



**Universidad
Europea**

UNIVERSIDAD EUROPEA DE MADRID
ESCUELA DE ARQUITECTURA INGENIERIA Y DISEÑO

Voxel Knights

Arturo Rodríguez Severino

TRABAJO FINAL DEL GRADO DE DISEÑO DE VIDEOJUEGOS

Dirigido por María Teresa Barranco Crespo

Convocatoria de Junio de 2020/2021

Voxel Knights, Arturo Rodríguez Severino

Deja esta página en blanco. Es la “guarda”, sirve para proteger el documento y que el tribunal haga anotaciones.

Voxel Knights, Arturo Rodríguez Severino

UNIVERSIDAD EUROPEA DE MADRID
ESCUELA DE ARQUITECTURA INGENIERIA Y DISEÑO

Voxel Knights

Arturo Rodríguez Severino

PROYECTO FINAL

DEL GRADO DE DISEÑO DE VIDEOJUEGOS

Dirigido por María Teresa Barranco Crespo

Convocatoria de Junio 2020/2021

Voxel Knights, Arturo Rodríguez Severino

Agradecimientos

Especial agradecimiento a mis tutores durante la realización del proyecto, María Teresa Barranco Crespo y Guillermo Castilla Cebrián.

También agradecer la ayuda a todas las personas que han probado el proyecto y han dado su opinión, ayudando a la mejora del mismo.

Índice

Resumen	10
Abstract.....	11
Capítulo 1. Introducción, descripción y objetivos.....	12
1.1 Introducción	12
1.2 Descripción general del proyecto.....	12
1.3 Objetivo general.....	12
1.4 Objetivos específicos	12
Capítulo 2. Marco conceptual y contextual.....	13
Capítulo 3. Arquitectura del videojuego.	15
3.1 Análisis de requerimientos.....	15
3.2 Implementación	15
Capítulo 4. Metodología y organización	16
4.1 Diagrama de GANTT	16
4.2 Análisis DAFO	17
Capítulo 5. Plan de Marketing y distribución.....	17
5.1 Estudio del mercado.....	17
5.2 Previsión de ventas	18
5.3 Distribución	19
5.4 Previsión de costes.....	19
5.4.1 Equipos para procesos de información.....	19
5.4.2 Los gastos	20
Capítulo 6. Desarrollo del proyecto.....	20
6.1 Preproducción	20
6.1.1 Diseño del juego.....	21
6.1.2 Diseño de personajes (concept).....	43
6.1.3 Diseños narrativos	44

6.2 Arte	45
6.2.1 Modelado 3D y texturizado.....	45
6.2.2 Rigging de personajes.	52
6.2.3 Animación de personajes	53
6.2.4 UX e UI.....	54
6.3 Programación	62
6.4 Postproducción.....	65
6.4.1 Ambientación e iluminación.	65
6.4.2 Sonido.....	65
Conclusiones.....	65
Glosario	66
Bibliografía.....	66
Anexo I. Índice de figuras.	67
Anexo II. Actividades realizadas durante la investigación.....	69
Anexo III. Requerimientos de licencia de música.	70

Resumen

Voxel Knights es un juego de estrategia en tablero, con vista isométrica y con elementos de *roguelike*¹. El jugador deberá dirigir a uno o varios caballeros para lograr el objetivo establecido de cada nivel, el cual gira en torno a derrotar o sobrevivir contra un equipo enemigo formado por esqueletos.

Como su nombre indica, el juego usa una estética *voxel art*, la cual complementa a la rigidez de un juego de tablero.

La progresión de los niveles tiene un pequeño grado de aleatoriedad, tanto en el nivel al que se avanza como la progresión de los soldados. Esto hace que en cada partida que se juega no sólo los niveles que se encontrarán serán diferentes, sino que la resolución de los mismos variará en función de que características hayan sido otorgadas al jugador.

Esta aleatoriedad pretende fomentar tanto la rejugabilidad como la capacidad de adaptación del jugador, otorgando una experiencia diferente en cada partida.

Este proyecto es una demostración limitada del juego, compuesta tanto por el tutorial del mismo para mostrar las mecánicas básicas, como por el número necesario de niveles para poder ofrecer la experiencia *roguelike*.

La demostración ha sido desarrollada en la versión 2019.4.18f1 del motor de videojuegos *Unity* y todos los modelos 3D han sido modelados en el programa de edición de voxels, *Qubicle*. Las animaciones y “*rigging*”² de personajes han sido realizados en el programa de desarrollo de gráficos 3D *Autodesk Maya* en la versión 2020, y algunas animaciones de efectos menores han sido creadas en *Unity*. Los sonidos, debido al bajo nivel de edición requerida, han sido modificados en el editor de vídeo *Adobe Premiere*.

Palabras clave

Videojuego, voxel art, estrategia, roguelike.

¹ Los videojuegos de mazmorras (conocidos comúnmente como *roguelike*), son un subgénero de los videojuegos de rol que se caracterizan por una aventura a través de niveles y desafíos generados aleatoriamente. Este género nació a partir del juego homónimo *Rogue* (Toy, Wichman, & Arnold, 1980).

² El *rigging* hace referencia a todos los procesos de preparación necesarios para poder animar un modelo 3D utilizando animación con huesos.

Abstract

Voxel Knights is an isometric tabletop strategy game with roguelike elements. The player must lead one or more knights to achieve the established objective of each level, which revolves around defeating or surviving against an enemy team made up of skeletons.

As the name suggests, the game uses a *voxel art* aesthetic, which complements the rigidity of a tabletop game.

The progression of the levels has a small degree of randomness, both in the level to which it is advanced and the progression of the soldiers. This means that in each game that is played not only the levels that will be found will be different, but their resolution will vary depending on what characteristics have been granted to the player.

This randomness is intended to promote both replayability and the player's ability to adapt, granting a different experience in each game.

This project is a limited demo of the game, consisting of both the tutorial to show the basic mechanics, and the necessary number of levels to be able to offer the roguelike experience.

The demo has been developed in version 2019.4.18f1 of the *Unity* videogame engine. All 3D models have been modeled in the voxel editing program, *Qubicle*. Character animations and rigging have been done in the 3D graphics development program *Autodesk Maya*, in the 2020 version, and some minor effect animations have been created in *Unity*. The sounds, due to the low level of editing required, have been modified in the *Adobe Premiere* video editor.

Keywords

Videogame, voxel art, strategy, roguelike.

Capítulo 1. Introducción, descripción y objetivos.

1.1 Introducción

Voxel Knights se originó a partir de la idea de combinar dos géneros de videojuegos: los videojuegos de estrategia y los *roguelike*. Además, he decidido aplicar mi reciente aprendizaje en el medio artístico del *voxel art*, el cual he aprovechado para crear un juego con una estética no demasiado explorada, aunque con gran éxito en algunos de los juegos que la utilizan.

1.2 Descripción general del proyecto

El proyecto pretende abarcar un total de 15 niveles, los cuales se pueden dividir en: 6 niveles de tutorial, y 9 niveles estándar. Los niveles estándar se alternarán de una manera determinada para ofrecer una experiencia diferente cada partida.

Además, en la demostración se incluirá un menú de niveles denominado “*cheat*”, que permitirá probar los niveles individualmente y con las estadísticas que se quieran. Este menú solo existe como parte de la demostración para facilitar la corrección y el acceso a cada uno de los niveles.

1.3 Objetivo general

El objetivo general es crear un producto en el cual se pueda apreciar una porción de la experiencia que ofrecería el producto completo. El cual pretende fomentar la adaptabilidad del jugador, haciendo que sea capaz de aprender y trabajar con las diferentes mecánicas disponibles, e interactuar con ellas de la manera óptima para lograr el éxito, todo esto sin haber creado una experiencia repetitiva.

1.4 Objetivos específicos

La manera en la que el juego pretende fomentar esta capacidad de adaptación es de varias maneras: La primera es mediante un tutorial que explore las mecánicas base del juego de manera intuitiva y con la menor cantidad de texto posible.

Tras completar el tutorial, el juego ofrecerá la experiencia “estándar” del juego, en la que el jugador tendrá que superar un total de tres niveles. Cada uno de estos niveles es elegido de manera aleatoria, y al completarlos el jugador recibirá una mejora a una de las características base, también elegida de manera aleatoria. Con esto no solo variarán los

niveles en cada partida, sino la manera de afrontar cada uno de ellos, ya que el mismo nivel será diferente dependiendo de las características que se tengan.

Capítulo 2. Marco conceptual y contextual

Debido al empleo de diferentes géneros de videojuegos y al estilo artístico no tan común que se quiere atribuir al proyecto, se pueden encontrar numerosos referentes.

En lo que se refiere al tipo de cámara y la información que se presenta al jugador, los videojuegos que destacan son los juegos de estrategia con tablero y vista isométrica. Algunos de los principales referentes de este estilo son los videojuegos de la saga *XCOM* (Gollop, 1993), o los de la saga *Fire Emblem* (Kaga, 1990).

Otro videojuego de gran importancia para este proyecto, es el reciente *Into the Breach* (Ma & Davis, 2018). De nuevo, es un juego táctico con vista isométrica, pero la simplicidad de las mecánicas y de la interfaz es mucho más cercana a lo que se pretende en *Voxel Knights*.

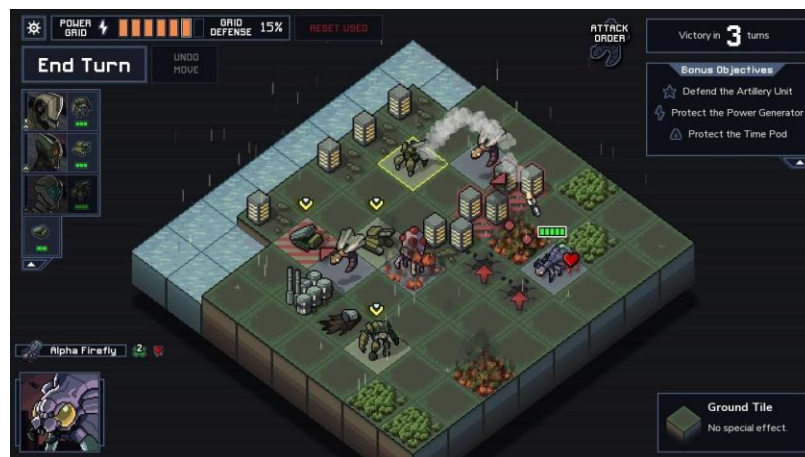


Figura 1. Imagen de gameplay de *Into the Breach* (Ma & Davis, 2018)

La capacidad de utilizar un número de mecánicas muy limitadas, consiguiendo un *gameplay*³ con mucha mayor profundidad, es algo que *Voxel Knights* pretende conseguir. Además, *Into the Breach* ofrece una experiencia *roguelike*, la cual va a ser la principal influyente de este proyecto en ese aspecto.

³ Manera específica en la que un jugador interactúa con el juego, este patrón se define a través de las reglas del juego.

El género *roguelike* ha ganado fuerza en la última década con títulos como *The Binding of Isaac* (McMillen, 2011), con uno de los mayores grados de variabilidad dentro del género, al menos en cuanto a las capacidades del personaje principal.

De todas maneras, el título más destacado en este género hoy en día es el galardonado *Hades* (Rao, 2018), el cual ofrece un gran nivel de rejugabilidad a través del desarrollo narrativo, algo muy poco explorado hasta ahora.

Por último, hay que abordar los principales referentes artísticos, los cuales, de manera obvia son juegos que utilizan el *voxel art* como medio. El primero que viene a la cabeza, es el videojuego más vendido de la historia, *Minecraft*, (Persson, 2011).



Figura 2. Minecraft emplea polígonos simples, detallados a través de la textura.

El estilo de esta obra es, dentro del propio *voxel art*, relativamente diferente a lo que se hará en el trabajo a realizar, ya que la mayor parte del detalle en *Minecraft* viene dado a través de las texturas, mientras que en *Voxel Knights* habrá diseños más suavizados.

El otro gran referente de este proyecto, es *Cube World* (von Funck, 2019), donde el nivel de detalle en lo que a los polígonos se refiere es mucho mayor. En este trabajo, se busca conseguir un estilo intermedio entre este juego y *Minecraft*, donde hay un cierto grado de suavizado en las formas, pero no llevado tan al extremo.

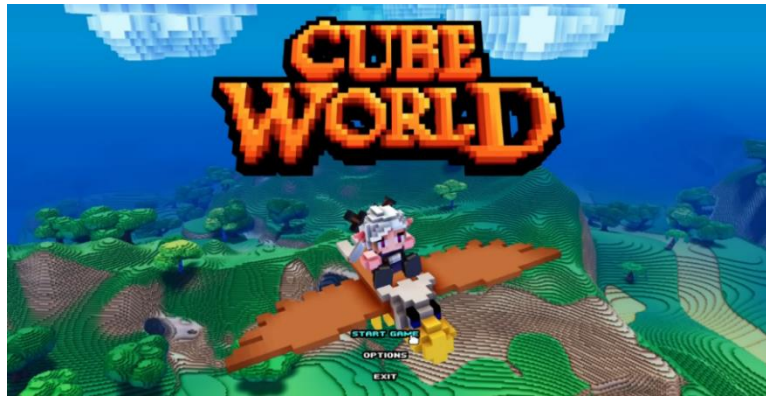


Figura 3. Pantalla de inicio de Cube World.

Capítulo 3. Arquitectura del videojuego.

3.1 Análisis de requerimientos

Los requerimientos mínimos para ejecutar y poder jugar a *Voxel Knights* son los siguientes:

OS: Windows® 10 64 bit, Windows® 8.1 64 bit, Windows® 7 64 bit,

Procesador: 1.1 GHz

Memoria: 2 GB de RAM

Gráficos: Tarjetas capaces de utilizar DX10, DX11 o DX12

Tarjeta de sonido: Estándar.

3.2 Implementación

Se han utilizado una variedad de *plug-ins* para mejorar tanto el resultado, como el proceso de trabajo, siendo la mayoría de ellos propios de *Unity*.

Para mejorar la calidad de los textos, Se ha empleado el paquete *Text Mesh Pro*⁴, el cual aumenta las posibilidades de personalización de estos.

El *plug-in* más importante del trabajo, y que sin embargo no se ve en el resultado final, es el llamado *ProGrids*⁵. Tiene muchas funcionalidades, pero la principal, y la que ha simplificado enormemente el proceso de construcción de escenarios, es la posibilidad de establecer una distancia fija del movimiento de los objetos en el entorno 3D.

Esto quiere decir, que si establecemos esa distancia como 0.5 unidades, cuando movamos un objeto, se moverá en incrementos de 0.5 unidades, ni más ni menos, lo cual,

⁴ <https://docs.unity3d.com/Manual/com.unity.textmeshpro.html>

⁵ <https://docs.unity3d.com/Manual/com.unity.progrids.html>

en un juego de tablero en el que todas las distancias son iguales, es extremadamente importante.

El último paquete utilizado, se mencionó con anterioridad, y no pertenece a *Unity Technologies*, este es, el *Quick Outline*, cuyo funcionamiento y utilidad en el proyecto se establecerán más adelante.

Capítulo 4. Metodología y organización

4.1 Diagrama de GANTT

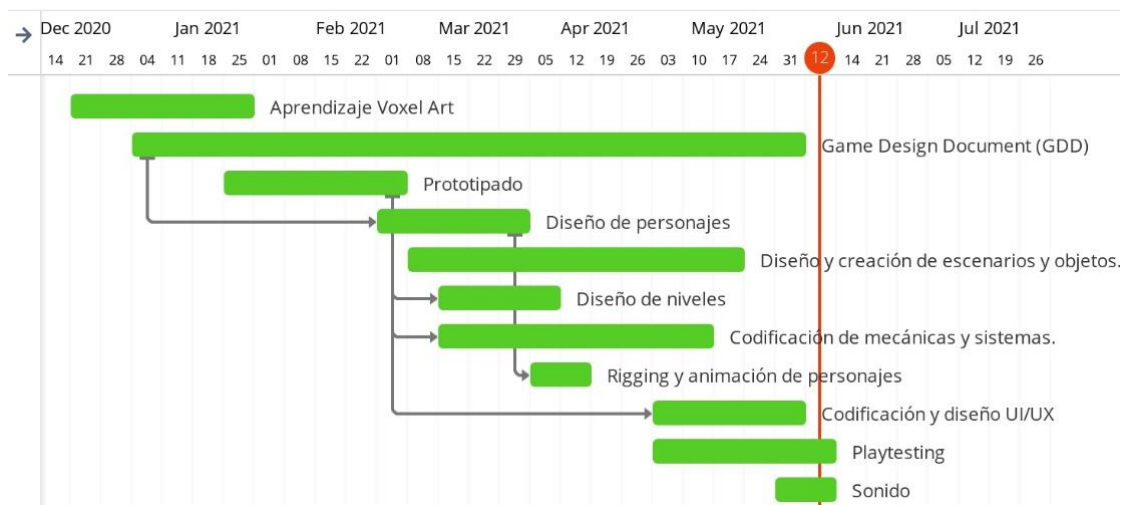


Figura 4. GANTT del desarrollo del proyecto.

Para tener una idea más visual de la partición de las tareas desarrollé un diagrama de Gantt, en el cual puse las metas enfocándome en cumplir una serie de tareas para una fecha específica. De esta manera obtuve una estimación realista del tiempo dedicado a cada una, evitando que alguna tarea no tuviera asignado menos tiempo del necesario para su desarrollo.

4.2 Análisis DAFO



Figura 5. DAFO

Las fuerzas del proyecto vienen dadas por los elementos definitorios del mismo, su diseño tanto a nivel visual como a nivel jugable. Contrastando con esto las debilidades y amenazas principales del mismo son las típicas de juegos independientes, las cuales vienen dadas por la cantidad de presupuesto disponible.

Este presupuesto y cómo repercute en el marketing se estudia en un apartado posterior.

Capítulo 5. Plan de Marketing y distribución

5.1 Estudio del mercado

Para definir el *target*⁶ de *Voxel Knights* hay que hacer dos consideraciones: si bien el público general de los juegos de estrategia suele estar entre los 25 y los 40 años de edad, el limitado uso de mecánicas puede atraer a la parte más joven de este rango de edad. Por esto se estima un target de edades en torno a los 25 años.

⁶ En publicidad, el target hace referencia al público objetivo del producto, el cual es designado en función de una serie de cualidades y características.

Esta combinación de géneros no es nada común, lo cual reduce significativamente la competencia más directa. Sí que es cierto que existen *roguelikes* de estrategia por turnos, pero tienden a ser de juegos cartas o *RPGs*⁷ más complejos.

Durante la creación del juego se tomaron dos decisiones directamente orientadas a la introducción del juego en el mercado: la primera es el uso del inglés como idioma, para llegar a una audiencia internacional, y el segundo es la elección del nombre.

El uso de la palabra “*voxel*” en el nombre no es casual, el *voxel art* es un medio que como ya se comentó en los últimos años ha ganado fuerza, y hay usuarios que buscan juegos específicamente con este estilo. Poniendo esa palabra en el nombre, aumentan mucho las posibilidades de que un individuo cualquiera lo encuentre en búsquedas con “*voxel*” como palabra clave.

La competencia más directa, posiblemente sean juegos que también usen esta última estrategia de nomenclatura, como el futuro *Voxel Tactics* (Söderberg & Lucken). También se puede encontrar rivalidad en juegos independientes de puzzles, que tienen en común muchos elementos con este proyecto.

Comparando con este tipo de juegos en la plataforma *Steam* (Valve Corporation, 2003) se estima un precio de venta inicial de 9.99 euros, el cual es utilizado para juegos de una escala media-baja.

5.2 Previsión de ventas

La previsión de ventas es alterada drásticamente en función de la campaña de marketing realizada, la cual en este caso no puede tener mucho alcance, por limitaciones económicas.

Publicitándolo de la manera correcta, y gracias al bajo precio del proyecto, se estima que se podría vender unas 5.000 unidades a nivel global, obteniendo un beneficio aproximado de 33.300 euros, una vez restado el porcentaje que se lleva *Steam*.

⁷ Los Role Playing Games, o RPGs, son juegos en los que el usuario controla a uno o varios personajes inmersos en un mundo detallado, generalmente con opciones de personalización y mejora de estos.

5.3 Distribución

Pese a que el cálculo de beneficios se ha calculado suponiendo que todas las unidades se venden en *Steam*, el juego saldría también para la plataforma de *Epic Games Store* (Epic Games, 2018).

5.4 Previsión de costes

Se parte de una base bastante desarrollada del juego, al menos en cuanto a la programación. Por esto, para el desarrollo completo del juego, con un equipo formado por mí y por otra persona, se estima que se requieren seis meses extra de desarrollo, para crear y diseñar el resto de mundos y niveles del juego.

5.4.1 Equipos para procesos de información

El equipo desde el que se ha realizado este proyecto, y desde el que se seguirá trabajando, incluyendo periféricos, tiene un valor en torno a los 1200 euros. Con un equipo similar por parte del segundo miembro, el valor de ambos ascenderá a unos 2400 euros.

En cuanto a licencias: *Unity* y *Qubicle* requieren unos beneficios superiores a 100.000 dólares estadounidenses (en torno a 84.000 euros) al año para requerir una versión de pago, en el caso de *Unity*, o una versión profesional en el caso de *Qubicle*, por lo que se podrán usar las versiones gratuita y de 30 euros respectivamente. Las dos versiones de *Qubicle* implican un coste de 60 euros.

Se pagarán dos meses de una licencia de *Autodesk Maya*, que con una correcta organización pueden ser suficientes. Esto implica un coste total de 558 euros.

Para la edición de audio se utilizará el programa gratuito *Audacity*, y para la edición de imágenes se comprará una licencia de *Photoshop* durante tres de los seis meses, para un coste total de 89,97 euros.

Con todo esto, el coste de equipos y licencias ascendería a unos 3.108 euros, aunque restando lo que ya se ha usado durante la demo: un equipo y una licencia de *Qubicle*, este coste sería de unos 1.878 euros.

5.4.2 Los gastos

Debido al tamaño del equipo y el reducido tiempo de desarrollo, los gastos del proyecto serán relativamente reducidos.

- Gastos de personal: Tomando como referente el sueldo medio español, y que ambas personas del proyecto están formadas a nivel universitario, el sueldo bruto de cada una será de unos 2.400 euros mensuales.
Tras aplicar los cargos por la seguridad social (6.35%) y la retención del IRPF (13%), el sueldo de cada uno quedará en unos 1.935 euros netos, lo cual se considera un sueldo digno para alguien con ese nivel de formación.
- Arrendamientos y cánones: El proyecto se realizaría de manera remota por lo cual no sería necesario un lugar de trabajo común.
- Suministros y otros servicios: Se garantizaría que ambos miembros del proyecto tengan una conexión a internet adecuada, para facilitar la comunicación e intercambio de archivos. Esto resulta en unos 60 euros mensuales.
- Publicidad, propaganda y relaciones públicas: Los gastos en publicidad de un producto de estas características suelen rivalizar con el resto de gastos del proyecto. Debido a esto se invertirán en torno a 24.000 euros en todo el proceso de publicidad online y visita de ferias y convenciones.
- Amortizaciones: Los equipos informáticos son lo único que hay que amortizar, usando el coeficiente máximo esto implica una amortización del 26% anual por equipo, o lo que es lo mismo, 13% en los seis meses del proyecto.
- Gastos financieros: No hay gastos financieros, ya que la inversión para el proyecto es propia. Si fuese necesario, se pediría un préstamo.

Gasto	Gasto mensual	Gasto total
Personal	4.800€	28.800€
Suministros	60€	360€
Publicidad		24.000€
Amortizaciones		624€

Capítulo 6. Desarrollo del proyecto

6.1 Preproducción

Una vez realizada la investigación previa, y estimados los diferentes tiempos que se dedicarán a cada una de las tareas, se procede a la preproducción del proyecto. En esta fase se realizan las primeras actividades que permitirán ver cuan realistas han sido esas estimaciones, y empezarán a dar forma al juego.

6.1.1 Diseño del juego

Lo primero que se escribió fue el documento de diseño del juego, el cual define todas las necesidades de arte y programación posteriores.

Pese a esto, lo que se ve aquí no es esa primera versión: o bien las limitaciones técnicas, o bien los cambios, como consecuencia de probar algo a nivel jugable o visual y que no terminase de parecer válido, dan lugar a un documento final muy cambiado respecto a la idea inicial, que es el que se muestra a continuación.

6.1.1.1 Mecánicas

Todas las mecánicas del juego giran en torno a un sistema de juego en tablero, en el que los personajes se mueven de casilla en casilla por turnos. El jugador controla a uno de los dos equipos, pudiendo derrotar al oponente de diferentes maneras para avanzar de nivel.

6.1.1.1.1 Mecánicas de los personajes

A continuación, se explicarán las mecánicas que afectan a todos los personajes, tanto las del jugador como las de los enemigos:

- **Características:** Todos los personajes poseen tres características.

- Puntos de vida:** Los puntos de vida representan la cantidad de daño que un personaje puede recibir antes de ser eliminado.

- Velocidad:** La velocidad representa la cantidad de casillas que un personaje puede moverse en un turno, ignorando diagonales.

- Ataque:** La cantidad de puntos de vida que el personaje sustrae cuando ataca a un objetivo. Los personajes solo pueden atacar a las unidades en casillas adyacentes, ignorando diagonales.

6.1.1.1.2 Mecánicas de los caballeros

- **Cambio de unidad:** Si un soldado no ha realizado ninguna acción, puede cambiarse a otro caballero pulsando el botón de “*change unit*”.

- **Movimiento:** Al principio del turno del soldado, aparecerán señaladas las casillas a las que se puede desplazar. Esto hace que, entre la posición del caballero y

cualquier casilla, solo exista un camino posible entre ambos, sin posibilidad de modificarse.

- **Ataque:** Antes o después del movimiento del soldado, se puede seleccionar a un enemigo que se encuentre en una casilla adyacente para atacarle, lo cual terminará el turno del caballero.

6.1.1.1.3 Mecánicas del enemigo

-**Movimiento:** Cada esqueleto se mueve hasta el caballero que tenga más cerca.

- **Ataque:** Tras completar el movimiento, el esqueleto atacará a un caballero que tenga en una casilla adyacente, en caso de no haber soldados en casillas adyacentes, el esqueleto termina su turno.

6.1.1.1.4 Casillas

Se entienden como casillas a cada uno de los elementos sobre los cuales un personaje puede desplazarse. Cada casilla puede tener un único personaje encima y en caso de no estar moviéndose, estará en el centro de dicha casilla. Se pueden encontrar 5 tipos de casillas:

- **Estándar:** Casillas sin ningún efecto adicional.

- **Móvil:** Estas casillas se mueven un espacio al final del turno de los esqueletos (también denominado final de la ronda). Además, se llevan consigo a los personajes que tengan encima al final de la ronda. Un personaje que salga del escenario estando encima de esta casilla será eliminado.

- **Dañina:** Hace uno de daño a los personajes en una de las dos instancias siguientes: el personaje entra en el espacio de la casilla, o cuando el personaje termina la ronda encima.

- **Pared:** Tiene dos estados, o bien bloqueando el paso, o bien comportándose como una casilla estándar. Puede empezar el nivel en cualquiera de los dos estados, cambiando al otro cuando lo permita la casilla de la que hablamos a continuación.

-**Botón:** Se encarga únicamente de cambiar el estado de la casilla pared, el cual cambia mientras haya un personaje encima del botón, y volverá al estado inicial cuando el personaje deje el botón. Hay una excepción: si hay un personaje encima de la casilla

pared cuando se deja de pulsar el botón, el estado no cambiará hasta que el personaje no deje la casilla.

6.1.1.1.5 Mecánicas de juego.

En esta demostración existen tres modos de juego, aunque uno es exclusivo de esta demostración.

- **Tutorial:** Este modo de juego está compuesto por los seis niveles que componen el tutorial del juego, los cuales están destinados a explicar en cierta medida las mecánicas que encontraremos a lo largo del juego.

- **Standard:** El modo de juego base, en el que iremos avanzando nivel tras nivel, siendo cada uno aleatorio, aumentando una característica aleatoria cada vez que superemos cada uno.

- **Cheat:** Creado únicamente para la demo, este modo de juego permite acceder directamente a cada uno de los nueve niveles del modo *Standard*, para facilitar la corrección el trabajo sin que la aleatoriedad influya en la capacidad de probar todos los niveles.

El modo *cheat* permite además aumentar las estadísticas de los soldados para cada nivel, incluso por encima de lo que el modo *Standard* normalmente permitiría.

6.1.1.1.6 Condiciones de victoria/derrota

Voxel Knights posee una serie de condiciones de victoria posibles para cada nivel. La eliminación de todos los caballeros siempre es una condición de derrota, aunque puede haber adicionales:

- **Derrota a los enemigos:** Acabar con todos los enemigos del nivel.
- **Derrota a los enemigos antes de que se acabe el tiempo:** Igual que la anterior, pero se indica en pantalla el número de turnos máximos para completarla.
- **Sobrevive hasta el turno “x”:** Si queda algún caballero sin eliminar cuando se llega al turno “x” se logrará la victoria.

6.1.1.1.7 Controles de cámara

El juego se ve con una cámara isométrica elevada aun lado del nivel con un ángulo de 30°, se pueden usar los controles de cámara para mover la cámara a otro de los cuatro

lados del nivel, o para ponerla en un plano cenital, el cual permite ver el juego desde un punto de vista más estratégico.

6.1.1.2 Dinámicas

Todo el juego funciona a través del ratón, a través de pulsar en las diferentes partes de la interfaz y del juego para realizar las acciones deseadas.

La dinámica principal del juego es la de dirigir a los enemigos por el camino que se desee. Esto se hace sacando ventaja de que cada esqueleto se mueve al soldado más cercano por el camino más corto, independientemente si este camino es el óptimo. Por ejemplo, los esqueletos no tienen en cuenta si el camino más corto está lleno de casillas dañinas, aunque estas vayan a eliminarlo al pasar por ellas. Tampoco tienen en cuenta si el camino que siguen está formado por casillas móviles que al cambiar de posición eliminarán dicho camino, llevando a los esqueletos a estar atrapados.

Otra dinámica que nace de esta es de la capacidad de todos los personajes de bloquear el camino de los enemigos. Dado que un enemigo no se moverá a no ser que exista un camino abierto hasta un soldado, es posible forzar esqueletos a bloquear caminos. Esto es reforzado por el hecho de que los esqueletos siempre se mueven en un orden determinado, lo cual hace que a veces el primer esqueleto bloquea el camino al siguiente, haciendo que el segundo no se mueva, cuando lo óptimo hubiera sido que se movieran en el orden inverso.

Pese a que es la manera más común de hacerlo, no solo los esqueletos pueden bloquearse entre ellos: tanto los propios soldados como las casillas pared pueden bloquear caminos. Usar los caballeros para cerrar caminos hasta otros soldados es a veces la opción óptima, sacrificando a caballeros con menos vida para poder golpear con los que se encuentran en mejor estado. Este uso de la dinámica es especialmente útil en los niveles de supervivencia, donde todo el tiempo que se pueda ganar es beneficioso.

Por último, esta dinámica se puede usar para bloquear a esqueletos encima de bloques de ácido, un esqueleto encima de ácido que no encuentre ningún camino abierto hasta un caballero se quedará bloqueado encima del ácido, recibiendo daño al final de cada uno de sus turnos.

Las casillas móviles, como mencioné con anterioridad, mueven al personaje que tengan encima con ellas al final de la ronda de los enemigos, lo que las hace especialmente valiosas para aumentar la capacidad de desplazamiento de los soldados.

Otra dinámica es la capacidad de ver las estadísticas de cada personaje poniendo el ratón encima, de esta manera el jugador puede utilizar este conocimiento para colocar a los caballeros en posiciones beneficiosas, dando al jugador una enorme ventaja táctica que hace posible resolver niveles donde los esqueletos son estadísticamente superiores.

6.1.1.3 Diseño de niveles

Hay una clara separación entre los modos de juego *Tutorial* y *Standard*, y la manera en la que se ha afrontado el diseño de niveles para cada uno.

A la hora de hablar del diseño de niveles falta introducir un elemento que no se ha tratado hasta el momento, el bloque obstáculo, el cual es un tetraedro de *Unity* que se ha utilizado para marcar qué casillas son cruzables y cuáles no.

Éste es precisamente a nivel funcional el que hace que una casilla no sea transitable. Una vez hecho el *blocking*⁸ del nivel, desactivo las mallas de todos estos bloques, dejando sus *collider*, y pongo los objetos de decoración necesarios encima.

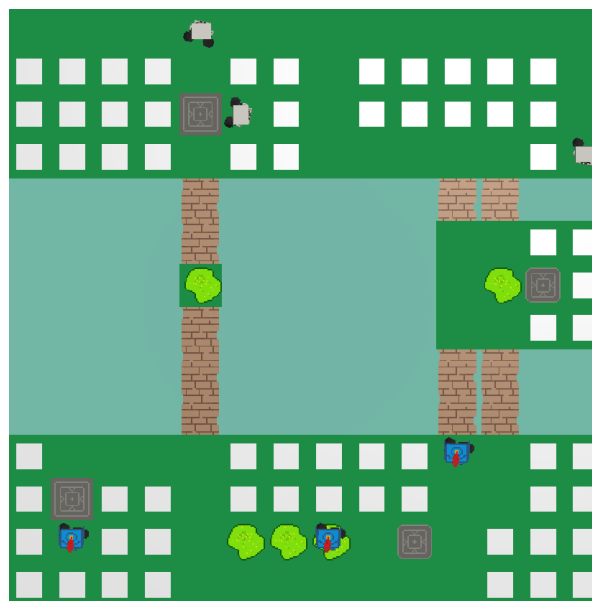


Figura 6. Blocking del nivel 3.1.

⁸ En diseño de niveles, el blocking hace referencia a la fase de maquetación de un nivel, en la que se crea estructuralmente con formas básicas, para poder probarlo a nivel funcional.

Los niveles del tutorial, tienen una única función: explicar una o varias mecánicas del juego, de la manera más intuitiva posible. Además, son deliberadamente cortos para que, en caso de fallo del jugador, pueda reintentarlos rápidamente.

Al principio de cada nivel del tutorial hay un texto guía, el cual intenta evitar el uso de lenguaje directo. La excepción son los niveles uno y dos, donde se dice explícitamente lo que el jugador ha de hacer para aprender los controles más básicos.

Las estadísticas de los caballeros cambian dependiendo del nivel del tutorial, para que el jugador se acostumbre a ir comprobándolas mirando el bloque de estadísticas.

Nivel tutorial 1

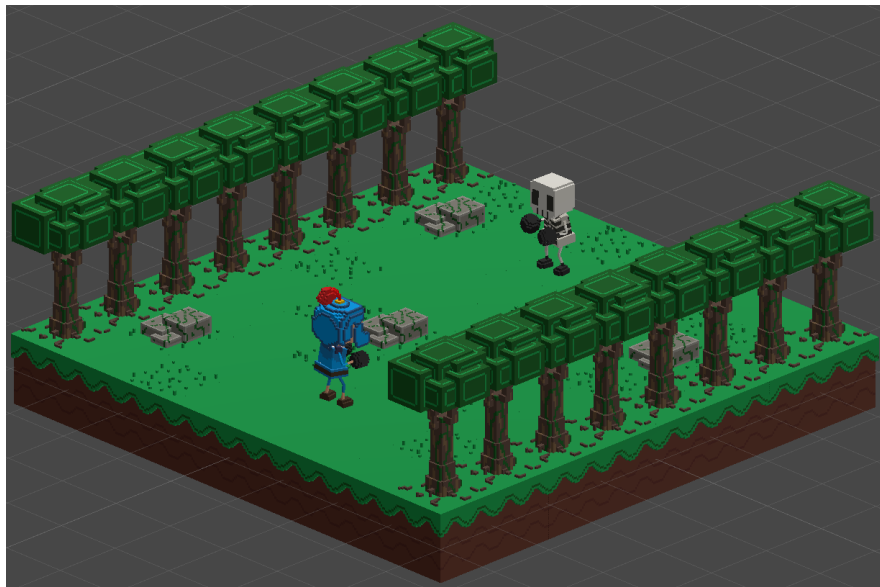


Figura 7. Nivel Tutorial 1.

Texto inicial: *“Move the knight near the skeleton and then attack to defeat them!”*.

En este nivel se introducen las dos mecánicas más básicas: el movimiento y el ataque, se deja al jugador en medio de un nivel prácticamente vacío para que no tenga muchas más opciones que ir y atacar al enemigo, contra el que es claramente superior.

Aquí se introducen los primeros elementos decorativos, usando las piedras para romper la simetría y para mostrar que existen casillas no atravesables.

Nivel tutorial 2

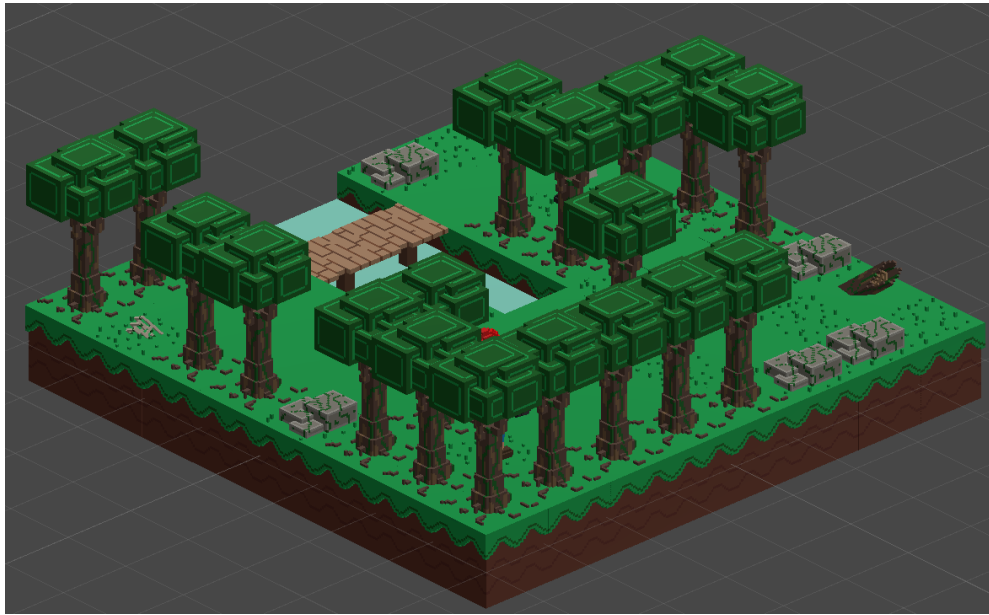


Figura 8. Nivel Tutorial 2.

Texto inicial: *“If you are having problems with the view, try the camera controls on the top left!”*.

Tanto el texto inicial como el propio nivel tratan de comunicar algo: la visibilidad es limitada, así que hay que utilizar los movimientos de cámara para poder ver el terreno por completo. Este nivel también ayuda a reafirmar los controles básicos del juego.

Nivel tutorial 3

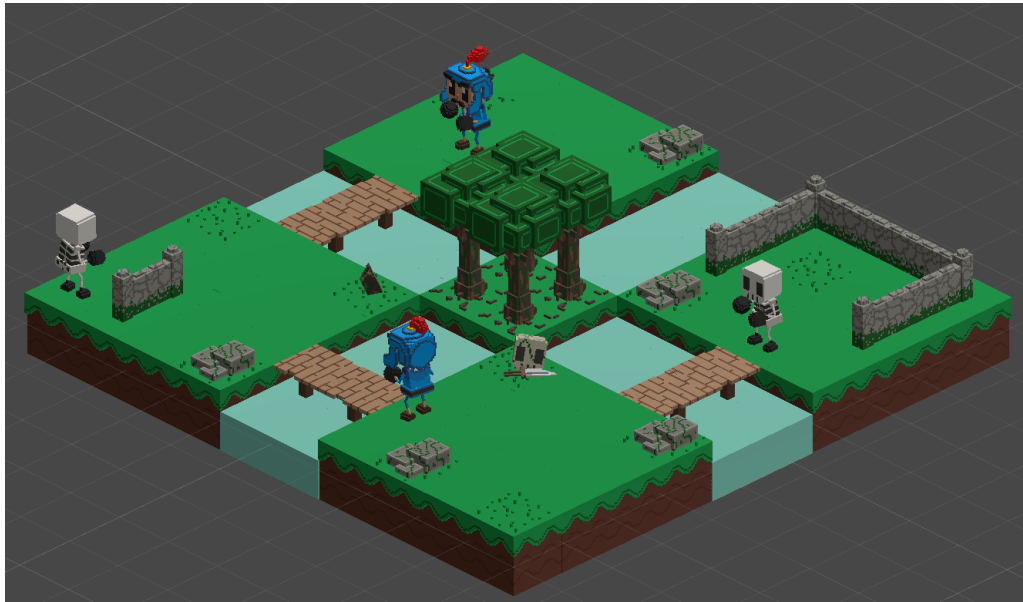


Figura 9. Nivel Tutorial 3.

Texto inicial: *“In this challenge your soldiers are weaker!, try to get a numerical advantage to win!”*.

En este nivel se introducen dos cosas: el uso de más de un soldado, con el posible cambio entre ellos, y una idea extremadamente importante del juego: el cómo la superioridad numérica supone una gran ventaja táctica, la cual es clave en muchas situaciones para superar un reto.

Nivel tutorial 4



Figura 10. Nivel Tutorial 4.

Texto inicial: *“That enemy is strong! That acid pool might help you tho...”*

Este es el primer contacto con una casilla que no es estándar, la dañina.

También se obliga a usar una dinámica, la cual consiste en sacrificar a un soldado con tal de hacer un poco de daño al enemigo.

El nivel está planteado de manera que el soldado de arriba ha de ir y atacar al esqueleto, de otra manera el esqueleto atacará y eliminará al soldado quedando impune.

Aquí, una vez más, se hace importante ver la capacidad de movimiento del esqueleto para poder golpearle primero.

Otro detalle del nivel, es que el soldado de abajo empieza junto al puente equivocado, esto es para que el jugador, si toma la decisión equivocada de ir directamente a por el esqueleto, perderá, obligándole así a buscar una solución alternativa.

Nivel tutorial 5

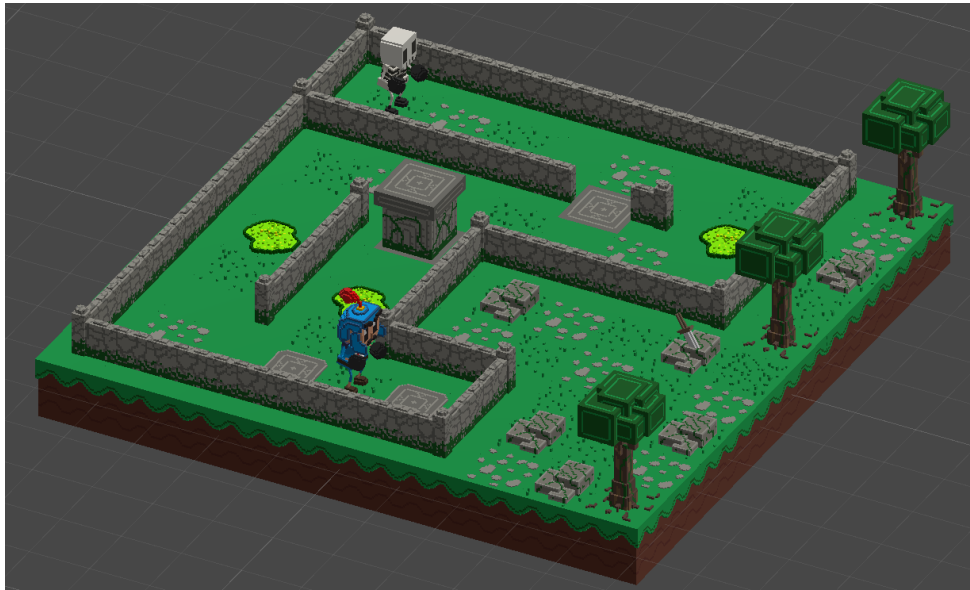


Figura 11. Nivel Tutorial 5.

Texto inicial: *“Sometimes changing the enemy’s path is better than attacking them directly!”*

De nuevo, existe una mecánica primaria y una secundaria para aprender en este nivel, la primaria es la más obvia, el caballero empieza rodeado de botones, no se pueden evitar, (el soldado solo tiene 1 de movimiento) y con ellos manipulará el movimiento del esqueleto.

La mecánica secundaria del nivel aparece al llegar a la segunda parte: pulsando el botón haríamos que el esqueleto por el camino con ácido más cercano al jugador.

Aparentemente, ambos caminos suponen lo mismo: el esqueleto pasa por encima del charco de ácido, se queda a 1 de vida, y nos gana porque nuestro soldado tiene 0 de ataque. Aquí es donde aparece la mecánica secundaria. Parece extraño que, exista un botón para forzar al esqueleto a ir por un camino que hace lo mismo que el otro.

La diferencia es que, está medido para que en el camino que se abre al pulsar el botón el esqueleto termine su turno en el ácido, recibiendo daño adicional al final del turno descubriéndose así la mecánica secundaria del nivel.

Nivel tutorial 6

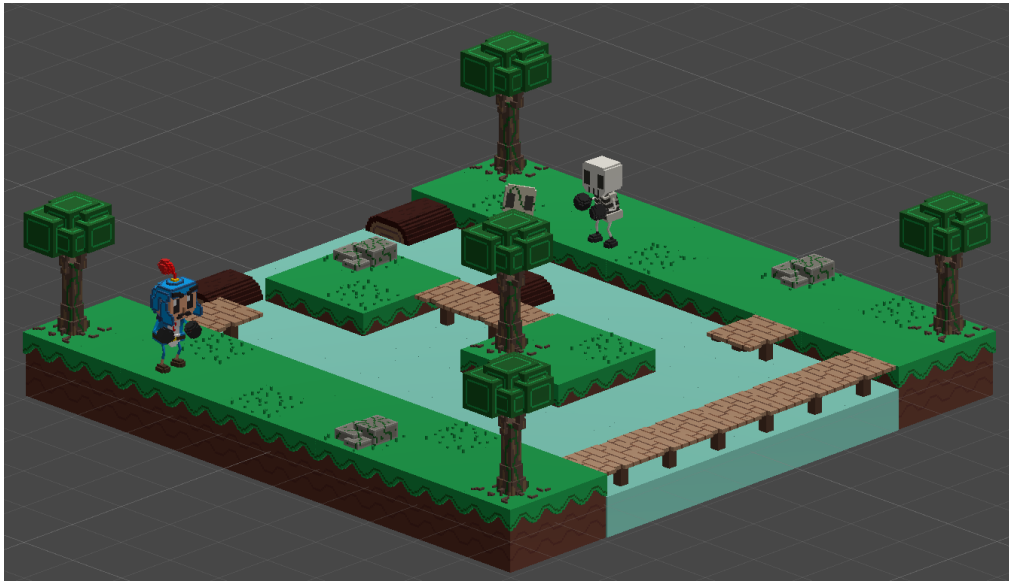


Figura 12. Nivel Tutorial 6.

Texto inicial: *“Try to get to the enemy before time runs out!”*.

El último nivel del tutorial es simple, y a la vez el menos explícito respecto a la resolución del mismo. El texto inicial simplemente te indica que tienes tiempo limitado, introduciendo así esta condición de victoria. Y aquí es donde el jugador puede tomar dos posibles caminos.

El primer camino es ir por el puente de abajo, en cuyo caso no llegará a tiempo, pero a partir del primer turno verá cómo se mueven los troncos del agua.

Si en su primer intento se ha equivocado, el jugador ya intentará ir a través de los troncos hasta el enemigo y vencerá, proceso durante el cual aprenderá dos cosas: primera, un personaje encima de un tronco al final de la ronda se mueve con el tronco; segunda, los esqueletos no se mueven al no tener ruta.

En cuanto al modo de juego *Standard* y sus niveles:

Es importante recalcar que en la demostración hay tres listas de niveles aleatorios, y cada una contiene tres niveles entre los que elegir, pero esto no sería así en la versión final. En la versión final del juego idealmente habría en torno a diez niveles en la primera lista, y cinco en el resto. Esto es así porque en los niveles de la primera lista no existe la variabilidad de tener alguna estadística mejorada, además de que son los niveles que más

se va a encontrar el jugador y por tanto más fáciles de recordar la solución, lo cual se pretende evitar.

Como nota adicional, en este proyecto solo se muestra el primer mundo, el pantano, el cual se compondría de cuatro niveles y un nivel de jefe. Al completar el pantano en el modo *Standard* por primera vez, se desbloquearía acceso a la siguiente zona, en la que habría nuevos tipos de casilla con nuevas mecánicas, y nuevos enemigos.

Cada zona diferente se accedería en una partida nueva, por ejemplo: el mundo que se desbloquea tras completar el pantano no se accede en esa misma partida, simplemente se podría empezar de cero partidas nuevas en la nueva zona.

Voxel Knights es una amalgama de numerosas categorías de juegos, lo cual hace que cada una de ella impregne a los niveles de una manera específica, especialmente a los del modo *Standard*.

La aleatoriedad del nivel al que avanzas y la aleatoriedad de la característica que se mejora generan una problemática: una vez que pasas de la primera lista a la segunda y sube una de tus estadísticas, ha de existir una solución para la resolución del nivel, para todas las combinaciones de estadísticas posibles. En resumen: no puede ser posible entrar en un nivel que no se pueda pasar, y para ello ha de existir una solución para cada una de las posibles mejoras.

Existe una segunda problemática, que solo se aplica a los niveles de la segunda lista con rigidez, esta es: tiene que ser imposible pasarse un nivel sin la mejora de característica del nivel anterior. De ser posible, esa mejora de característica sería irrelevante y el nivel siempre se podría resolver de la misma manera, eliminando enormemente la complejidad del juego y banalizando la mitad del sistema.

Por ejemplo: si la mejora que se ha obtenido en el nivel uno no es necesaria para superar el nivel dos, significa que existe una solución que no requiere ninguna mejora, la cual se podría realizar siempre independientemente de qué mejora se obtenga.

A pasar a la tercera lista de niveles esto es mucho más complicado, porque ahora el jugador tiene la mejora del primer nivel, y la del segundo, por lo cual a partir del segundo nivel solamente me he centrado en que existan soluciones independientemente de tu combinación de estadísticas.

Toda esta problemática acaba dando sus frutos: un juego en el que el número de niveles numéricamente no es necesariamente tan alto, pero la variedad de soluciones a cada nivel le añade una gran rejugabilidad proporcionalmente.

Además, el texto inicial que veíamos en los niveles del tutorial sigue existiendo, aunque ahora simplemente nos señala los objetivos de victoria para cada nivel.

En el modo de juego *Standard*, cada nivel es independiente del resto, pero sí que es cierto que todos se rigen por dos reglas, ambas con la intención de ir aumentando la complejidad a medida que el jugador avanza.

Estas reglas son: El número de caballeros con los que se juega es igual al nivel en el que se esté, y el tamaño del escenario aumenta en dos filas y dos columnas cada vez que se avance un nivel.

Las estadísticas base para los soldados del modo *Standard* son: 1 de daño, 2 de movimiento, y 3 de vida.

Las mejoras de movimiento aumentan el control del entorno y del movimiento de los enemigos; las de vida permiten usar mejor las casillas dañinas pudiendo pasar por encima, además de que, al tener más vida, se puede bloquear a esqueletos en charcos de ácido durante más turnos; por último, mejoras en daño favorecen el combate más directo ya que se acaba con enemigos el significativamente más rápido.

Nivel 1.1

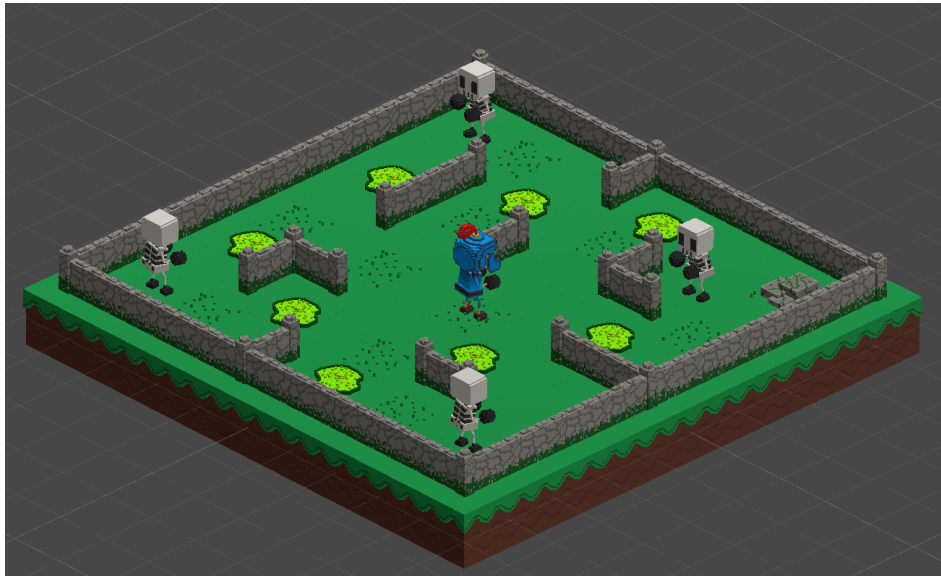


Figura 13. Nivel 1.1 Standard.

Los tres niveles de la primera lista han sido creados para reforzar el conocimiento sobre las mecánicas del juego ya vistas en el tutorial. Este nivel en concreto pretende reforzar la comprensión de las casillas dañinas, cuándo infligen daño y cuándo no.

La condición de victoria de este nivel es acabar con los enemigos sin límite de tiempo. En este nivel el jugador practicará y pondrá a prueba sus habilidades para controlar el movimiento de los esqueletos, y además tendrá que hacer un juicio correcto sobre si debe, y cuándo debe, pasar por encima de casillas dañinas para acabar con el enemigo.

Nivel 1.2



Figura 14. Nivel 1.2 Standard.

Este nivel es donde se refuerza el uso de los botones y de la manipulación del camino de los enemigos.

El objetivo es sobrevivir hasta el turno 10, y esto ha de llevarse a cabo retrasando a los enemigos el mayor número de turnos posibles, y esto se hace con el uso de tres botones, colocados respectivamente en el orden de las paredes que levantan.

Nivel 1.3

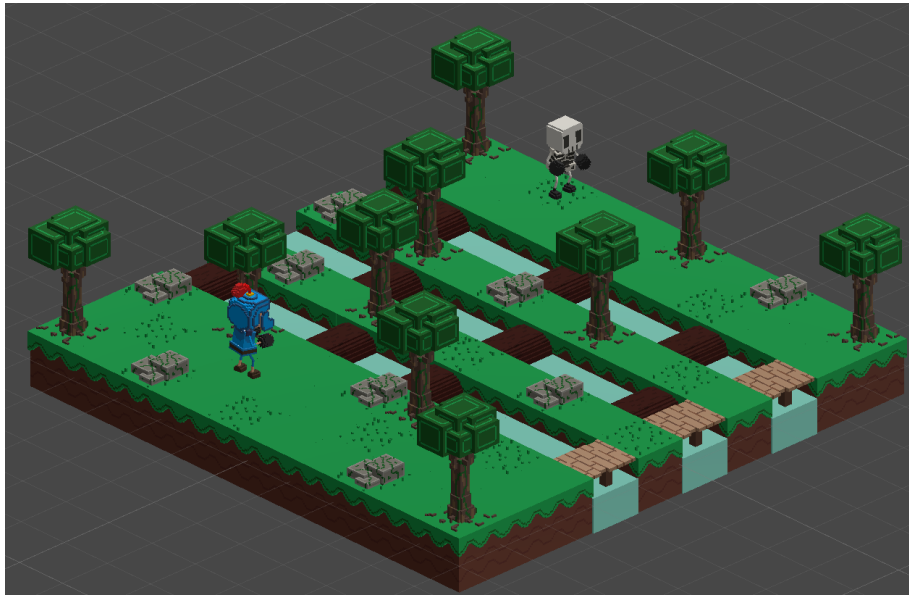


Figura 15. Nivel 1.3 Standard.

En el último nivel de la primera lista se exploran las mecánicas de las casillas móviles, haciendo mandatorio el uso de las mismas para encontrar un camino que llegue hasta el enemigo a tiempo. El puente, al igual que en el último nivel del tutorial, está para engañar al jugador, ya que no se puede llegar a través de él.

Este nivel pone a prueba tanto la capacidad del jugador para anticiparse a los cambios de escenario en cada turno: saber cuándo el esqueleto se va a mover o no, ver en qué tronco se puede subir para moverse con él e ir más rápido, qué camino le da la iniciativa contra el enemigo, etc.

Nivel 2.1



Figura 16. Nivel 2.1 Standard

Aquí es donde se empiezan a ver diferentes maneras de afrontar un nivel en función de la característica aumentada, no se van a comentar cada una, pero por ejemplo: si la característica aumentada de los soldados es el movimiento, esta característica será 3 (recordemos que el movimiento es 2 de base). Ahora el soldado de abajo puede liberar al de arriba en el primer turno, ya que está a justo 3 casillas del botón.

Esto permite al soldado de arriba quedarse encima de la casilla móvil, atrayendo al esqueleto de arriba hacia el ácido, y luego puede meterse de nuevo en su posición inicial, habiendo dañado al esqueleto y saliendo impune.

Mientras esto ocurre, el esqueleto de abajo irá a por el soldado que ha activado el botón, sin sufrir demasiado daño ya que el camino de arriba tiene solo un charco de ácido, pero el soldado puede “bailar” entre los charcos de ácido de abajo para acabar con el esqueleto.

Estas son la serie de ventajas y sacrificios que surgen de tener una característica u otra: el movimiento extra significa que se tiene mucho mayor control sobre el entorno, pero a costa de poder en combate directo, por lo que se premia tu capacidad para hacer que los esqueletos se metan en casillas de daño.

Nivel 2.2

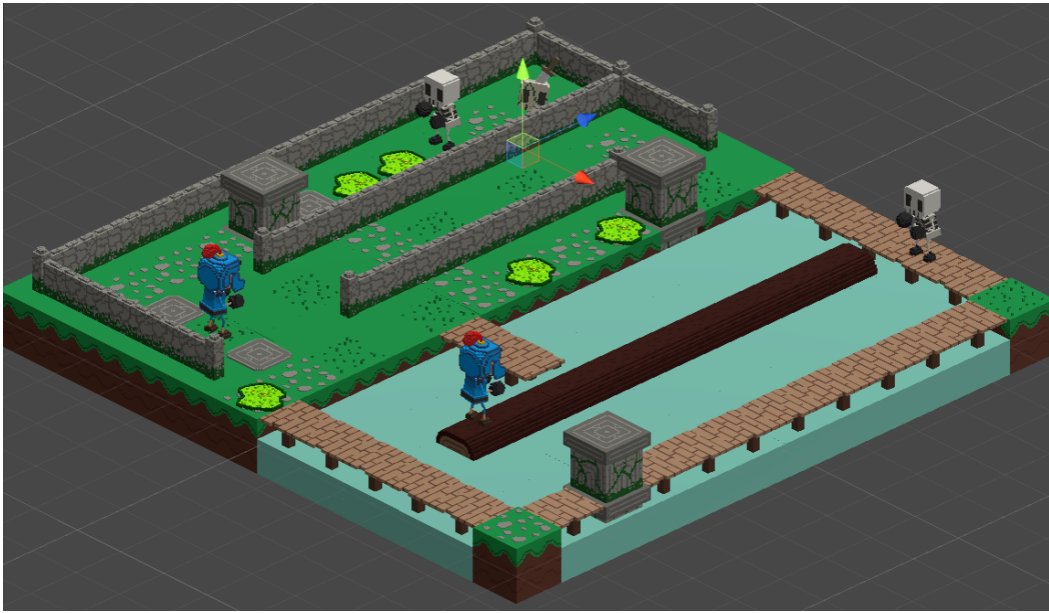


Figura 17. Nivel 2.2 Standard.

Aquí pongo en práctica el uso de piedras planas para marcar la conexión entre casillas de botón y casillas de pared. También el esqueleto de arriba va a activar un botón en el primer turno, dejando claro a qué pared está conectado, dejando al jugador determinar las otras dos por descarte.

En este nivel el jugador tendrá que intentar usar el ácido y la ventaja numérica para vencer a los enemigos: esto nace de la capacidad del jugador para conseguir que el caballero de abajo escape, o bien en el turno uno si tiene una mejora de movimiento, o bien más tarde con la ayuda de la pared de abajo para bloquear al esqueleto.

Nivel 2.3

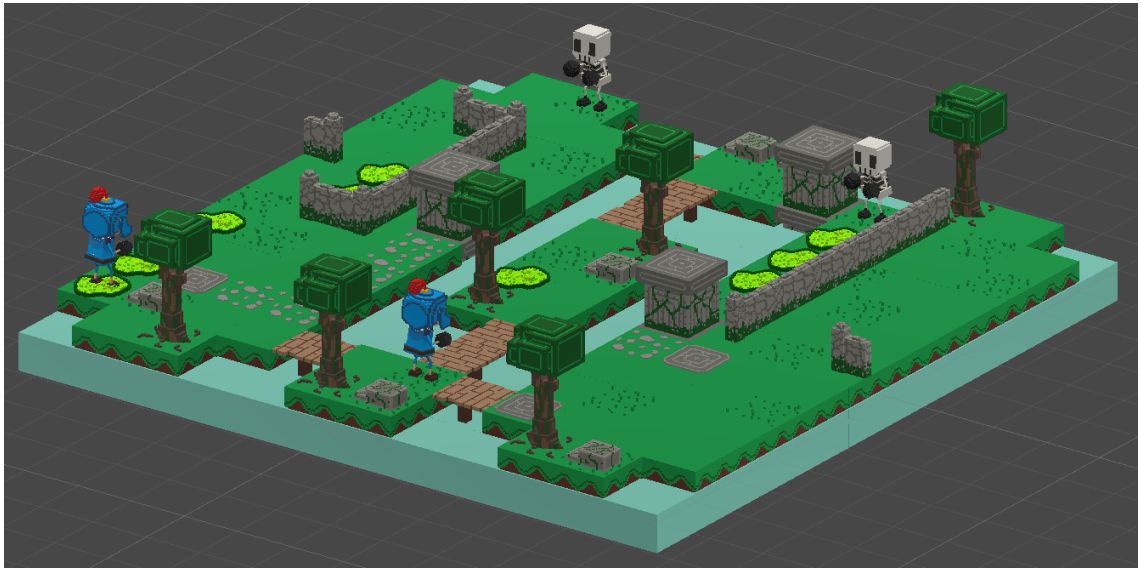


Figura 18. Nivel 2.3 Standard

El jugador tiene un tiempo muy limitado para eliminar a los esqueletos, hay multitud de ácido para acabar con ellos, pero hay que encontrar la solución óptima para acabar con ellos en el menor tiempo posible.

El que un caballero empiece en ácido es para obligarle a pasar por el otro charco, esto es relevante porque en caso de haber obtenido una mejora de vida o no, la diferencia entre tener 2 o 3 de vida tras pasar por el charco implica morir de uno o dos golpes contra los esqueletos. Este es otro de los ejemplos de cómo el diseño de niveles está hecho para cambiar la estrategia del jugador en función de sus características.

Nivel 3.1

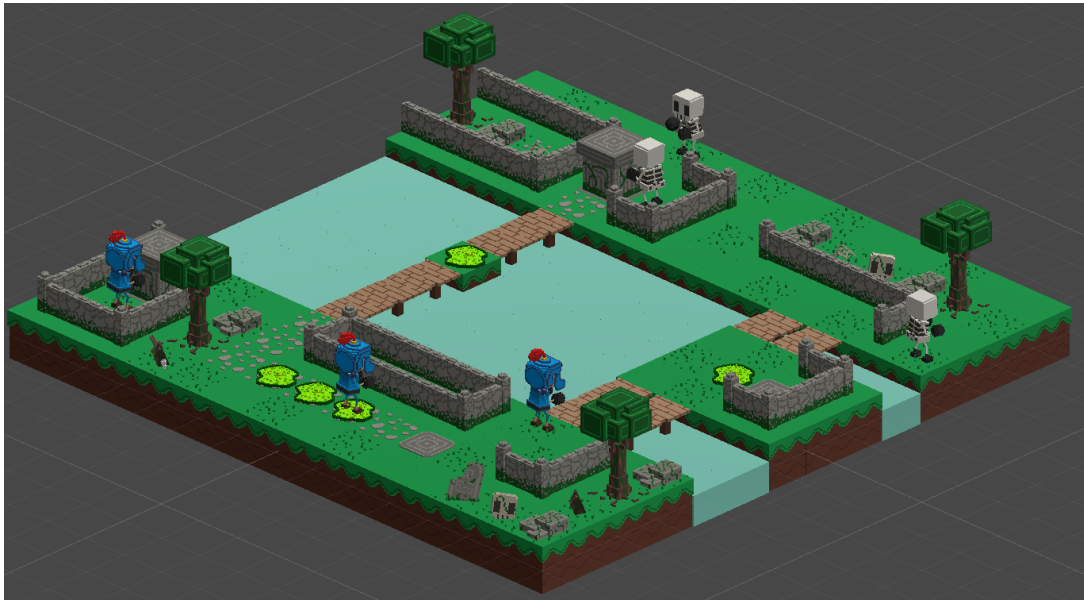


Figura 19. Nivel 3.1 Standard.

Los niveles de la tercera lista son más caóticos, dado el número de caballeros a manejar, y por ello he intentado que sean más sencillos de base, con un abanico de soluciones más amplio, ya que aumento del número de caballeros ya aumenta suficiente la complejidad.

En este primer nivel hay que decidir por qué camino se quiere hacer venir a los esqueletos, además de decidir si merece la pena intentar liberar al soldado de la izquierda, ya que haciendo esto ponemos a otro soldado en riesgo. Un correcto equilibrio de estas variables es necesario para la victoria.

Nivel 3.2

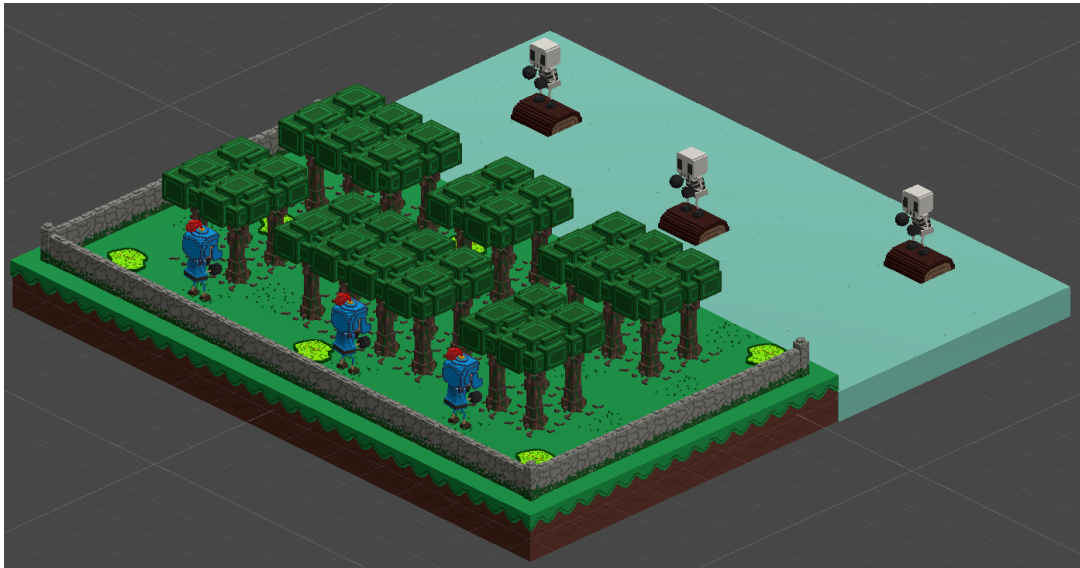


Figura 20. Nivel 3.2 Standard.

Este nivel trabaja con la capacidad de planificación del jugador: los esqueletos son muy superiores en términos de características, pero a la vez llegan en turnos diferentes y hay multitud de ácido por el suelo para irlos desgastando hasta que lleguen hasta los soldados.

La estrategia a seguir independientemente de las características que se tengan es similar en todos los casos, pero con pequeñas variaciones también en función de ellas

Nivel 3.3

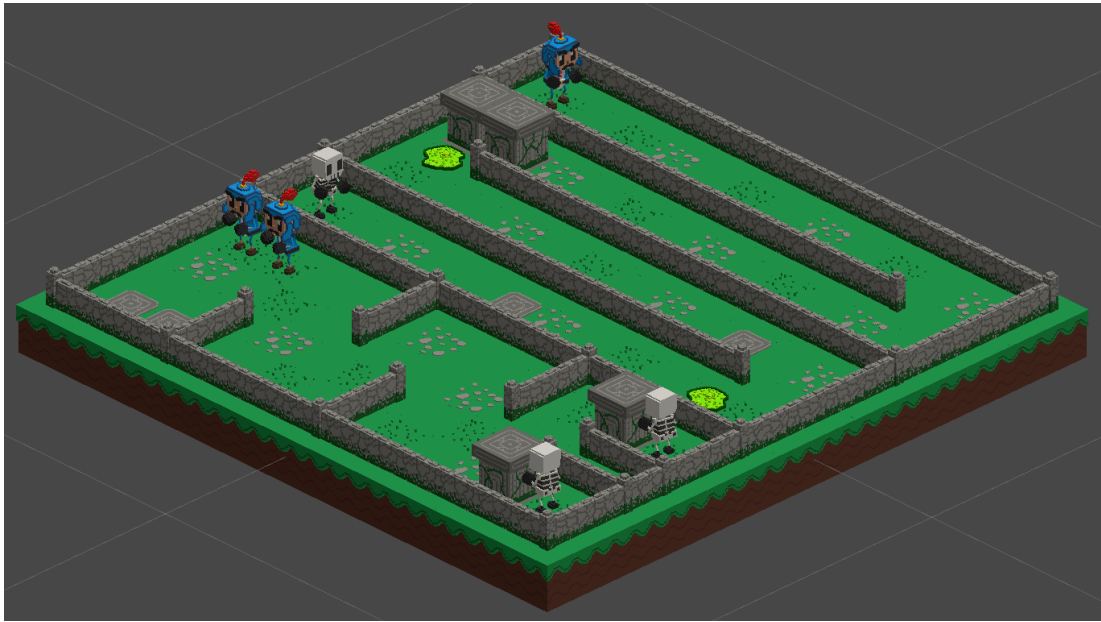


Figura 21. Nivel 3.3 Standard.

De todos los niveles, este quizá sea el que más partido saca de las mecánicas menos obvias del juego: por un lado hay que controlar al esqueleto de arriba abriéndole y cerrándole el camino, haciendo que se quede así atrapado en charcos de ácido; por el otro lado, este esqueleto irá liberando a los otros dos, y hay que medir tanto cuando le permitimos liberarlos, tanto cómo usar nuestros soldados para acabar con ellos.

Este nivel requiere acabar con todos los enemigos antes de un tiempo determinado, así que estrategias que impliquen dejar al esqueleto de arriba en ácido un tiempo indefinido no son válidas.

6.1.2 Diseño de personajes (concept).

Solamente existen dos tipos de personajes en el juego, los caballeros y los esqueletos. Durante las primeras aproximaciones, los soldados tuvieron formas totalmente diferentes de la final, de hecho, se contempló que hubiese varios tipos, para finalmente se decidiese que solo habría uno.



Figura 22. Pruebas diferentes personajes de baja resolución.



Figura 23. Prueba diferentes personajes de alta resolución.

Durante estas pruebas de diseño se tuvieron en cuenta tres cosas: cómo se animaría al personaje dada la distribución de sus extremidades, el número de *voxels* de las mismas, y el nivel de detalle del personaje, ya que afecta directamente a la forma de las armas. Un personaje con muy poco detalle usará armas extremadamente simples. A continuación, vemos ejemplos de esto.



Figura 24. Pruebas de animación para baja resolución.



Figura 25. Prueba de personaje de mayor resolución con arma

Al final se decidió que ninguno de estos diseños era válido, y que se necesitaba una resolución mucho mayor, de no ser así, el nivel de detalle de los escenarios y armas se vería seriamente perjudicado. Además, pronto se percibió que estos diseños pseudo-realistas se iban a perder visualmente en el tipo de juego que se estaba desarrollando, con una cámara relativamente lejana.

6.1.3 Diseños narrativos

El juego carece de una narrativa compleja, más allá del enfrentamiento entre los caballeros y los enemigos que se encuentren.

De lo que si dispone es de una ligera narrativa visual encapsulada en cada uno de los niveles, donde por ejemplo la presencia de una calavera en el suelo da a entender que ha habido un enfrentamiento anterior, una acumulación de tumbas da a entender que el enfrentamiento está ocurriendo en un cementerio de donde salen los esqueletos, etc.

6.2 Arte

La estética tiene un papel fundamental en el juego, de ahí el título heredado. Lo primero, antes de entrar en cómo ha sido implementada la estética en este juego, es necesario plantearse dos cuestiones:

La primera cuestión es, ¿por qué utilizar *voxel art* para el juego?, lo cierto es que, el *voxel art* es un medio que, pese a que puede tener una enorme profundidad y complejidad, permite un flujo de trabajo cómodo, y da buenos resultados en alguien no tan creativo en el sentido artístico. Además, es ideal para un juego formado por casillas.

Pese a que cada apartado que se tratará a continuación habla de las decisiones de diseño para cada elemento, hay una decisión que se cumple para todos los elementos: el tamaño del *voxel* no cambia. En caso de ser así el juego desentonaría visualmente. Esto además limita el detalle máximo posible en los elementos más pequeños del juego.

6.2.1 Modelado 3D y texturizado

Dada la metodología de trabajo del *voxel art*, el modelado y el texturizado ocurren casi de manera simultánea, ya que el uso del color en este medio es esencial para la comprensión de las formas.

6.2.1.1 Diseño de personajes

El diseño acabó siendo algo más “*cartoon*” con respecto a los conceptos iniciales, con formas claramente marcadas y con algún detalle menor para aumentar la personalidad, para ello se tomó inspiración en otro juego de perspectiva isométrica en el que los personajes están bien diferenciados: los juegos de la saga *Bomberman*, con el consiguiente resultado. (Konami, 1983)



Figura 26. Comparación ilustrativa de la diferencia de resolución.



Figura 27. Diseño final del caballero.

Hay numerosas partes que sirven para darle personalidad a la cabeza del caballero, la que a su vez es lo más importante del diseño, ya que es la que el ojo del jugador ve primero. Estas partes se pueden dividir en dos a su vez: las estáticas, que son los ojos, la nariz y el bigote; y las móviles, que son las solapas del casco, la pluma y las cejas, que se mueven de manera independiente y cumplen la función de darle algo más de dinamismo al personaje eliminando rigidez.

En cuanto a brazos y piernas, se contempló, dejando solo manos y pies, como ocurre con el personaje *Rayman* pero se desechó esa opción debido a que se perdía visualmente un poco la forma del personaje desde la posición de la cámara. (Ancel, 1995).

Las manos quedaron finalmente como esferas porque si no, no se apreciaría el detalle de las mismas, y así se simplificó el proceso de animación.

Respecto a los esqueletos, una vez creado el soldado, este proceso fue muy simple, se necesitaba un personaje con prácticamente las mismas proporciones para reutilizar las animaciones, por lo que lo único que se hizo fue adaptar el diseño de un esqueleto a las proporciones ya dadas por el caballero.

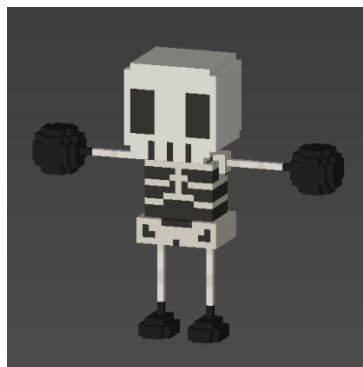


Figura 28. Diseño final esqueleto.

6.2.1.2 Diseño de las casillas

Para el diseño de las casillas se ha cuidado mucho la forma de las mismas, para que encajen perfectamente al escenario, y también se ha buscado la claridad visual para que sean rápidamente reconocibles. A excepción de las casillas estándar, cada uno de los tipos tiene un único estilo visual, para evitar confusión en el jugador.

Realmente solo existen dos tipos de casillas estándar, los puentes y los bloques de césped, aunque los bloques de césped tienen muchas variantes para darle algo de profundidad al diseño, ya que sobresale un poco el césped del resto de la forma, y depende de la esquina.

También añadido aquí el bloque de agua, aunque realmente es un bloque de decoración, sirve para ayudar a construir el nivel en conjunto con el resto de casillas.

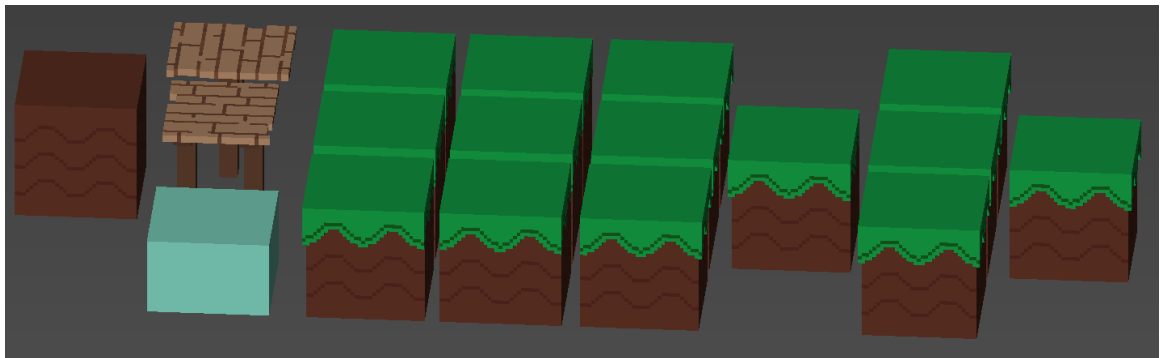


Figura 29. Bloques estándar + bloque de agua.

En la siguiente imagen se explica la expresión “sobresale el césped”, podría simplificarse mucho el proceso usando una sola casilla sólida donde no sobresalga, pero ese ligero cambio mejora la silueta lo suficiente como para que merezca la pena usar catorce diseños en vez de uno solo.

Se ve que muchas de estas casillas están repetidas, y únicamente rotadas, es necesario que sea así porque la orientación de cada casilla ha de ser la misma para el correcto funcionamiento del juego. Si se girasen, el juego no funcionaría correctamente.

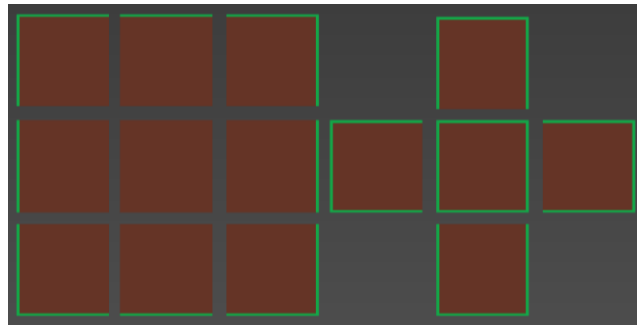


Figura 30. Casillas de césped vistas desde abajo.

Las casillas móviles son representadas por un tocón, y están siempre en el agua, están diseñadas de manera que encajan entre ellas para poder formar un tronco más largo. Este diseño es mucho más oscuro que el resto de casillas, para que sea rápidamente reconocible.



Figura 31. Casilla móvil.

Los bloques de pared y de botón, al ir juntos, tienen un diseño superficial muy similar, esto es para que se reconozca rápidamente su conexión.

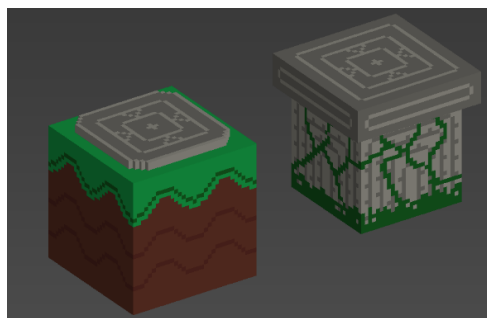


Figura 32. Casillas botón y pared.

La casilla dañina es una casilla estándar con un charco de ácido en su parte superior, ésta, para darle un especial cuidado y sensación de importancia y peligrosidad, ha recibido cambios con respecto a las otras.



Figura 33. Casilla dañina.

Estos cambios son: colores llamativos con respecto al resto de la paleta del juego, además de ser la única casilla con animación propia.

Se verán burbujas elevándose y explotando encima, además de pequeños cambios visuales en el propio charco. Esta animación es una mezcla entre movimiento estándar de las burbujas, y cambio completo de los modelos en fotogramas determinados.

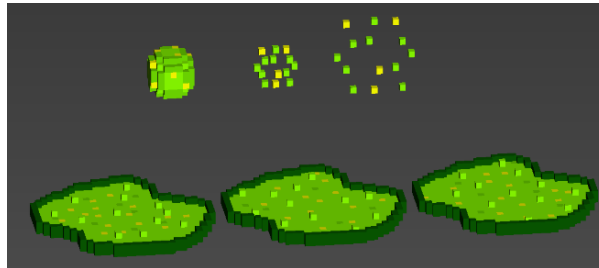


Figura 34. Modelos utilizados en la animación del ácido.

6.2.1.3 Diseño de la decoración

Los elementos de decoración son complementarios a las casillas y sirven para mejorar la experiencia visual del jugador dándole más riqueza a los escenarios.

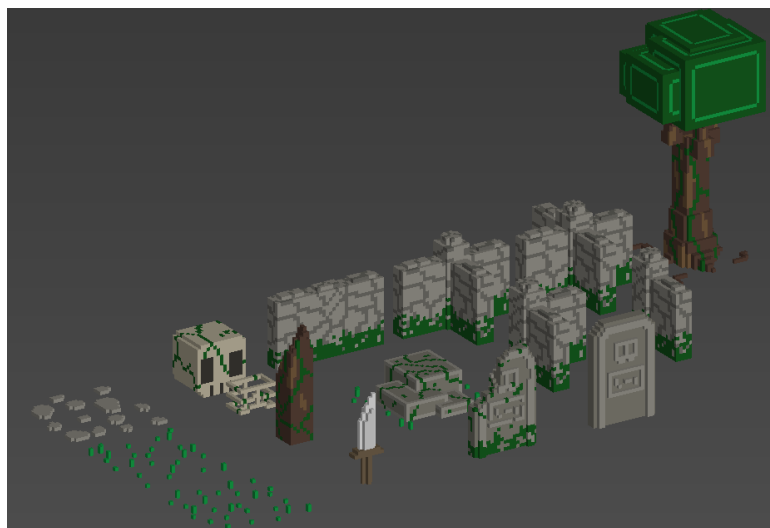


Figura 35. Assets decorativos.

Los usos de la decoración son limitados, pero importantes, la gran mayoría de ellos, a excepción de los dos objetos de césped, y el de piedras planas (a la izquierda del todo de la imagen anterior), marcan casillas por las que el jugador no puede pasar, además de usarse en el diseño de niveles para oscurecer la posición de algún enemigo o casilla relevante, dando así importancia al uso de los controles de cámara.

6.2.1.4 Elementos de la interfaz

Todos los iconos y botones son objetos 3D han sido modelados y renderizados para obtenerlos en formato *png*. *Unity* permite el uso de objetos 3D en la interfaz, pero en caso de implementarse así, son afectados por la propia iluminación de la escena, por lo que la mejor opción resultó ser el uso de imágenes.

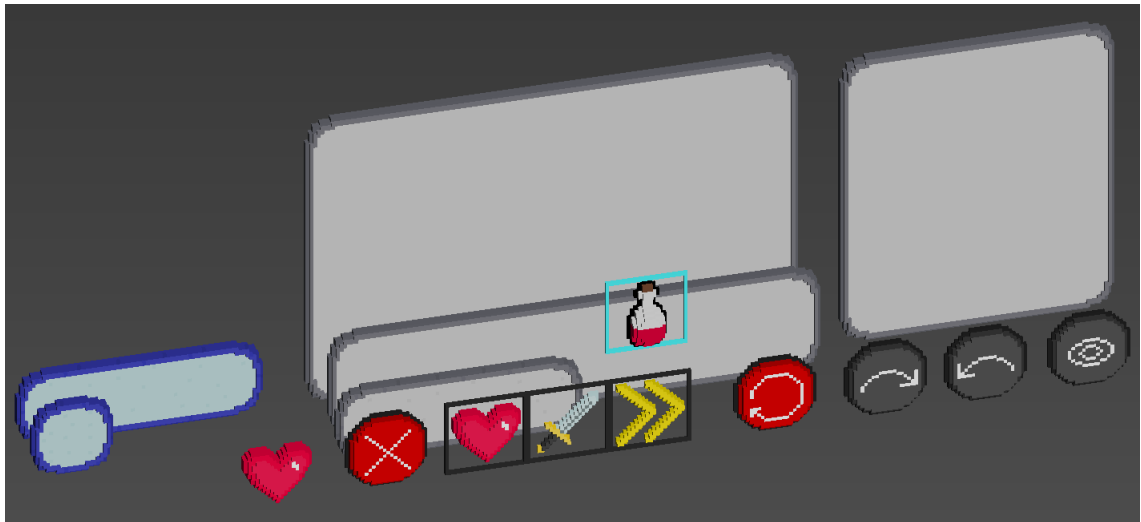


Figura 36. Elementos de la UI en *Qubicle*.

6.2.1.5 Skybox

Un elemento decorativo especialmente importante es el *skybox*, o el fondo, este es un material tridimensional que rodea al escenario, sirviendo de fondo y complementando la iluminación. Para este proyecto se realizó uno propio, con nubes *voxel art*.

Para esto creé el modelo de las nubes en *Qubicle*, rendericé los modelos en *Maya* usando *Arnold*, y luego lo monté en *Photoshop*, obteniendo el siguiente resultado:

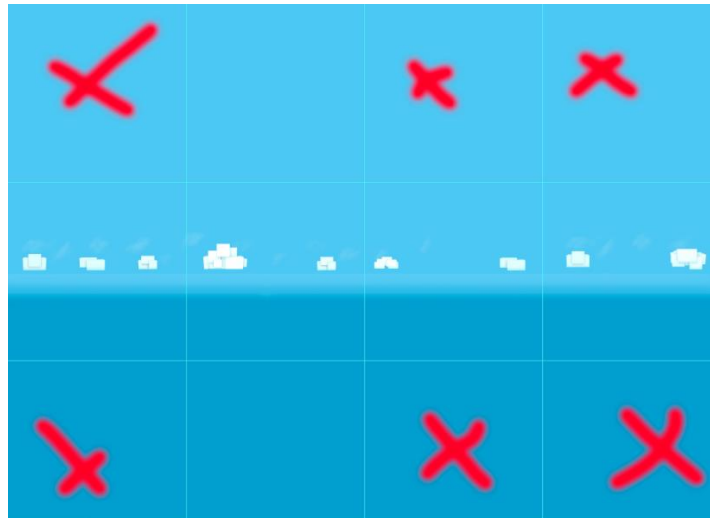


Figura 37. Archivo Photoshop del Skybox.

Este es el archivo necesario para añadir un *Skybox* de 6 lados a *Unity*, exportando cada uno de los cuadrados por separado para luego juntarlos en el material de *Unity* como si fuera un dado, los cuadrados con cruces son los que no se necesitan.

La dificultad de hacer un *skybox* reside en colocar y escalar las nubes al tamaño apropiado para el juego que se esté haciendo. El resultado de aplicar este *skybox* es el siguiente:

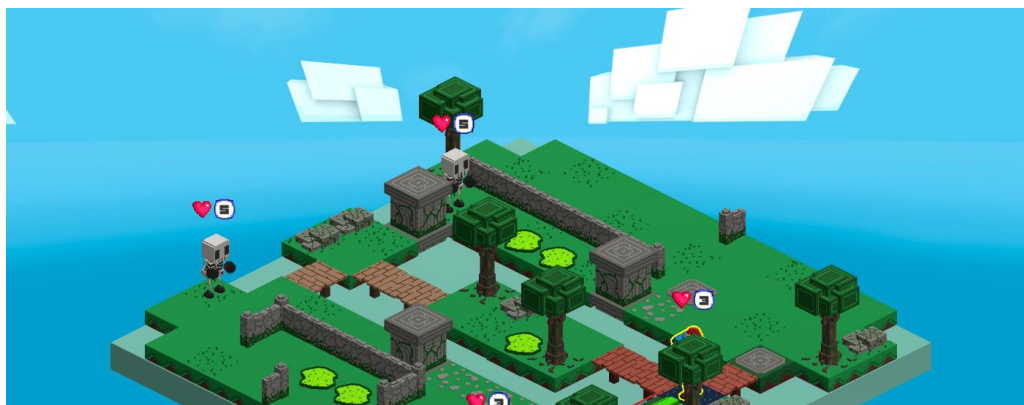


Figura 38. Skybox en Unity

6.2.1.6 Paleta de colores

Se ha elegido una paleta de colores con un rango muy pequeño, pero con bastante variedad dentro del mismo. Prácticamente todos los colores que componen el escenario son azules, verdes, grises y marrones, a excepción de la pluma de los soldados que ayuda a encontrarlos en pantalla usando un color rojo vivo.

En los colores de la interfaz se han usado múltiples colores, entre ellos más variedad de rojos y naranjas para que contrasten frente al resto.

6.2.2 Rigging de personajes.

Una vez más, ambos personajes comparten el mismo *rig*, hecho inicialmente en el caballero ya que tiene un ligero grado de complejidad mayor, aunque para ambos se ha hecho un *rig* muy simple:

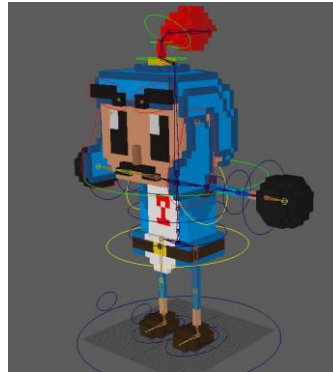


Figura 39. Rig del caballero

Para el *rig*, se han combinado dos estilos de *rigging* comunes en *voxel art*, “*hard rigging*” y “*soft rigging*”, la principal diferencia es que el primero hace que cada elemento solo tiene influencias de los huesos que lo componen, haciendo que no haya deformación entre elementos contiguos. El segundo estilo, “*soft rigging*” hace referencia a la manera tradicional de *rigging*, con deformaciones como las esperaríamos.

Por ejemplo, los objetos que no tienen influencia de otros son: las manos, los pies, las piernas, la cabeza y los brazos.

Por otro lado, elementos como el tronco sí que se ven afectados por huesos colindantes, como los de los brazos. Además, los brazos, las piernas, y los pies sí que tienen deformaciones dentro de ellos, en el codo, rodilla y puntera respectivamente, pese a que la totalidad de estos elementos no se ven afectados por otros, como puede ser en la unión con el torso de los brazos y piernas.

Las cejas, y las solapas y la pluma van independientes, aunque la pluma tiene algo de deformación dentro de ella para hacerla más dinámica. Estos elementos son la única diferencia entre ambos *rigs*, debido a que el del esqueleto carece de ellos.

Todas estas decisiones en el *rig* se han tomado deliberadamente para complementar a la estética *cartoon* mencionada anteriormente, dando esa sensación de independencia a las extremidades.

6.2.3 Animación de personajes

El número de animaciones de personajes es limitado: existen tres animaciones que son iguales en ambos personajes. Estas animaciones son *idle*⁹, ataque, y ciclo de correr.

El idle y el ciclo de correr son bastante estándar. Lo más destacable son las animaciones a las cejas, a las solapas del casco y a la pluma del casco para dar más personalidad al personaje.

Como se puede observar durante el juego, los personajes, por defecto, no llevan ningún arma, esto define la animación de ataque:

Ésta es un poco más compleja: a mayores de la animación en *Maya* del movimiento del personaje, en *Unity* he añadido la animación de la espada apareciendo a medida que el personaje carga el ataque.



Figura 40. Espada generándose en mitad del ataque.

Para obtener este resultado, se alterna entre varios modelos de la espada que han sido previamente creados en *Qubicle*, obteniendo una manera de que el personaje empiece y termine la animación de ataque sin el arma, conectando así bien con el resto del movimiento.

⁹ Animación de un personaje en reposo, para evitar que dé una sensación totalmente estática.



Figura 41. Espada desapareciendo al final del ataque.

6.2.4 UX e UI

En todos los apartados del juego existen detalles para hacer la interfaz de usuario intuitiva y fluida, intentando que el jugador se pueda centrar en el juego sabiendo cómo puede actuar en cada momento. Para esto se han tomado una serie de decisiones.

El uso de la vista isométrica es fundamental, debido a que es importante la claridad de la distancia entre las casillas, la cual sería alterada si se usase una vista en perspectiva.

El estilo de estos elementos es *voxel art*, al igual que el resto del juego, y para el texto se utiliza una fuente de estilo *pixel art*, la cual debido a la similitud se adopta bastante bien. La fuente se denomina “*Thalea Fat*”. (Hoppmann, 2019).

6.2.4.1 Menú

El menú principal es lo primero que el jugador encuentra al empezar a jugar, y por donde ha de pasar cuando es derrotado, cuando completa un nivel en el tutorial o en el modo *cheat*, o cuando termina por completo el modo *Standard*.

Para una navegación sencilla, el menú dispone de los botones estrictamente necesarios: unas opciones en las que cambiar el volumen del sonido, el acceso a los diferentes modos de juego, y el botón para cerrar la aplicación por completo.

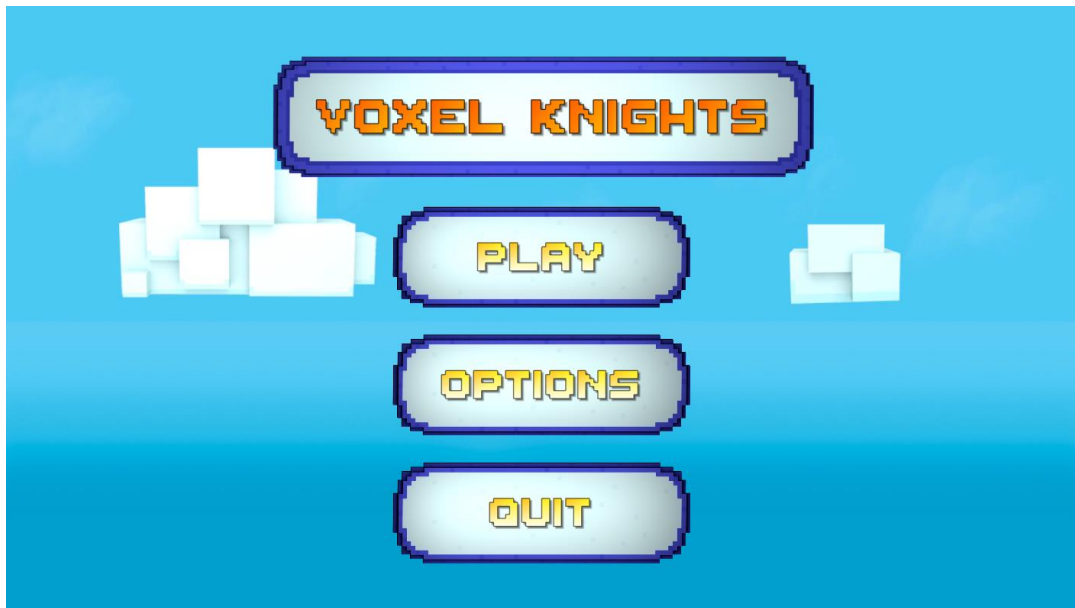


Figura 42. Menú de inicio.

6.2.4.2 Elementos no diegéticos

Todos los elementos flotantes que rodean la pantalla durante los niveles son diegéticos, y tienen la finalidad tanto de otorgar la información necesaria al jugador, como la finalidad de permitir una navegación correcta por el juego.



Figura 43. Alrededor de la pantalla se sitúan los elementos no diegéticos.

Estos elementos están separados claramente en dos zonas: la parte superior y la inferior.

En la parte superior se sitúan los elementos no indispensables para la realización del nivel: a la izquierda, en gris, se pueden observar los tres botones que permiten controlar la cámara, girándola (las flechas curvas) o cambiando a una vista cenital.

En el otro extremo de la parte superior, se pueden observar dos botones rojos, se ha elegido específicamente este color para remarcar que su uso tiene consecuencias críticas en el juego. Estos botones o bien vuelven al menú principal (la cruz), o reinician el nivel actual (la flecha en círculo).

Estos dos botones se sitúan en este extremo para evitar ninguna interacción errónea por parte del usuario.

En el medio, se observan dos paneles únicamente informativos, con el número de turnos actual y la condición de victoria del nivel actual.

Por último, abajo, se pueden observar dos botones, los más utilizados y esenciales para el juego: “cambiar de unidad” y “terminar turno”. Son coloreados de manera drásticamente diferente para evitar confusión, y además son los elementos interactivos más grandes de la pantalla.

El botón de “cambiar de unidad” desaparece cuando sólo queda una unidad por mover para, una vez más, simplificar la información en pantalla.

Hay dos piezas no diegéticas más: aquellas que no son permanentes en pantalla. La primera solo aparece cuando el jugador coloca el cursor encima de un personaje.

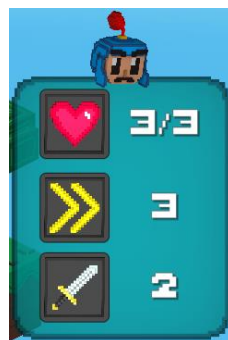


Figura 44. Panel de estadísticas de un caballero

Este elemento muestra las estadísticas del personaje bajo el cursor, y aparece a la derecha de la pantalla. Su funcionalidad reside en ayudar al jugador a entender las capacidades de cada personaje.

La decisión de que no sea un objeto permanente, es para evitar la desubicación del jugador en cuanto a qué personaje pertenecen las estadísticas mostradas. Además, para reforzar la claridad, cambia tanto el color como el icono superior en función de qué tipo de unidad se esté seleccionando.

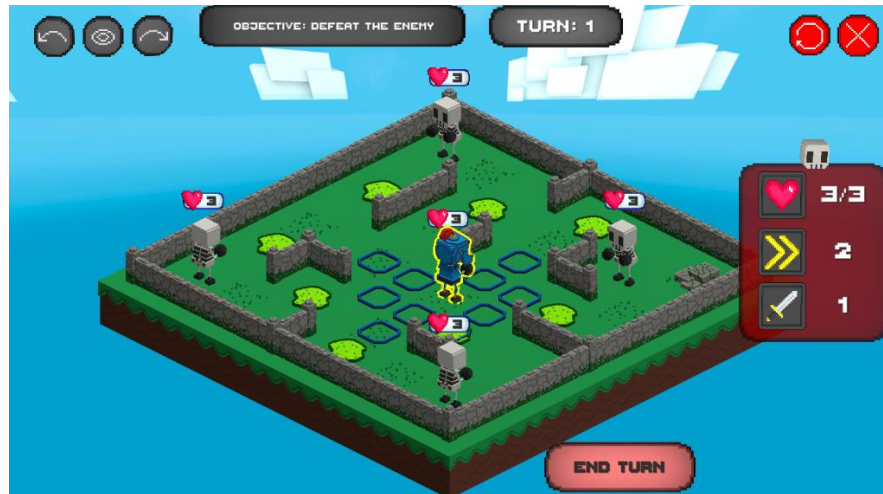


Figura 45. Ejemplo de estadísticas de esqueleto en pantalla.

Por último, quedan los paneles de inicio y final del nivel, que sirven como transiciones para indicar tanto el objetivo, como el éxito/derrota dentro del mismo.

Al igual que el bloque de estadísticas, estos paneles tienen un ligero grado de transparencia, esto es para dar cierta visibilidad al jugador de lo que hay detrás y suavizar la implementación del elemento dentro del entorno.

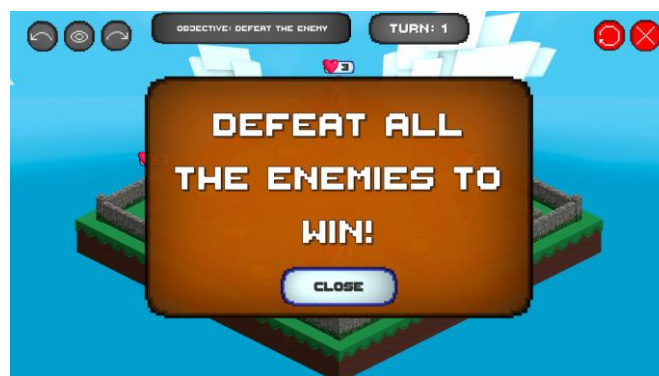


Figura 46. Panel de inicio del nivel.

6.2.4.3 Elementos diegéticos

Se ayuda al usuario a comprender rápidamente el entorno de manera diegética: los objetos decorativos son los que marcan las casillas no accesibles, a excepción del césped y las piedras planas.

Los objetos de césped y de piedras planas sirven para decorar, pero también tienen otra función muy importante: ante la ausencia de una rejilla que marque cada casilla, estos objetos se utilizan como “marcadores” del tablero, sirviendo así de ayuda visual a la hora de ver la localización y tamaño de las casillas.



Figura 47. La existencia del césped y piedras ayuda a la comprensión del entorno.

Las piedras planas, además, tienen un segundo propósito, el cual es indicar conexión entre una casilla botón, y una casilla pared en aquellos niveles en los que hay más de un par de estos.

En la siguiente imagen se muestra como se implementa esto en el nivel 2.3, en el que hay tres pares de casillas pared-botón: las piedras planas unen dos de ellos, permitiendo al jugador saber por descarte como está conectado el último par.

El sonido es el otro componente diegético del juego, y sirve para indicar la ocurrencia de ciertos eventos, estos son: Se ha realizado un ataque, un personaje ha recibido daño por ácido, y se ha activado/desactivado una casilla pared.

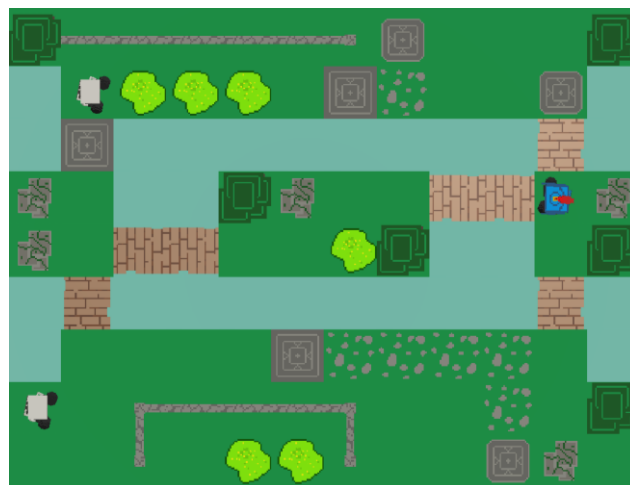


Figura 48. Muestra de la decoración usándose de guía.

6.2.4.4 Elementos espaciales

Son todos no diegéticos, aunque con el añadido de que la posición de estos es relativa al resto de objetos del nivel.

El primero y más importante de todos, es el indicador de movimiento de los personajes. Estos aparecen cuando un personaje se puede mover o se está moviendo, y señalan dónde es posible hacerlo.

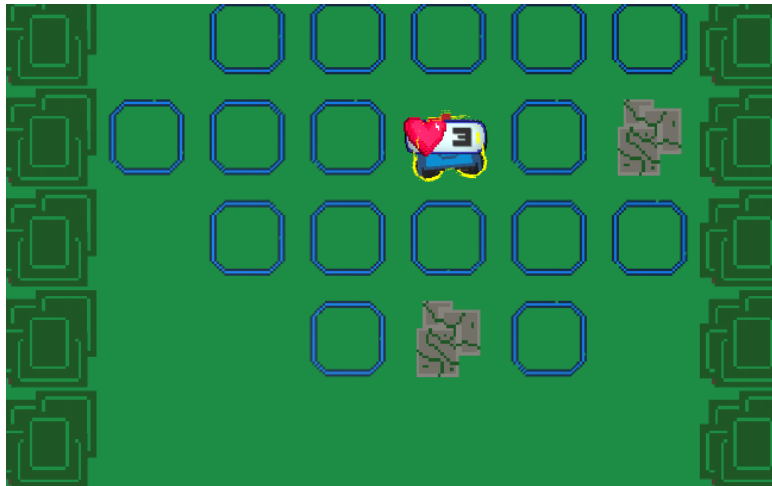


Figura 49. Ejemplo de los indicadores de movimiento.

Otro objeto espacial de la interfaz, y directamente ligado a este último, es la “flecha del camino” (o “*path arrow*”) que, en función de la posición del cursor, mostrará el camino que seguirá el personaje cuando el jugador elija una casilla.

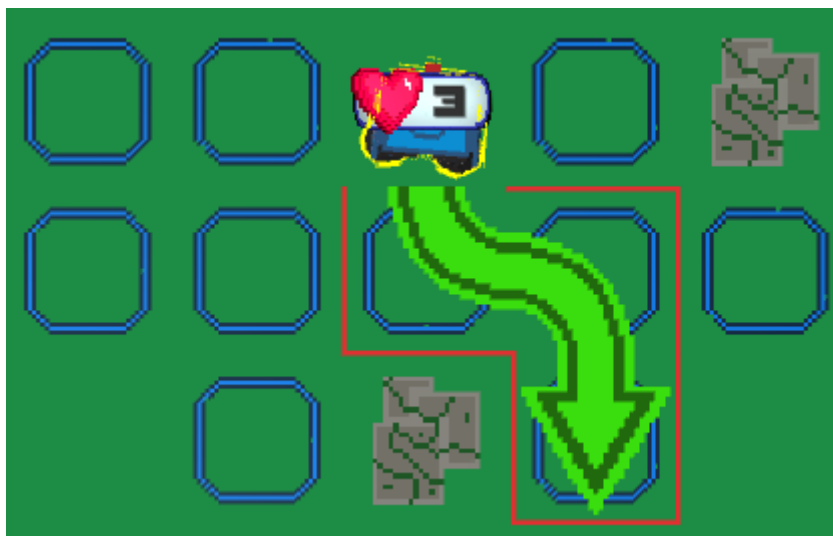


Figura 50. “Arrow path” en funcionamiento.

Como complemento al bloque de estadísticas que se trataba en el apartado de elementos no diegéticos, encima de cada personaje está indicada la cantidad de puntos de vida actuales de cada personaje. Pese a que esto es información redundante, al ser la única estadística que puede cambiar durante el juego, es necesario que se encuentre todo el rato en pantalla y mostrada claramente para cada personaje.

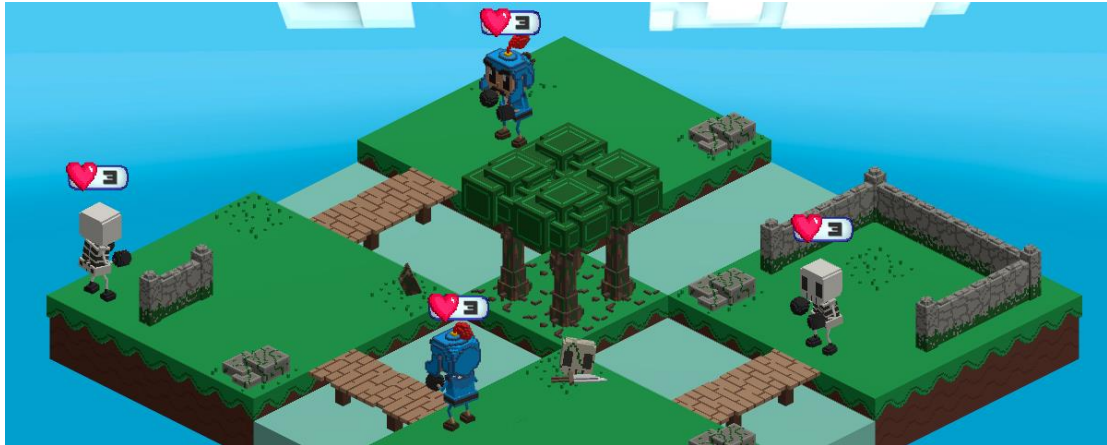


Figura 51. Los puntos de vida de cada personaje se muestran en todo momento.

El último elemento *voxel art* de la interfaz es el indicador de ataque, el cual aparece encima de los enemigos seleccionables para atacar por cada soldado.



Figura 52. Indicadores de ataque sobre los esqueletos seleccionables.

En muchas de las imágenes se puede ver el último objeto que se utiliza para mejorar la interacción del usuario con el entorno: el contorno.

Éste aparece alrededor de un personaje cuando ocurren ciertas situaciones específicas, y sirve para dar *feedback*¹⁰ al jugador respecto a dicha situación, tanto con su color, como su grosor.

Estos casos son: Cuando se está manipulando a un caballero (contorno amarillo); cuando un caballero ha finalizado su turno (contorno negro); cuando el cursor se encuentra encima de un esqueleto que puede ser atacado (contorno rojo); y cuando un personaje sufre daño por ácido (contorno verde, grueso, y momentáneo).

El sistema de generación y manipulación del contorno no es propio, y se puede encontrar en la tienda de *assets* de *Unity*¹¹.

¹⁰ Información que el jugador recibe por parte del juego.

¹¹ Quick Outline: <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

6.3 Programación

Toda la programación del juego se ha realizado en *Unity*, utilizando el lenguaje C#, en este apartado, se muestra mediante diagramas el funcionamiento de los sistemas implementados más importantes del juego,

- **Flujo del modo de juego estándar dentro de la demo:** A continuación, se muestra gráficamente el orden de selección de niveles dentro de la demostración:

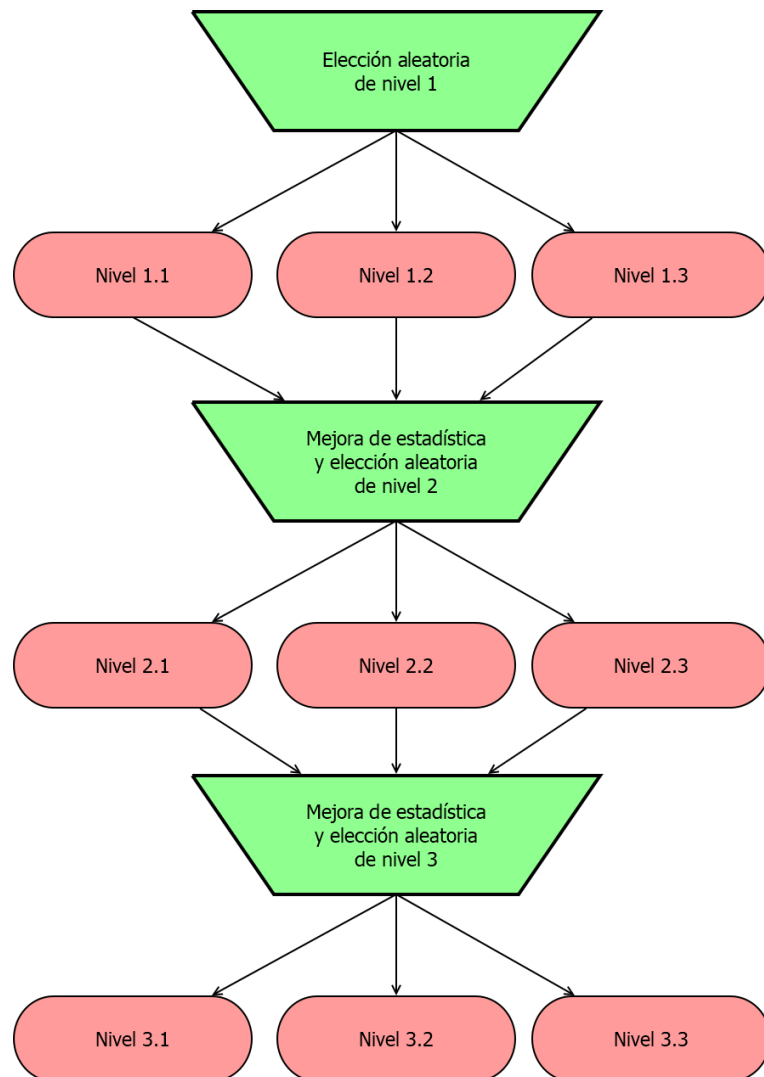


Figura 53. Diagrama de flujo del modo Standard dentro de la demo

-Flujo del turno de los caballeros: Cuando el turno del jugador comience, podrá realizar las acciones que se indican en el siguiente diagrama de flujo, el turno terminará cuando todos los caballeros hayan llegado al fin de su turno.

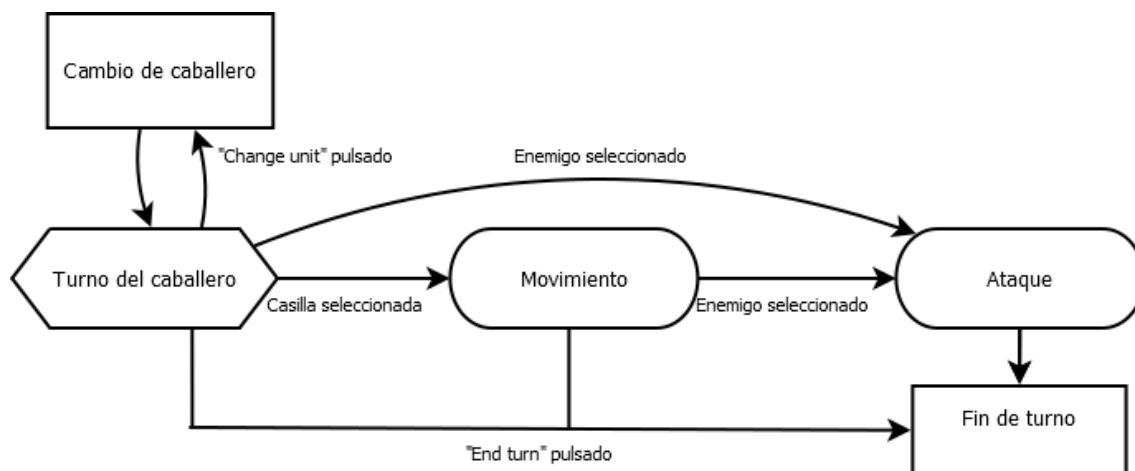


Figura 54. Diagrama de flujo de cada caballero.

- Flujo del turno: Las acciones que puede realizar el enemigo son las mismas en principio, pero al ser controlado por el ordenador el diagrama de flujo cambia de la siguiente manera.

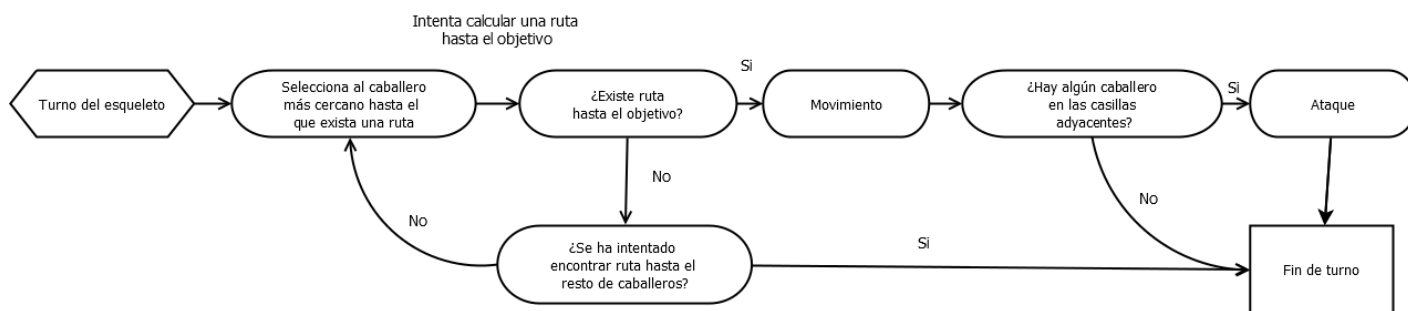


Figura 55. Diagrama de flujo del turno de cada esqueleto

- Creación del camino que sigue un personaje al seleccionar una casilla para moverse: Este camino es creado mediante un algoritmo *Breadth First Search* o de “Búsqueda en anchura”. Explicado de manera simple, este algoritmo recorre la lista de todas las casillas alcanzables, en función del movimiento del jugador, y busca el camino a cada casilla que requiera pasar por menos casillas. Muchas veces habrá varios caminos

de la misma “longitud”, en cuyo caso siempre elegirá el primero calculado, por eso el camino no es modificable.

- **Algoritmo utilizado para la IA de los esqueletos:** Para encontrar el camino hasta cada caballero, los esqueletos utilizan el algoritmo A*, ya que el *Breadth First Search* es demasiado ineficiente para buscar caballeros a lo largo de todo el mapa.

La ventaja del algoritmo A* frente al *Breadth First Search* es la utilización de la función de evaluación para cada casilla, lo cual hace la búsqueda mucho más rápida. A continuación, comparo como trabajan ambos algoritmos para una misma escena. En gris tenemos los obstáculos que bloquean el camino, en verde la casilla de origen, en rojo la casilla final, y las casillas oscurecidas son todas las casillas que han sido evaluadas hasta encontrar el camino. Esta simulación ha sido realizada en Github, en la página del usuario Qiao¹².

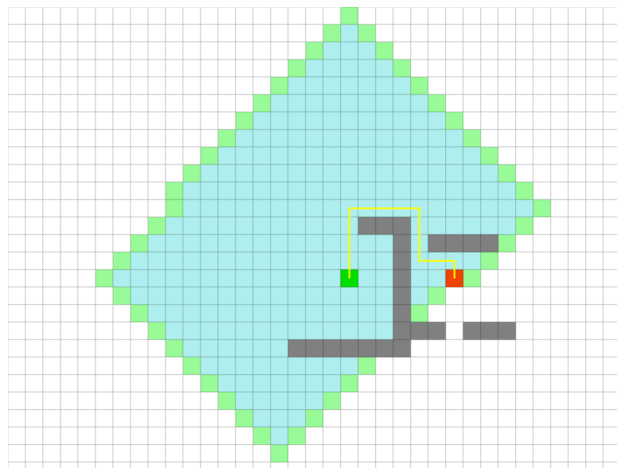


Figura 56. Simulación con *Breadth First Search*

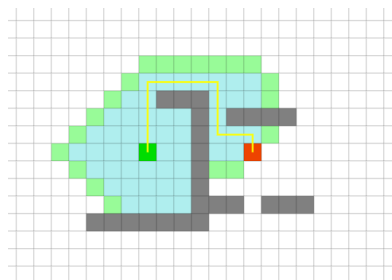


Figura 57. Simulación con A*

¹² <https://qiao.github.io/PathFinding.js/visual/>

6.4 Postproducción

6.4.1 Ambientación e iluminación.

El sistema de iluminación es muy simple: se utiliza una luz direccional para iluminar de manera suave la escena y se han eliminado las sombras, para que no de la sensación de venir de ningún punto en concreto, y la manera en que se vea la escena no cambie al rotar la cámara.

6.4.2 Sonido

El sonido en *Voxel Knights* sirve un rol complementario: no es indispensable, pero acompaña a los elementos de la pantalla y da *feedback* al jugador en ciertos eventos.

La música utilizada ha sido sacada de la página web *Incompetech* (MacLeod, s.f.), y los sonidos de la biblioteca de sonidos virtual *Freesound* (de Jong, 2005). Algunos de los sonidos han sido editados ligeramente en *Adobe Premiere*, cambiando únicamente longitud de los audios y volumen.

Conclusiones

Voxel Knights es un juego que utiliza ideas ya existentes, para formar un concepto nuevo y original, aportando una nueva entrada al mundo de los videojuegos *voxel art*.

El objetivo era crear una demostración que funcionase de principio a fin, completa, y capaz de mostrar lo que “podría ser” *Voxel Knights*, mediante el empleo de una cantidad de contenido limitada.

Se cree que esto se ha logrado, que se ha obtenido un producto sólido durante el cual el manejo de los distintos procesos que forman el desarrollo de un videojuego ha mejorado drásticamente.

Glosario

blocking

En diseño de niveles, el blocking hace referencia a la fase de maquetación de un nivel, en la que se crea estructuralmente con formas básicas, para poder probarlo a nivel funcional., 24

feedback

Información que el jugador recibe por parte del juego, 61, 65

gameplay

Es la manera específica en la que un jugador interactúa con el juego, este patrón se define a través de las reglas del juego., 13

idle

Animación de un personaje en reposo, para evitar que dé una sensación totalmente estática., 54

rigging

Hace referencia a todos los procesos de preparación necesarios para poder animar un modelo 3D utilizando animación con huesos., 10, 11, 53

roguelike

Los videojuegos de mazmorras (conocidos comúnmente como roguelike), son un subgénero de los videojuegos de rol que se caracterizan por una aventura a través de niveles y desafíos generados aleatoriamente. Este género nació a partir del juego homónimo Rogue (Toy, Wichman, & Arnold, 1980)., 10, 11, 12, 13, 14

RPG

Los Role Playing Games, o RPGs, son juegos en los que el usuario controla a uno o varios personajes inmersos en un mundo detallado, generalmente con opciones de personalización y mejora de estos., 17

target

En publicidad, el target hace referencia al público objetivo del producto, el cual es designado en función de una serie de cualidades y características., 17

Bibliografía

Ancel, M. (9 de Septiembre de 1995). Rayman. París, Francia.

de Jong, B. (5 de Abril de 2005). *Freesound*. Obtenido de <https://freesound.org/>

Epic Games. (6 de Diciembre de 2018). *Epic Games Store*. Obtenido de Epic Games Store: <https://www.epicgames.com/>

Game Programming Academy. (9 de Noviembre de 2017). *Youtube*. Obtenido de Youtube: https://www.youtube.com/watch?v=cX_KrK8RQ2o

Gollop, J. (31 de Diciembre de 1993). X-COM. Baltimore, Maryland, Estados Unidos.

- Hoppmann, R. (24 de Febrero de 2019). Thaleah Fat.
- Kaga, S. (20 de Abril de 1990). Fire Emblem. Uji, Kioto, Japón.
- Konami. (1983). Bomberman.
- Ma, J., & Davis, M. (27 de Febrero de 2018). Into the Breach.
- MacLeod, K. (s.f.). *Incompetech*. Obtenido de <https://incompetech.com/>
- Mcmillen, E. (28 de Septiembre de 2011). The Binding of Isaac. Santa Cruz, California, Estados Unidos.
- Nolet, C. (16 de Julio de 2018). *Unity*. Obtenido de <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>
- Persson, M. (18 de Noviembre de 2011). Minecraft. Estocolmo, Suecia.
- Rao, A. (6 de Diciembre de 2018). Hades. San Francisco, California, Estados Unidos.
- Soares, Z. (22 de Septiembre de 2020). *Domestika*. Obtenido de Domestika: <https://www.domestika.org/es/courses/1404-introduccion-al-voxel-art-para-el-diseno-de-personajes>
- Söderberg, J., & Lucken, G. J. (s.f.). Voxel Tactics. Londres, Reino Unido.
- Toy, M., Wichman, G., & Arnold, K. (1980). Rogue. Berkeley, California, Estados Unidos.
- Valve Corporation. (12 de Septiembre de 2003). *Steam*. Obtenido de Steam: <https://store.steampowered.com/>
- von Funck, W. (30 de Septiembre de 2019). Cube World. Estocolmo, Suecia.

Anexo I. Índice de figuras.

Figura 1. Imagen de gameplay de Into the Breach (Ma & Davis, 2018).....	13
Figura 2. Minecraft emplea polígonos simples, detallados a través de la textura. ...	14
Figura 3. Pantalla de inicio de Cube World.	15

Figura 4. GANTT del desarrollo del proyecto.	16
Figura 5. DAFO.....	17
Figura 6. Blocking del nivel 3.1.	25
Figura 7. Nivel Tutorial 1.	26
Figura 8. Nivel Tutorial 2.	27
Figura 9. Nivel Tutorial 3.	28
Figura 10. Nivel Tutorial 4.	29
Figura 11. Nivel Tutorial 5.	30
Figura 12. Nivel Tutorial 6.	31
Figura 13. Nivel 1.1 Standard.....	34
Figura 14. Nivel 1.2 Standard.....	35
Figura 15. Nivel 1.3 Standard.....	36
Figura 16. Nivel 2.1 Standard.....	37
Figura 17. Nivel 2.2 Standard.....	38
Figura 18. Nivel 2.3 Standard.....	39
Figura 19. Nivel 3.1 Standard.....	40
Figura 20. Nivel 3.2 Standard.....	41
Figura 21. Nivel 3.3 Standard.....	42
Figura 22. Pruebas diferentes personajes de baja resolución.	43
Figura 23. Prueba diferentes personajes de alta resolución.....	43
Figura 24. Pruebas de animación para baja resolución.	44
Figura 25. Prueba de personaje de mayor resolución con arma	44
Figura 26. Comparación ilustrativa de la diferencia de resolución.	45
Figura 27. Diseño final del caballero.	46
Figura 28. Diseño final esqueleto.	46
Figura 29. Bloques estándar + bloque de agua.....	47
Figura 30. Casillas de césped vistas desde abajo.....	48
Figura 31. Casilla móvil.	48
Figura 32. Casillas botón y pared.	48
Figura 33. Casilla dañina.	49
Figura 34. Modelos utilizados en la animación del ácido.	49
Figura 35. Assets decorativos.....	49
Figura 36. Elementos de la UI en Qubicle.	50
Figura 37. Archivo Photoshop del Skybox.....	51

Figura 38. Skybox en Unity.....	51
Figura 39. Rig del caballero	52
Figura 40. Espada generándose en mitad del ataque.	53
Figura 41. Espada desapareciendo al final del ataque.	54
Figura 42. Menú de inicio.	55
Figura 43. Alrededor de la pantalla se sitúan los elementos no diegéticos.	55
Figura 44. Panel de estadísticas de un caballero	56
Figura 45. Ejemplo de estadísticas de esqueleto en pantalla.	57
Figura 46. Panel de inicio del nivel.	57
Figura 47. La existencia del césped y piedras ayuda a la comprensión del entorno.	58
Figura 48. Muestra de la decoración usándose de guía.	58
Figura 49. Ejemplo de los indicadores de movimiento.	59
Figura 50. “Arrow path” en funcionamiento.	59
Figura 51. Los puntos de vida de cada personaje se muestran en todo momento.	60
Figura 52. Indicadores de ataque sobre los esqueletos seleccionables.	60
Figura 53. Diagrama de flujo del modo Standard dentro de la demo.	62
Figura 54. Diagrama de flujo de cada caballero.	63
Figura 55. Diagrama de flujo del turno de cada esqueleto	63
Figura 56. Simulación con Breadth First Search	64
Figura 57. Simulación con A*	64

Anexo II. Actividades realizadas durante la investigación

Para el desarrollo de los sistemas más básicos del juego, como la manipulación y el movimiento en el tablero, se han utilizado vídeo tutoriales del canal *Game Programming Academy*. (Game Programming Academy, 2017)

En cuanto al *voxel art* durante la investigación se ha recogido información de múltiples fuentes, pero la primera, y más influyente, fue el curso de *Domestika* sobre animación de personajes en *voxel art* de Zach Soares. (Soares, 2020)

Anexo III. Requerimientos de licencia de música.

Ethernight Club by Kevin MacLeod

Link: <https://incompetech.filmmusic.io/song/7612-ethernaut-club>

License: <https://filmmusic.io/standard-license>.

Thatched Villagers by Kevin MacLeod

Link: <https://incompetech.filmmusic.io/song/4481-thatched-villagers>

License: <https://filmmusic.io/standard-license>.