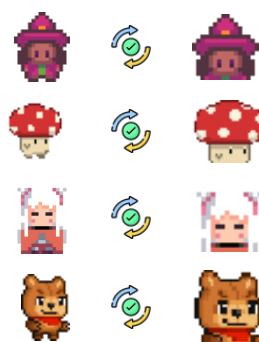


Figura 43: Edición NPC

Se han evaluado opciones más desafiantes, tras el estudio de distintas mejoras, como *quests* con temporizador límite para realizarla, restricciones para alcanzar ciertos objetos si no se cumplen ciertas condiciones, cambios dinámicos del escenario para que los ítems cambien de ubicación temporalmente, pero se han rechazado para no comprometer la experiencia de aprendizaje, evitando posibles niveles de estrés que podrían surgir al incorporar capas adicionales de dificultad y afectar a función principal de enseñanza.

4.11. Estructura de escenas

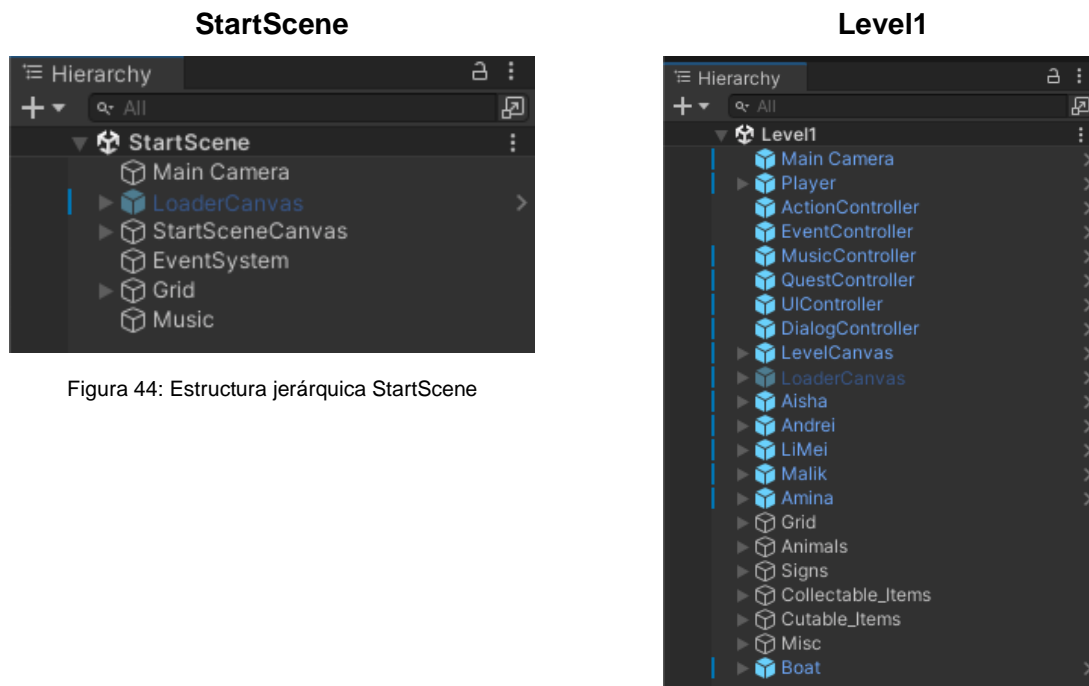


Figura 44: Estructura jerárquica StartScene

Figura 45: Estructura jerárquica Level1

4.12. Feedback cuestionarios

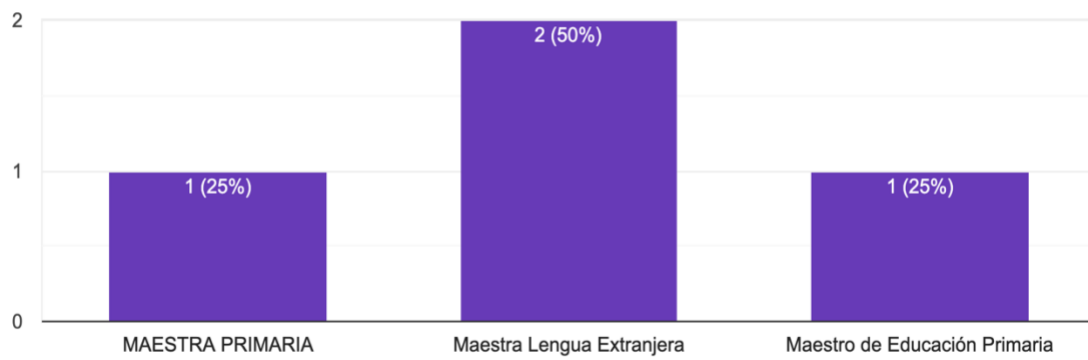
Se ha realizado un cuestionario a docentes en activo que han experimentado con una demo del prototipo para que evalúen la viabilidad de este videojuego como herramienta de docencia. Esta demo ha tenido que subirse a la plataforma Itch.io debido a las restricciones de instalación de aplicaciones en el centro educativo de los encuestados, lo que ha dificultado su instalación; se ha valorado y esta solución ha sido la más óptima y ágil teniendo en cuenta el tiempo disponible.

A continuación, se mostrarán las preguntas formuladas e información recopilada, así como los resultados y conclusiones del cuestionario a estas cuestiones.

En el [Anexo C](#) se pueden consultar las evidencias con los cuestionarios íntegros.

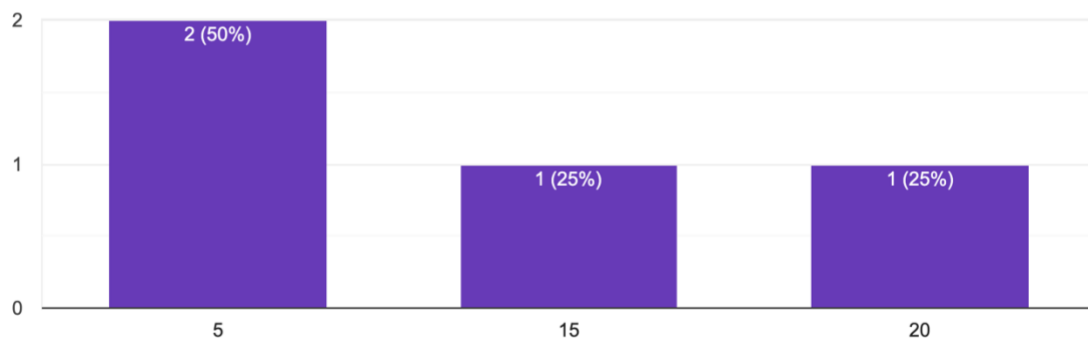
Titulación

4 responses

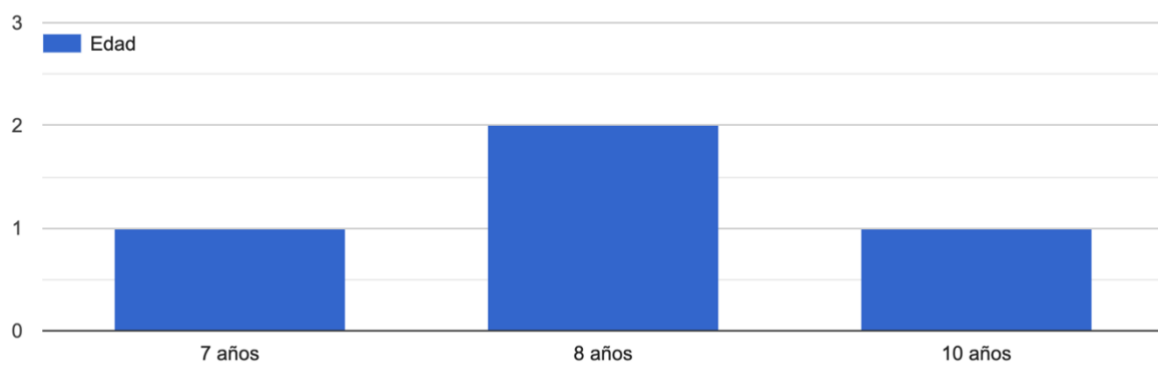


Años de experiencia

4 responses

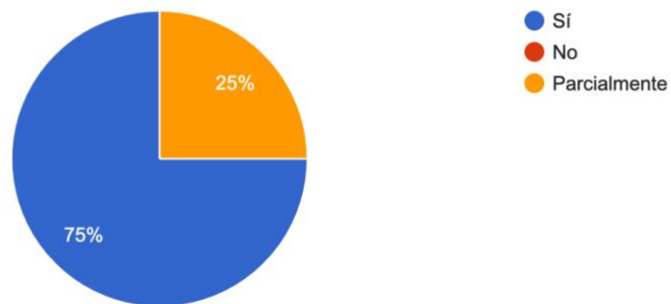


Rango de edad de los niños con los que trabajas



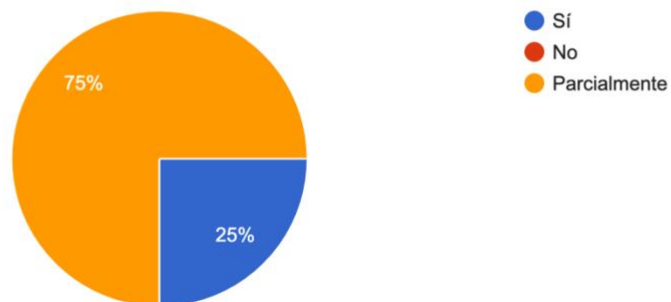
¿Crees que el videojuego puede lograr captar la atención de los niños de 7 a 11 años?

4 responses



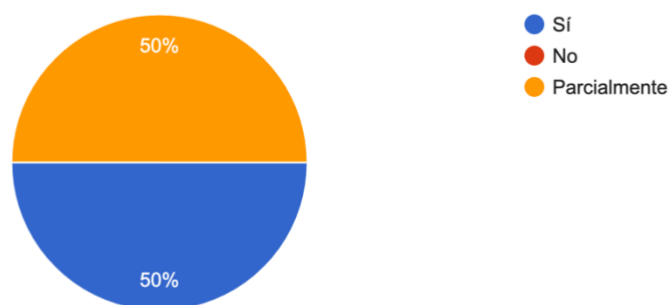
¿Consideras que los objetivos educativos del juego están claramente definidos?

4 responses



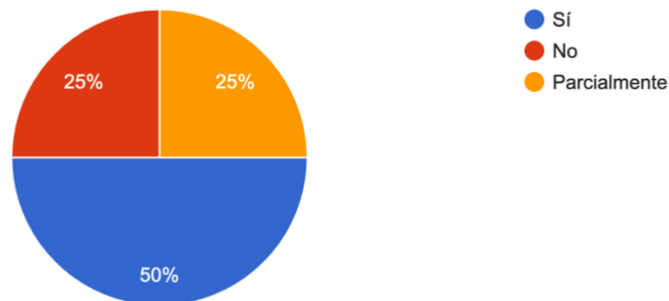
¿Encuentras el aspecto gráfico del juego fácil de entender para los niños del rango de edad indicada?

4 responses



¿Crees que el contenido del juego es adecuado en términos de dificultad para los niños de 7 a 11 años?

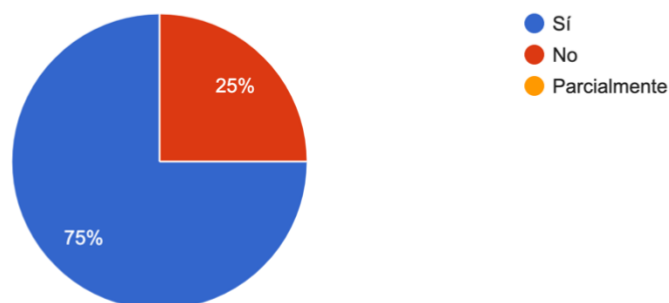
4 responses



Ante esta cuestión, la profesora que imparte clases a alumnos de 7 años, ha evidenciado que en esta franja de edad el alumno está en proceso de adquisición de la lectoescritura y no han desarrollado en su totalidad la competencia digital. Por ello piensa que sería más apropiado para otro rango de edad superior.

¿Te parece que el juego ofrece suficiente variedad para mantener el interés de los niños a lo largo del tiempo?

4 responses



En esta cuestión, se produce la misma situación que en la anterior, para niños de 7 años no lo ve apropiado.

Se ha evidenciado también que esta herramienta de docencia en forma de videojuego, ayuda a fomentar la mejora en la competencia digital, de acuerdo a sus observaciones los juegos que desarrollan en el aula son más sencillos.

Asimismo, este prototipo ayuda a practicar la comprensión lectora y entender objetivos. Por otro lado, durante la jugabilidad se ha apreciado que es capaz de captar la atención y “un pequeño enganche”.

Finalmente, se ha evidenciado que el nivel para un colegio Bilingüe no parece una herramienta válida, pero puede servir como herramienta de apoyo para aquellos alumnos con más dificultades y también para la atención a la diversidad.

A continuación, evidencias de alumnos interactuando con el prototipo demostración.



Figura 46: Evidencias de uso de la demo del prototipo

5. Implementación

5.1. Requisitos de instalación

Para la ejecución de este prototipo de nivel del videojuego Trufo's adventure, no es necesario un equipo con Hardware de última generación, debido a la naturaleza del mismo, un entorno 2D con animaciones y mecánicas sencillas, se puede ejecutar en la mayoría de los PC/MAC de sobremesa y equipos portátiles.

En cuanto requisitos de software, requiere un sistema operativo de 64 bits tanto para Windows como MacOS, y para ejecutarlo en la plataforma Itch.io un navegador con Javascript activado. Es necesario disponer de un teclado y ratón o *trackpad* para los controles del juego.

Requerimientos del Mínimos del sistema:

- Sistema Operativo: Windows 7 o superior, MacOS 10.12 o superior.
- Procesador: Intel Core i3 / AMD equivalente o superior. En MAC procesador Intel Core i3 (no compatible con procesadores ARM).
- Memoria RAM: 2 GB o superior.
- Tarjeta Gráfica: Gráficos integrados o dedicados con al menos 512 MB de VRAM.
- Almacenamiento: 80,7 MB de espacio disponible en disco para MAC, y 88,3 MB para la versión de Windos.

5.2. Instrucciones de instalación

La instalación es sencilla, basta con acceder al repositorio de Github en el apartado releases accesible en este enlace: <https://github.com/Ocester/TFG/releases>

Seguidamente, se seleccionará la versión del videojuego deseada, que está disponible tanto para MAC como para Windows, y se descarga el archivo comprimido en formato .zip; se descomprime y se procede a abrir el ejecutable con el nombre del prototipo Trufo's adventure incluido en este para comenzar a jugar.

6. Demostración

6.1. Instrucciones de uso

Para comenzar a Jugar a Trufo's adventure hay que familiarizarse con los controles de movimiento que permiten desplazarse por el nivel mediante las teclas A, S, D y W:

- A: desplazamiento lateral izquierdo.
- S: desplazamiento vertical abajo.
- D: desplazamiento lateral derecho.
- W: desplazamiento vertical arriba.

Asimismo, se dispone de una barra de herramientas para interactuar con los elementos del nivel que permiten realizar las acciones Point, Speak, Grab, Dig, Cut. Para poder seleccionar estas acciones es necesario posicionarse sobre ellas con el cursor y hacer clic izquierdo para seleccionarlas y activarlas. Una vez activa, se puede observar que el ratón se modifica y adopta la imagen de la herramienta elegida, y de esta forma esta lista para interactuar con el entorno o los personajes.

En cualquier momento el jugador puede consultar la guía de uso de controles si se pulsa sobre el botón "Guía" que está disponible en la parte inferior derecha de la pantalla. Asimismo, si desea salir en cualquier momento se puede presionar el botón "Exit" que se encuentra en la parte inferior derecha, o la tecla "Q" para finalizar la partida y salir del juego.

6.2. Guía de usuario

Nada más comenzar el videojuego, el jugador se encontrará con la pantalla de inicio con tres opciones: Play, donde podrá iniciar la partida, Guía, donde se podrá consultar los controles básicos de movimiento, así como las distintas herramientas disponibles y las acciones que realizan cada una; por último, se podrá salir del videojuego pulsando el botón Exit.

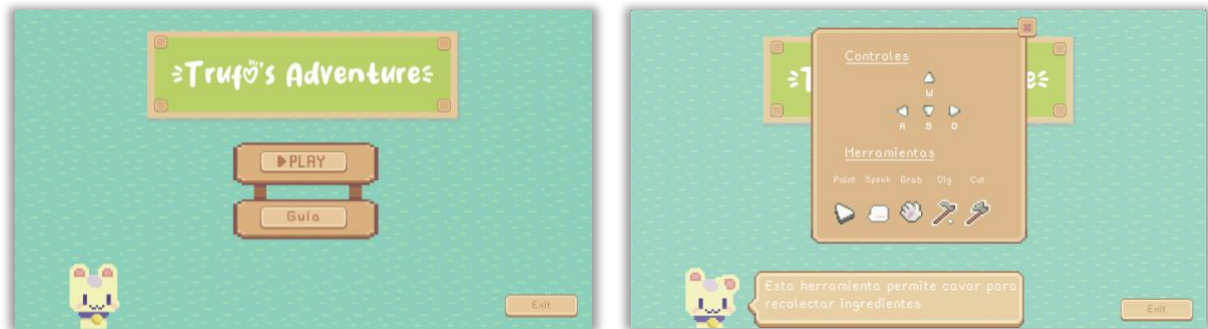


Figura 47: Menú inicio (izquierda)

Figura 48: Menú Guía (derecha)

Una vez se pulsa Play, se accede al primer nivel tras la pantalla de carga. Nada más iniciar el nivel se visualizará una breve introducción y bienvenida al jugador, una vez cerrada esta introducción, comenzará el juego. El jugador deberá de encontrar al aldeano con un símbolo "!" que parpadea sobre él, lo que indica que este personaje solicita nuestra ayuda para realizar una misión.



Figura 49: NPC inicio de misión.

Una vez hablamos con el personaje se mostrará un pequeño diálogo (1) donde se indicará el objeto u objetos necesarios para poder completar la receta. Asimismo, aparecerá a la izquierda de la pantalla la misión (2) con detalles de esta para poder ser finalizada, donde se puede observar las cantidades que se poseen y las necesarias totales, donde las que se poseen se incrementan conforme sean recolectadas.



Figura 50: Diálogo y detalles de la receta.

Una vez recolectados todos los ingredientes, se muestra un mensaje de la aldeana que nos ha pedido ayuda, además de actualizarse el menú de recetas con un tic verde indicando que se tienen los ingredientes necesarios. En este momento se entregan al NPC para finalizar la misión.



Figura 51: Misión finalizada.

Durante el transcurso de las misiones se deberán de recoger los distintos ingredientes con las herramientas correctas, para ello se deben seleccionar de la barra de herramientas que se encuentra en la parte inferior y clicar sobre la herramienta indicada para seleccionarla. Si se intenta recoger un ingrediente con la herramienta equivocada, el cursor se mostrará de color gris y no se podrá realizar la acción, sin embargo, si seleccionamos la herramienta correcta se mostrará el cursor a color, lo que indica que se puede realizar la acción.



Figura 52: Selección de herramienta correcta vs. incorrecta

Asimismo, hay ciertas zonas que se encuentran inaccesibles, para acceder es necesario abrirse paso cortando lo que pueda bloquear el paso con el hacha y poder realizar la acción de talar.



Figura 53: Herramienta para talar

Se dispone de la herramienta Point para mostrar el nombre del objeto que hemos seleccionado haciendo clic sobre este, de esta forma se facilita localizar el objeto adecuado. En la siguiente imagen se puede observar que, una vez clicado el objeto, en la zona de diálogo se muestra el nombre de este.



Figura 54: Función de la herramienta Point.

Finalmente, una vez se hayan completado todas las misiones requeridas por los habitantes de la isla, se mostrará un pequeño diálogo de agradecimiento y una pantalla de fin de nivel.



Figura 55: Pantalla fin de partida.

7. Conclusiones y líneas de futuro

7.1. Repercusiones Ético-Sociales de los videojuegos en el aula

De acuerdo a las conclusiones extraídas del cuaderno número siete de Cuadernos de Ética en Clave Cotidiana sobre Videojuegos, gamificación y reflexiones éticas [42], se concluye que los videojuegos y la gamificación ofrecen oportunidades educativas, con juegos didácticos en evolución y experiencias exitosas en centros educativos, lo que indica que pueden ser una alternativa eficaz a la educación tradicional, especialmente para alumnos con dificultades de aprendizaje o inclusión.

La ética en videojuegos y gamificación es crucial para mantener su éxito humano y enfatizan la importancia de promover valores positivos, evitando la deshumanización y fomentando la solidaridad y el apoyo mutuo.

Asimismo, en el artículo Los videojuegos en la educación [43], se lanzan preguntas como: ¿Qué papel pueden desempeñar los videojuegos en las nuevas estrategias de formación?, ¿Se pueden mantener las mismas metodologías docentes utilizadas tradicionalmente, para alcanzar las exigencias del actual sistema educativo?

Entre otras conclusiones se destaca que los juegos digitales son capaces de incorporar experiencias y situaciones complejas en la formación; esta herramienta permite a profesores y estudiantes diseñar experiencias de aprendizaje que van más allá de los contenidos tradicionales, y el aprendizaje a través del juego se destaca como beneficioso, ya que proporciona un entorno estimulante donde los estudiantes pueden cometer errores y aprender mediante la práctica.

Estas conclusiones se extraen de acuerdo a lo que la profesora la profesora Begoña Gros denomina “videojuegos serios”, es decir, videojuegos desarrollados para la educación, no solamente lúdicos, ya que los videojuegos también pueden convertirse en elementos que impulsen la violencia, tanto o más que el cine y la televisión; se tiene que hablar siempre dentro de un entorno controlado en el aula entre docente y alumnado.

7.2. Conclusiones

Durante la realización de este videojuego como herramienta educativa me ha servido para adquirir una visión completa de la complejidad que es el desarrollo en solitario de un videojuego y la integración con el entorno educativo. Una clara planificación inicial ha sido crucial para el desarrollo y llegar a los objetivos planteados con éxito.

Asimismo, se ha sido bastante conservador a la hora de elegir implementar muchas funcionalidades que poseen la mayoría de los videojuegos, priorizando la finalización de un prototipo jugable y robusto con menos funcionalidades; esto ha sido acertado puesto que la carga de trabajo asumida para este prototipo ha sido mayor de la esperada, y posiblemente no podrían haberse implementado.

Hasta el momento los conocimientos del autor de este TFG en el desarrollo de videojuegos se limitaban al visionado de vídeos sobre las bondades de cada uno de los motores con ejemplos aislados de las capacidades e implementaciones de distintas funcionalidades; orquestarlas todas para que funcionen como un todo implican una complejidad mucho mayor de la esperada inicialmente.

El diseño centrado en el usuario final también se ha mantenido como un principio fundamental, comprendiendo sus necesidades educativas y adaptando la jugabilidad en consecuencia, ha resultado esencial para crear una experiencia educativa y también ha sido un punto de complejidad añadido.

La metodología Ágil y su enfoque iterativo y flexible, se ha convertido en un aliado clave ya que se ha requerido una adaptabilidad constante para abordar todas las fases del desarrollo, desde del diseño de nivel, arquitectura del videojuego, búsqueda de *assets*, hasta la implementación, enfocándose en maximizar la eficiencia y responder a cambios inesperados.

La conclusión más relevante bajo el punto de vista del autor del TFG, es la planificación que, aunque se ha estudiado y desarrollado previamente, muchos de los aspectos del videojuego en ocasiones se han tenido que replantear tras investigar, modificar e iterar en varias ocasiones para obtener un producto viable.

Finalmente, el desarrollo del TFG y la memoria han servido para una primera toma de contacto con el mundo del desarrollo de videojuegos muy gratificante; ha terminado siendo una gran fuente de información adquirida a través de toda la investigación realizada para futuros proyectos personales.

7.3. Líneas de futuro

La necesidad de buscar retroalimentación externa ha sido prioritaria en las fases finales del desarrollo del prototipo, y la interacción con educadores ha proporcionado nuevas y valiosas perspectivas para ajustar y mejorar la experiencia de juego educativa, así como mejoras para el futuro como, por ejemplo:

- **Adaptar el nivel de lenguaje:** de acuerdo a las aportaciones de los docentes entrevistados, destacan que, para ciertos centros, como por ejemplo los Bilingües, este prototipo es muy sencillo, por lo que es interesante implementar una selección de nivel más avanzado.
- **Crear un nivel demostración:** Se ha destacado que para que los alumnos se puedan familiarizar con los controles y saber la finalidad del videojuego sería muy valioso una explicación previa, por lo que un nivel demostración será una opción a contemplar.
- **Implementar *quests* más diversas:** Añadir distintos tipos de desafíos que puedan ampliar la jugabilidad y mantener la atención del jugador a largo plazo.
- **Diseñar más niveles:** la duración del prototipo es corta ya que ha sido concebido para ser una primera aproximación de las posibilidades de esta herramienta en el entorno docente; en el futuro distintos niveles se estiman necesarios para poder mejorar la inmersión y jugabilidad de los alumnos.
- **Crear un sistema de guardado de niveles:** Ante la implementación de más niveles de progresión se hace necesario un sistema de guardado que permita a los jugadores avanzar y retomar en la siguiente sesión de juego en el aula; hay que recordar que como herramienta educativa el tiempo de juego en el aula es limitado.

- **Mejoras de Accesibilidad:** Se añadirá una herramienta de Text To Speech (TTS) para jugadores con deficiencias visuales que pueda reproducir mediante audio texto mostrado por pantalla de forma automática. En el momento de la realización de este TFG no ha dado tiempo a implementar una herramienta de estas características debido a la complejidad y la falta de herramientas gratuitas; las de pago tenían un coste de licencia muy elevado.

Bibliografía

1. Unity Technologies. Unity [Internet]. Unity Technologies. [citado 1 de octubre de 2023]. Disponible en: <https://unity.com/es>
2. Epic Games inc. Unreal Engine [Internet]. Epic Games inc. [citado 1 de octubre de 2023]. Disponible en: <https://www.unrealengine.com/es-ES>
3. Moreno F. Teoría de la instrucción vs. teoría del aprendizaje significativo: contraste entre J. Bruner y D. Ausubel [Internet]. El Cid Editor | apuntes; 2009. p. 18. Disponible en: <https://elibro.net/es/lc/uoc/titulos/29632>
4. Juan Rubio AD, García Conesa IM. Características relevantes para el aprendizaje efectivo de una lengua extranjera. 2012 [citado 23 de octubre de 2023]; Disponible en: <http://hdl.handle.net/10317/2017>
5. Valve Corporation. Atelier Arland Series [Internet]. Valve Corporation. [citado 24 de octubre de 2023]. Disponible en: https://store.steampowered.com/bundle/9121/Atelier_Arland_series_Deluxe_Pack/
6. Borondo S. Análisis de Harvest Moon: Light of Hope (PS4, PC, Xbox One, Switch) [Internet]. Vandal. 2020 [citado 23 de octubre de 2023]. Disponible en: <https://vandal.elespanol.com/analisis/ps4/harvest-moon-light-of-hope/48435#p-73>
7. ConcernedApe LLC. Stardew Valley [Internet]. Atelier Arland Series. [citado 23 de octubre de 2023]. Disponible en: <https://www.stardewvalley.net/>
8. Unity Technologies. Unity User Manual 2022.3 (LTS) [Internet]. Unity Technologies. Unity Technologies; 2022 [citado 1 de octubre de 2023]. Disponible en: <https://docs.unity3d.com/Manual/index.html>
9. Wikipedia. GitHub [Internet]. [citado 23 de octubre de 2023]. Disponible en: <https://en.wikipedia.org/wiki/GitHub>
10. GitHub Inc. Github Desktop [Internet]. [citado 23 de octubre de 2023]. Disponible en: <https://docs.github.com/es/desktop/overview/about-github-desktop>
11. Cave Story Wiki Español. Cave Story [Internet]. [citado 26 de octubre de 2023]. Disponible en: https://cavestory.fandom.com/es/wiki/Cave_Story
12. Nafría R. Daisuke Amaya [Internet]. Vandal. 2020 [citado 26 de octubre de 2023]. Disponible en: <https://vandal.elespanol.com/indispensable/daisuke-amaya>
13. Dukope. Papers, please [Internet]. TIGSource. 2012 [citado 26 de octubre de 2023]. Disponible en: <https://forums.tigsource.com/index.php?topic=29750>
14. ABC News. ABC News [Internet]. 2013 [citado 26 de octubre de 2023]. Disponible en: https://abcnews.go.com/ABC_Univision/papers-game-surprise/story?id=19135556
15. Pope D. TIGSource - November 2012 [Internet]. 2012 [citado 26 de octubre de 2023]. Disponible en: <https://dukope.com/devlogs/papers-please/tig-00/>

16. Pope D. Cramming «Papers, Please» Onto Phones [Internet]. 2022 [citado 26 de octubre de 2023]. Disponible en: <https://dukope.com/devlogs/papers-please/mobile/>
17. Steam [Internet]. [citado 26 de octubre de 2023]. Disponible en: <https://store.steampowered.com/news/app/239030/view/3483000339649202293>
18. Cullen J. Papers, Please creator wants to do PS Vita version [Internet]. VG247. 2014 [citado 26 de octubre de 2023]. Disponible en: <https://www.vg247.com/papers-please-creator-wants-to-do-ps-vita-version>
19. Lin A. Stardew Valley: Pushing The Boundaries of Farming RPGs [Internet]. The Cornell Daily Sun. 2016 [citado 4 de noviembre de 2023]. Disponible en: <https://cornellsun.com/2016/02/23/stardew-valley-pushing-the-boundaries-of-farming-rpgs/>
20. Baker C. The 4 years of self-imposed crunch that went into Stardew Valley [Internet]. Game Developer. 2016 [citado 4 de noviembre de 2023]. Disponible en: <https://www.gamedeveloper.com/business/the-4-years-of-self-imposed-crunch-that-went-into-i-stardew-valley-i->
21. Marks T. Interview: What's next for Stardew Valley [Internet]. PCGAMER. 2016 [citado 4 de noviembre de 2023]. Disponible en: <https://www.pcgamer.com/stardew-valley-interview/>
22. Griffin SM. Sea of Stars review – like a lost, late classic of the Super Nintendo era [Internet]. The Guardian. 2023 [citado 4 de noviembre de 2023]. Disponible en: <https://www.theguardian.com/games/2023/aug/31/sea-of-stars-review-like-a-lost-late-classic-of-the-super-nintendo-era>
23. Padilla del Valle B. «Blasphemous»: culto al barroco pixelado [Internet]. MERCURIO CULTURA DESORBITADA. 2020 [citado 4 de noviembre de 2023]. Disponible en: <https://www.revistamercurio.es/2020/01/14/blasphemous-culto-al-barroco-pixelado/>
24. Wikipedia. Blasphemous (video game) [Internet]. Wikipedia. 2023 [citado 4 de noviembre de 2023]. Disponible en: [https://en.wikipedia.org/wiki/Blasphemous_\(video_game\)](https://en.wikipedia.org/wiki/Blasphemous_(video_game))
25. Contingent99 LLC. Wizard Of Legend [Internet]. Wizard Of Legend. 2019 [citado 4 de noviembre de 2023]. Disponible en: <https://wizardoflegend.com/>
26. Valera R. Análisis de Wizard of Legend (PC, Switch, Xbox One, PS4) [Internet]. Vandal Online. 2018 [citado 4 de noviembre de 2023]. Disponible en: <https://vandal.elespanol.com/analisis/pc/wizard-of-legend/37085#p-13>
27. Johnson E. Know Your Market: Making Indie Games That Sell [Internet]. GAME DEVELOPERS CONFERENCE. 2018 [citado 5 de noviembre de 2023]. Disponible en: <https://www.youtube.com/watch?v=uyODfr-mnUY>
28. Rougal. Tell me about your failed game [Internet]. Reddit. 2021 [citado 6 de noviembre de 2023]. Disponible en: https://www.reddit.com/r/gamedev/comments/pyglcs/tell_me_about_your_failed_game/
29. Steam. Blackshift [Internet]. Steam. 2020 [citado 6 de noviembre de 2023]. Disponible en: <https://store.steampowered.com/app/741110/Blackshift/>

30. Harrys C. Harvest Moon: Friends of Mineral Town [Internet]. IGN. 2003 [citado 6 de noviembre de 2023]. Disponible en:
<https://www.ign.com/articles/2003/11/25/harvest-moon-friends-of-mineral-town-2>
31. Mackey B. Virtual Spotlight: Harvest Moon Hits a High Point with Friends of Mineral Town [Internet]. VG247. 2015 [citado 6 de noviembre de 2023]. Disponible en:
<https://www.vg247.com/virtual-spotlight-harvest-moon-hits-a-high-point-with-friends-of-mineral-town>
32. Itch.io. Itch.io [Internet]. Itch.io. [citado 8 de noviembre de 2023]. Disponible en: Itch.io
33. Cuesta L. El móvil en el aula: ¿herramienta o distracción? [Internet]. JUNIOR REPORT. 2023 [citado 8 de noviembre de 2023]. Disponible en:
<https://junior-report.media/el-movil-en-el-aula-herramienta-o-distraccion/>
34. INTEF. Internet en el Aula, red social docente para una educación en el siglo XXI [Internet]. INTEF. [citado 8 de noviembre de 2023]. Disponible en: <http://internetaula.ning.com/>
35. TodoELE. TodoELE [Internet]. TodoELE. [citado 8 de noviembre de 2023]. Disponible en:
<https://www.todoele.net/comunidad>
36. Unity Technologies. Unity Manual [Internet]. Unity. 2023 [citado 9 de diciembre de 2023]. Disponible en: <https://docs.unity3d.com/Manual/system-requirements.html>
37. Mikael Kalms. CÓMO CONSTRUIR EL CÓDIGO CONFORME CRECE TU PROYECTO [Internet]. Unity Technologies. 2019 [citado 10 de diciembre de 2023]. Disponible en:
<https://unity.com/es/how-to/how-architect-code-your-project-scales>
38. Ryan Hipple. Unite Austin 2017 - Game Architecture with Scriptable Objects [Internet]. Unity Technologies. 2017 [citado 10 de diciembre de 2023]. Disponible en:
https://www.youtube.com/watch?v=raQ3iHhE_Kk
39. Richard Fine. Unite 2016 - Overthrowing the MonoBehaviour Tyranny in a Glorious Scriptable Object Revolution [Internet]. Unity Technologies. 2016 [citado 10 de diciembre de 2023]. Disponible en: <https://www.youtube.com/watch?v=6vmRwLYWNRo>
40. Unity Technologies. MonoBehaviour [Internet]. Unity. 2023 [citado 10 de diciembre de 2023]. Disponible en: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>
41. Diana Silva. La mitad de los institutos catalanes regulan ya el uso del móvil [Internet]. Ara. 2023 [citado 12 de diciembre de 2023]. Disponible en:
https://es.ara.cat/sociedad/educacion/mitad-institutos-catalanes-regulan-movil_1_4851190.html
42. Martín M, Vílchez LF. Videojuegos, gamificación y reflexiones éticas. Cuadernos de ética en clave cotidiana [Internet]. 2017 [citado 27 de diciembre de 2023];7:63-5. Disponible en:
<https://funderetica.org/wp-content/uploads/2017/01/Cuaderno-7-web-def.pdf>
43. Peris FJS, Horacek GAE. Los videojuegos en la educación. Aularia: Revista Digital de Comunicación [Internet]. 2014 [citado 27 de diciembre de 2023];3(1):21-6. Disponible en:
https://rabida.uhu.es/dspace/bitstream/handle/10272/9259/Los_videojuegos.pdf?sequence=2

Anexos

Anexo A: Glosario

Asset: Elemento o recurso utilizado en el desarrollo del juego, como gráficos, sonidos ...

Casual gaming: Modalidad de juego dirigida a un público no especializado, con partidas cortas y mecánicas sencillas.

Collider2D: Componentes que definen las áreas de interacción en juegos 2D.

Deuteranopia: Tipo de daltonismo que afecta la percepción del color verde.

Discromatopsia: Trastorno visual que afecta la percepción del color.

Top-Down: Perspectiva de juego en la que la cámara observa desde arriba.

Frame: Un solo cuadro de animación en un videojuego.

Framework: Estructura o marco de trabajo para el desarrollo de software.

Gameplay: Experiencia de juego, incluyendo mecánicas, interactividad y dinámicas.

Gamificación: Integración de elementos de juego en contextos no lúdicos para motivar o educar.

GScript: Lenguaje de programación utilizado en el motor de juegos Godot.

Hack and slash: Género de juegos de acción caracterizado por el combate intenso.

IDE: Entorno de Desarrollo Integrado, herramienta para programar.

Indie: Juego desarrollado de forma independiente, sin el respaldo de grandes empresas.

JRPG: Juego de Rol Japonés (Japanese Role Playing Game).

Ágil: Enfoque de desarrollo que prioriza la flexibilidad y la colaboración.

NPC: Personaje no jugable (Non Playable Character).

Onedev: Desarrollo de videojuegos por una sola persona.

Pixel art: Estilo gráfico que utiliza píxeles para crear imágenes.

Protanopia: Tipo de daltonismo que afecta la percepción del color rojo.

Quest: Misión o tarea a cumplir en el juego.

Raycast2D: Método de detección de colisiones en juegos 2D para determinar qué objetos o colisionadores están en la trayectoria de un rayo en un entorno bidimensional.

Shader: Programa que define la apariencia visual de los objetos en un juego.

Slider: Elemento de la interfaz gráfica para seleccionar valores representado por una barra y un control deslizante.

Smartphones: Teléfonos inteligentes, plataforma común para juegos móviles.

Sprite: Gráfico 2D utilizado para representar objetos o personajes.

Steam Greenlight: Antiguo programa de Steam para votar juegos independientes.

Tablets: Dispositivos móviles con pantalla táctil.

Text To Speech: Conversión de texto a voz.

Tiles: Pequeñas imágenes utilizadas para crear mapas/escenas en el juego.

Trackpad: Superficie táctil en dispositivos de entrada.

Triple A: Juegos de alta calidad y presupuesto.

Tritanopia: Tipo de daltonismo que afecta la percepción del color azul.

User Interface: Interfaz de usuario, elementos visuales y de interacción en el juego.

Anexo B: Entregables del proyecto

[Github.](#)

[Itch.io.](#)

Anexo C: Resultados detallados de la encuesta

TFG Omar Cester

Cuestionario viabilidad prototipo videojuego educativo

The respondent's email (jsolalalin@educacion.navarra.es) was recorded on submission of this form.

Nombre y apellidos *

Jose Miguel Sola Lalinde

Titulación *

Maestro de Educación Primaria

Años de experiencia *

5

Rango de edad de los niños con los que trabajas

	Edad
7 años	<input type="checkbox"/>
8 años	<input type="checkbox"/>
9 años	<input type="checkbox"/>
10 años	<input checked="" type="checkbox"/>
11 años	<input type="checkbox"/>

¿Crees que el videojuego puede lograr captar la atención de los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Consideras que los objetivos educativos del juego están claramente definidos? *

- Sí
- No
- Parcialmente

¿Encuentras el aspecto gráfico del juego fácil de entender para los niños del rango de edad indicada? *

- Sí
- No
- Parcialmente

¿Crees que el contenido del juego es adecuado en términos de dificultad para los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Te parece que el juego ofrece suficiente variedad para mantener el interés de los niños a lo largo del tiempo? *

- Sí
- No
- Parcialmente

En términos educativos, ¿Qué aspectos del juego crees que podrían ser mejorados para adaptarse a las necesidades de los niños de esta franja de edad? *

La idea del juego es muy buena pero estos niños y niñas llevan mucho tiempo estudiando inglés y puede que sea demasiado básico pero si que nos pueden venir bien para aquellos alumnos y alumnas con más dificultades.

Política de privacidad

Autorizo voluntariamente a Omar Cester Gracia, en su calidad de estudiante de Grado de Ingeniería Informática, a recopilar y procesar los siguientes datos personales con fines educativos y de investigación:

1. Nombre y Apellidos.
2. Titulación o Profesión.
3. Años de Experiencia.
4. Edad de los Niños con los que Trabaja.
5. Email.

Entiendo que la recopilación de estos datos se realizará exclusivamente con el propósito de obtener retroalimentación sobre un videojuego educativo desarrollado por Omar Cester Gracia. Se garantiza la confidencialidad y seguridad de la información recopilada, y los datos no serán utilizados con fines distintos a los mencionados anteriormente. Esta autorización es voluntaria, y en cualquier momento tengo el derecho de retirar mi consentimiento sin consecuencias negativas.

Protección de datos *

- He leído y acepto la política de privacidad

This form was created inside of Fundació per a la Universitat Oberta de Catalunya.

Google Forms

TFG Omar Cester

Cuestionario viabilidad prototipo videojuego educativo

The respondent's email (mestradmed@educacion.navarra.es) was recorded on submission of this form.

Nombre y apellidos *

MAMEN ESTRADA MEDINA

Titulación *

MAESTRA PRIMARIA

Años de experiencia *

5

Rango de edad de los niños con los que trabajas

	Edad
7 años	<input checked="" type="checkbox"/>
8 años	<input type="checkbox"/>
9 años	<input type="checkbox"/>
10 años	<input type="checkbox"/>
11 años	<input type="checkbox"/>

¿Crees que el videojuego puede lograr captar la atención de los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Consideras que los objetivos educativos del juego están claramente definidos? *

- Sí
- No
- Parcialmente

¿Encuentras el aspecto gráfico del juego fácil de entender para los niños del rango de edad indicada? *

- Sí
- No
- Parcialmente

¿Crees que el contenido del juego es adecuado en términos de dificultad para los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Te parece que el juego ofrece suficiente variedad para mantener el interés de los niños a lo largo del tiempo? *

- Sí
- No
- Parcialmente

En términos educativos, ¿Qué aspectos del juego crees que podrían ser mejorados para adaptarse a las necesidades de los niños de esta franja de edad? *

Esta franja de edad está en proceso de adquisición de la lectoescritura. Tampoco han desarrollado en su totalidad la competencia digital. Por ello pienso que sería más apropiado para otro rango de edad superior.

Política de privacidad

Autorizo voluntariamente a Omar Cester Gracia, en su calidad de estudiante de Grado de Ingeniería Informática, a recopilar y procesar los siguientes datos personales con fines educativos y de investigación:

1. Nombre y Apellidos.
2. Titulación o Profesión.
3. Años de Experiencia.
4. Edad de los Niños con los que Trabaja.
5. Email.

Entiendo que la recopilación de estos datos se realizará exclusivamente con el propósito de obtener retroalimentación sobre un videojuego educativo desarrollado por Omar Cester Gracia. Se garantiza la confidencialidad y seguridad de la información recopilada, y los datos no serán utilizados con fines distintos a los mencionados anteriormente. Esta autorización es voluntaria, y en cualquier momento tengo el derecho de retirar mi consentimiento sin consecuencias negativas.

Protección de datos *

- He leído y acepto la política de privacidad

This form was created inside of Fundació per a la Universitat Oberta de Catalunya.

Google Forms

TFG Omar Cester

Cuestionario viabilidad prototipo videojuego educativo

The respondent's email (eserranz@educacion.navarra.es) was recorded on submission of this form.

Nombre y apellidos *

Elena Serrano Zueco

Titulación *

Maestra Lengua Extranjera

Años de experiencia *

20

Rango de edad de los niños con los que trabajas

	Edad
7 años	<input type="checkbox"/>
8 años	<input checked="" type="checkbox"/>
9 años	<input type="checkbox"/>
10 años	<input type="checkbox"/>
11 años	<input type="checkbox"/>

¿Crees que el videojuego puede lograr captar la atención de los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Consideras que los objetivos educativos del juego están claramente definidos? *

- Sí
- No
- Parcialmente

¿Encuentras el aspecto gráfico del juego fácil de entender para los niños del rango de edad indicada? *

- Sí
- No
- Parcialmente

¿Crees que el contenido del juego es adecuado en términos de dificultad para los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Te parece que el juego ofrece suficiente variedad para mantener el interés de los niños a lo largo del tiempo? *

- Sí
- No
- Parcialmente

En términos educativos, ¿Qué aspectos del juego crees que podrían ser mejorados para adaptarse a las necesidades de los niños de esta franja de edad? *

En primer lugar cuando lo conocí me pareció que presentaba cierta dificultad para mi alumnado , pero al llevarlo a la práctica con , me he dado cuenta que llamaba su atención, despertaba su motivación y "un pequeño enganche"

Política de privacidad

Autorizo voluntariamente a Omar Cester Gracia, en su calidad de estudiante de Grado de Ingeniería Informática, a recopilar y procesar los siguientes datos personales con fines educativos y de investigación:

1. Nombre y Apellidos.
2. Titulación o Profesión.
3. Años de Experiencia.
4. Edad de los Niños con los que Trabaja.
5. Email.

Entiendo que la recopilación de estos datos se realizará exclusivamente con el propósito de obtener retroalimentación sobre un videojuego educativo desarrollado por Omar Cester Gracia. Se garantiza la confidencialidad y seguridad de la información recopilada, y los datos no serán utilizados con fines distintos a los mencionados anteriormente. Esta autorización es voluntaria, y en cualquier momento tengo el derecho de retirar mi consentimiento sin consecuencias negativas.

Protección de datos *

- He leído y acepto la política de privacidad

This form was created inside of Fundació per a la Universitat Oberta de Catalunya.

Google Forms

TFG Omar Cester

Cuestionario viabilidad prototipo videojuego educativo

The respondent's email (mserrand@educacion.navarra.es) was recorded on submission of this form.

Nombre y apellidos *

Marta Serrano Duaso

Titulación *

Maestra Lengua Extranjera

Años de experiencia *

15

Rango de edad de los niños con los que trabajas

	Edad
7 años	<input type="checkbox"/>
8 años	<input checked="" type="checkbox"/>
9 años	<input type="checkbox"/>
10 años	<input type="checkbox"/>
11 años	<input type="checkbox"/>

¿Crees que el videojuego puede lograr captar la atención de los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Consideras que los objetivos educativos del juego están claramente definidos? *

- Sí
- No
- Parcialmente

¿Encuentras el aspecto gráfico del juego fácil de entender para los niños del rango de edad indicada? *

- Sí
- No
- Parcialmente

¿Crees que el contenido del juego es adecuado en términos de dificultad para los niños de 7 a 11 años? *

- Sí
- No
- Parcialmente

¿Te parece que el juego ofrece suficiente variedad para mantener el interés de los niños a lo largo del tiempo? *

- Sí
- No
- Parcialmente

En términos educativos, ¿Qué aspectos del juego crees que podrían ser mejorados para adaptarse a las necesidades de los niños de esta franja de edad? *

El video juego mantiene la atención del alumnado. Ayuda a desarrollar la competencia digital ya que los juegos que solemos practicar en clase requieren procedimientos más sencillos. Si que veo necesario una previa explicación antes de empezar el juego: las distintas herramientas, el teclado...etc.

Un aspecto positivo es que para saber de qué trata la misión es necesario usar la comprensión lectora y entender bien el objetivo.

Quizás para el alumnado de bilingüe sea un poco básico, aunque nunca viene mal repasar los conceptos adquiridos. Para este alumnado, el feedback podría ser en inglés y se adecuaría mejor a su nivel de competencia.

También pienso que puede ser apropiado como un recurso más para atender a la diversidad.

Política de privacidad

Autorizo voluntariamente a Omar Cester Gracia, en su calidad de estudiante de Grado de Ingeniería Informática, a recopilar y procesar los siguientes datos personales con fines educativos y de investigación:

1. Nombre y Apellidos.
2. Titulación o Profesión.
3. Años de Experiencia.
4. Edad de los Niños con los que Trabaja.
5. Email.

Entiendo que la recopilación de estos datos se realizará exclusivamente con el propósito de obtener retroalimentación sobre un videojuego educativo desarrollado por Omar Cester Gracia. Se garantiza la confidencialidad y seguridad de la información recopilada, y los datos no serán utilizados con fines distintos a los mencionados anteriormente. Esta autorización es voluntaria, y en cualquier momento tengo el derecho de retirar mi consentimiento sin consecuencias negativas.

Protección de datos *

He leído y acepto la política de privacidad

This form was created inside of Fundació per a la Universitat Oberta de Catalunya.

Google Forms