



AT NIGHTFALL

Portafolio – Muestra de IA

Nombre de los integrantes:

Huerta Estaragués Eduardo Damián

Fecha del Documento:
07/06/2023



Investigación:

¿Qué es un algoritmo?

Un algoritmo, es una parte esencial de las matemáticas, lógica, ciencias de la computación y sus relacionados. Siendo definido como un conjunto de instrucciones o condiciones (reglas), puestas en un orden en una estructura finita, permitiendo resolver un problema o acción. Siendo el enfoque principal de la “algoritmia”.

La etimología de esta palabra, de acuerdo con la Wikipedia es: “La palabra castellana algoritmo deriva del latín algoritmos, que se acuñaría en el siglo XV a partir del latín algorismus, con influencia del griego arithmos, que significa «número», o de la latinización del apellido del matemático persa Al-Juarismi. La RAE, por su parte, propone que deriva del latín algobarismus, que derivaría del árabe clásico ḥisābu lġubār, que significa «cálculo mediante cifras arábigas».” (colaboradores de Wikipedia, 2024f)

Utilidad de los algoritmos

Los algoritmos no son algo extraño para nadie, pues los empleamos sin siquiera saberlo. Desde realizar cálculos simples conllevan algoritmos, como toda acción que conlleve una orden o lógica de acción específica.

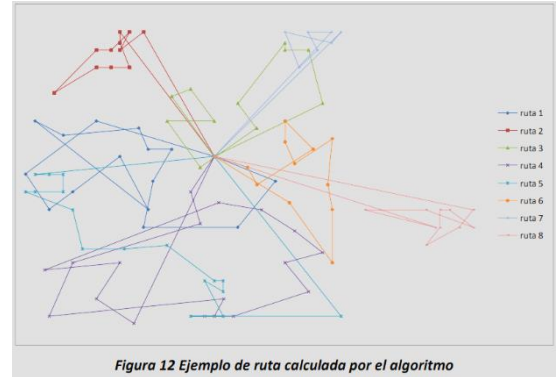
Sin embargo, en el ámbito más informático, son el fundamento de la mayoría de los sistemas. Pues, para poder hacer sistemas que funcionen en múltiples casos y para poder generar sistemas. Siendo importantes en cualquier sistema de cálculo, funciones, sistemas operativos y cualquier sistema de uso general. Y actualmente se ha popularizado mucho el termino en el ámbito de videojuegos, con términos como proceduralidad, automatización y sistemas de comportamientos.

Y en el presente y futuro solo se vuelven más y más importantes. Con su relevancia informática a futuro y el auge actual de la IA y la importancia de generación de estas soluciones de problemas.



Principales algoritmos en los videojuegos

Si bien en la actualidad, una gran parte de los juegos emplean sistemas como procedural, IA, automatización, auto generación, etc. Vamos a profundizar en algoritmos de trazado de rutas y otros algoritmos que son relevantes para la actividad a realizar para este trimestre. Pero, estas otras implementaciones de algoritmos son fundamentales y extremadamente importante.

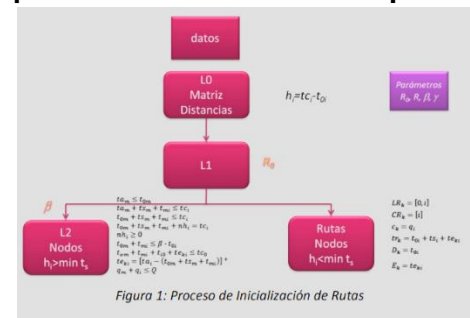


(Arturo Nieto de Almeida et al., s. f.)

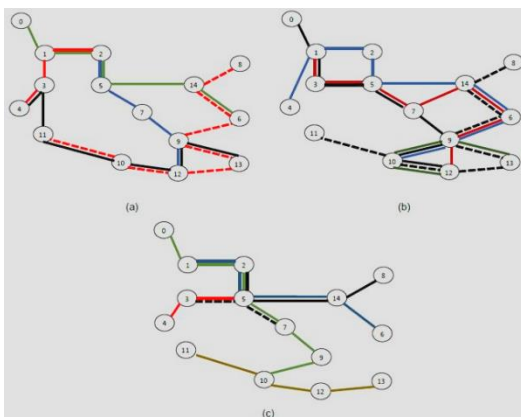
Para el trazado de rutas existen múltiples sistemas y soluciones, el cual se implementa suele depender de las preferencias del creador y lo que se busca hacer. Existen desde programas especializados en generar rutas de movimiento, pero se pueden implementar soluciones propias a la altura de lo que busca hacer.

Si bien existen infinitas soluciones, no todas son viables para videojuegos por tiempos de cálculo y recursos. Por lo que se suele recurrir a los mismos sistemas, como lo es el uso de celdas, puntos de control, sistemas de máquinas de estados y/o machine learning.

Para la optimización de el caso de sistemas de rutas calculadas previo al movimiento o las que se muestran en Mini mapas emplean una técnica en la que se simula una conexión entre puntos y determina cual es el mejor camino bajo algunos parámetros como lo es distancia a objetivo, tiempo, distancia, etc.



(Jiménez-Carrión et al., 2023)



(Jiménez-Carrión et al., 2023)

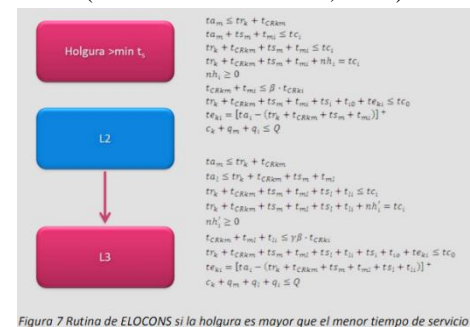
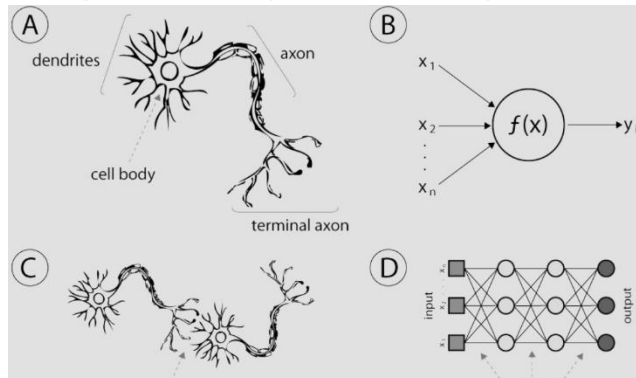


Figura 7 Rutina de ELOCONS si la holgura es mayor que el menor tiempo de servicio

(Jiménez-Carrión et al., 2023) pg. 3

Si bien este proceso puede llegar a ser pesado al no ser realizado de forma constante es bueno para sistemas de turnos, sistemas de mapas, entre otros.

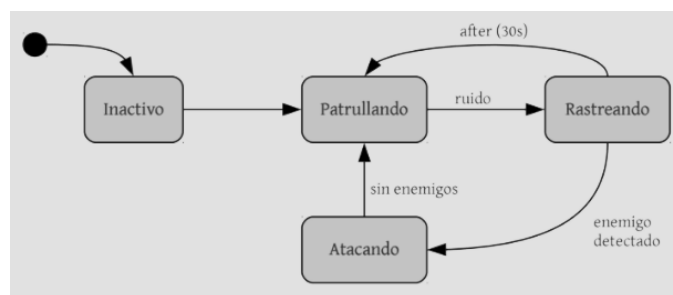
Pero para hacer algo dinámico se puede dar un sistema integrado en cada personaje, simulando sentidos como la vista y percepción espacial. Enseñando le o permitiendo que aprenda de forma autónoma con machine learning. Otra forma es establecer una máquina de estados que recibe algún parámetro y reacciona ante el entorno.



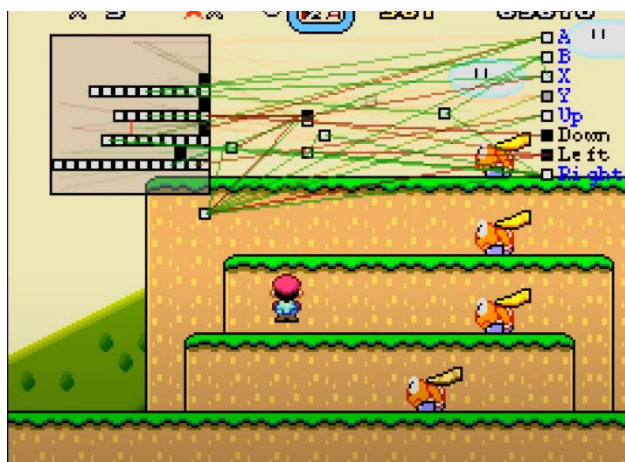
(Ruiz-Azuaga, 2016)

Estos sistemas dinámicos y constantes son útiles en juegos o sistemas cambiantes. Como lo son enemigos en un shooter, para búsqueda de enemigos y ataque, o para generar sistemas de NPC's. En el caso de una máquina de estados es más simple, ligero. Pero hasta un punto puede llegar a ser predecible, a la vez que todo cambio y posibilidad debe estar contemplada en esta estructura para que reaccione de la forma esperado. Mientras que estructuras neuronales son dinámicas y se pueden autorregular con una menor intervención humana, pudiendo cambiar con el tiempo a la vez que resultan en una mayor aleatoriedad o posible complejidad.

Sin embargo, con un buen sistema de cualquiera de ellos se puede lograr un gran resultado, casi irreconocible uno de otro. Sin embargo, hoy en día se ha recurrido a emplearlo como una forma de márketing o que se hable de un proyecto. Lo correcto es hacer un sistema acorde a tus necesidades y del proyecto.



(Ramirez, 2021)



(Ruiz-Azuaga, 2016)



Algoritmos y Teoría por aplicar

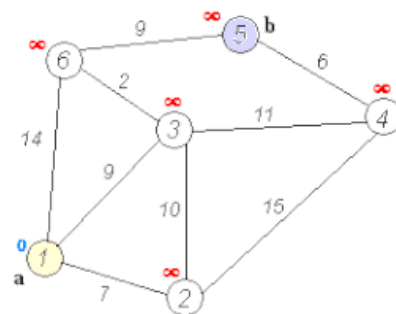
Para el Proyecto de este Trimestre se busca generar un sistema aleatorio de caminos, por lo que métodos como Dijkstra y A^* , son una gran referencia para comprender el funcionamiento básico de la generación de e caminos, pese a estos no ser empleados directamente en el proyecto.

Para empezar, debemos ver lo que es un sistema de Grafo, pues estos sistemas de una manera u otra dependen de estas estructuras para la generación de caminos. Un grafo de forma simple es un conjunto de puntos ligados entre sí, normalmente contando con más de 2 conexiones por punto o nodo. Y en este caso, estas conexiones tienen un valor dado por distancia, preferencias, o algún parámetro para dar el resultado esperado. Al tener múltiples conexiones entre ellos, no hay un único camino de un punto a otro.

Algoritmo Dijkstra

Lo que este algoritmo hace es “explorar” las diferentes rutas, mediante hacer una lista de “valores” o “pesos” y una “final”. La lista de pesos se selecciona los nodos adyacentes de una referencia y se obtiene el “Total” (la suma del “peso” del nodo anterior y el nuevo), en caso de este ser menor al Temporal anterior se sustituye, de lo contrario permanece igual.

Al encontrar el que tenga el “valor” o “peso” más bajo, su peso se vuelve final y se toma ese nodo como la nueva referencia, volviendo a checar sus adyacentes y repitiendo el proceso, sustituyendo en la lista de valores en caso de que mejore el valor previo para ese nodo.



Al final, obtendré un valor con el cual deberemos recorrer los nodos en reversa para encontrar nuestra ruta. Para esto, se resta el “peso” del nodo final con los “pesos” para llegar a los nodos adyacentes, si la resta es equivalente el valor “final” para dicho nodo, ese forma parte de la ruta optima y se repite el análisis tomando los parámetros de ese nuevo nodo.

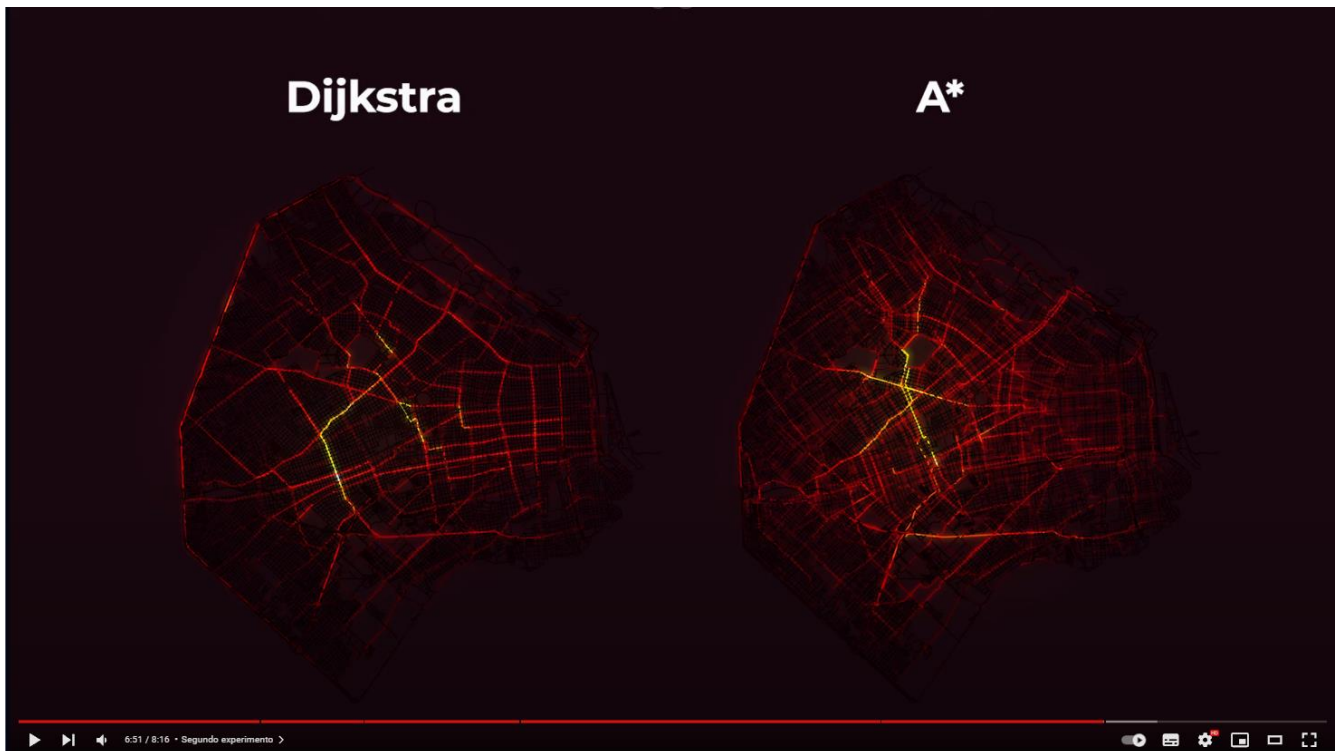


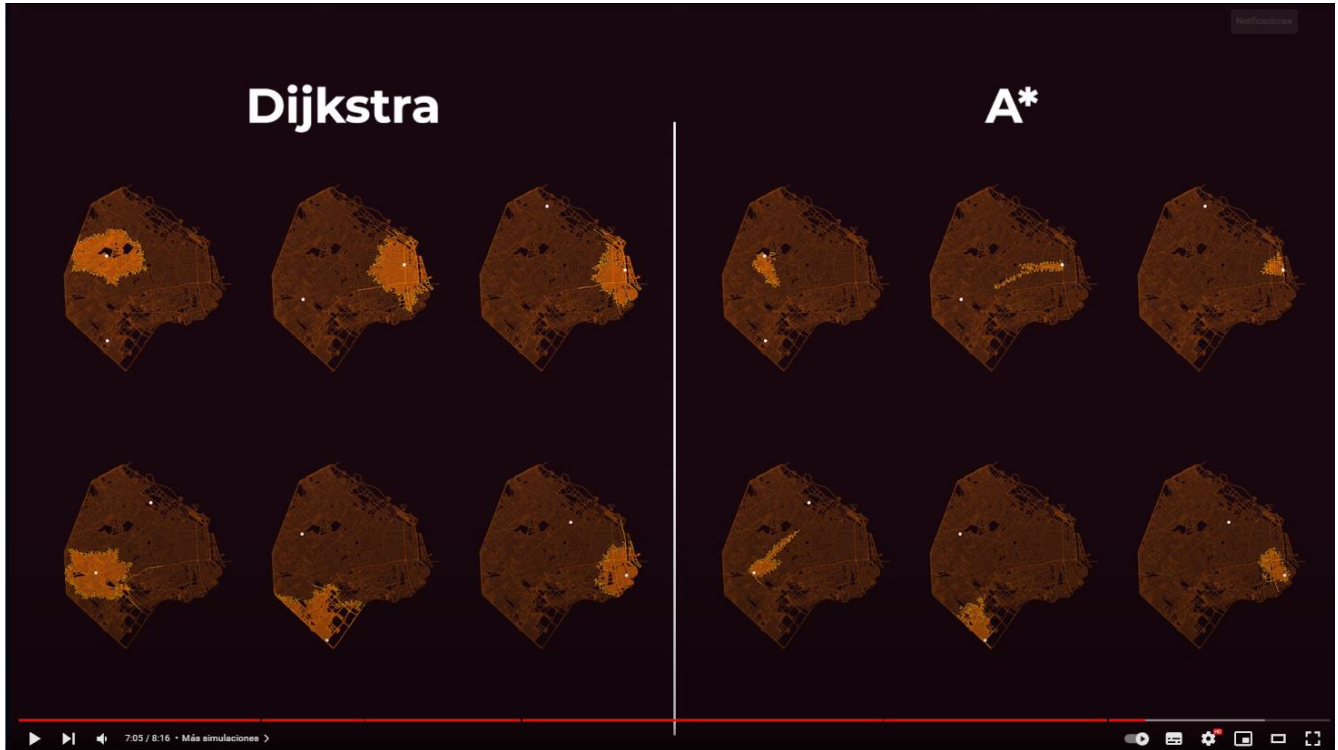
Este es mucho más completo, sin embargo, la forma simple en la que comprendí su funcionamiento es el que este compara y busca los caminos de mejor resistencia a la vez que busca siempre acercarse al punto final.

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9		11	12	13	14
6	5	4	5	6	7	8	9	10		10	11	12	13

Sin embargo, Dijkstra es mucho más pesado, por lo que, porque quería usar Dijkstra sobre A*. De forma simple y rápida, Dijkstra asegura que la ruta que proporciona es la mas optima, aunque requiera mucho más tiempo de ejecución, a la vez que, al hacer una tabla de pesos entre nodos, se puede visitar y calcular rutas optimas para todos los nodos previamente analizados. Como se puede ver y explica en el siguiente video de “Santiago Fiorino”:

<https://www.youtube.com/watch?v=oMgiGkFSgl0>





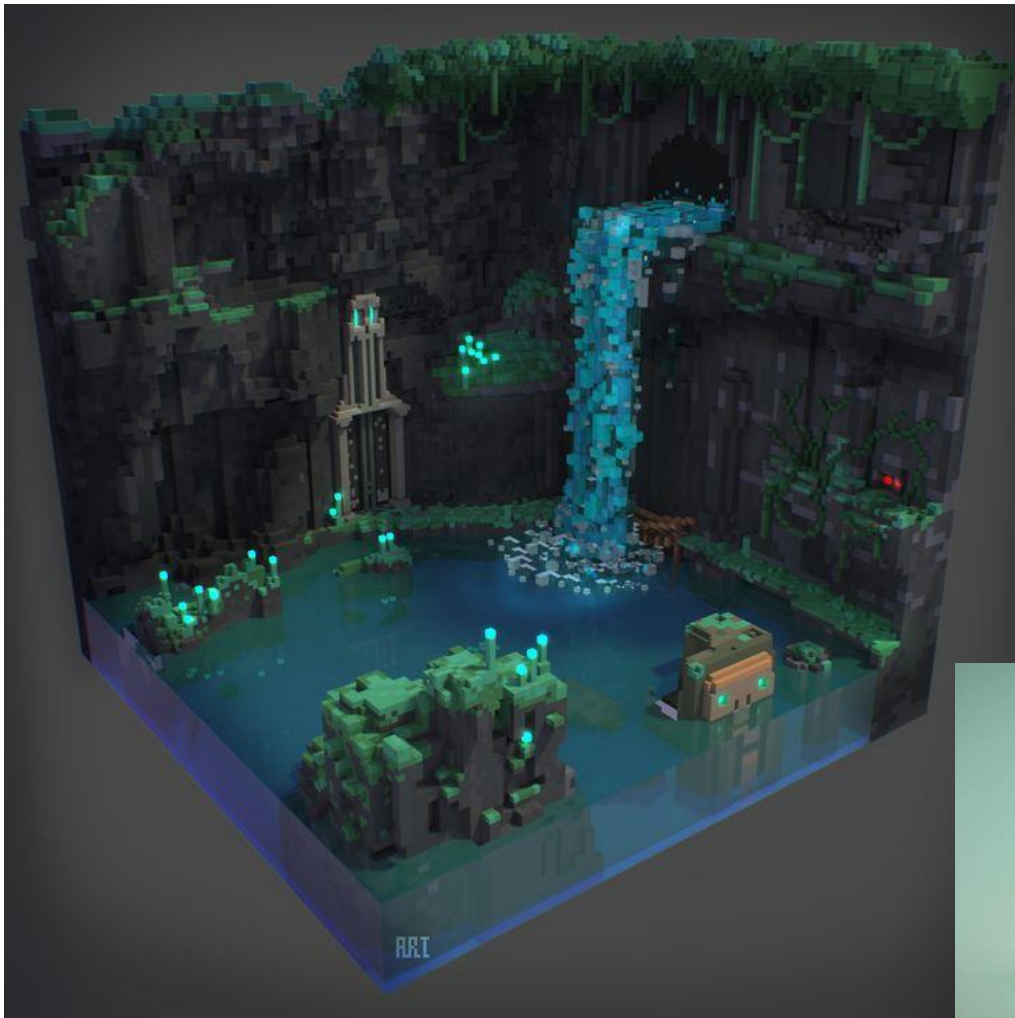
Resumen		
Algoritmo	A*	Dijkstra
Iteraciones	699	9298
Largo [km]	5,94	6,10
Velocidad [km/h]	42,3	50,3
Tiempo [min]	8,4	7,2

(Santiago Fiorino, 2024)

AT NIGHTFALL

Este escrito redactara un planteamiento para un juego del género “Tower defence”, el cual estará inspirado en referentes del género como “Bloons TD” y buscara una estética isométrica a 30° en un estilo de voxels, contando con una atmosfera de fantasía oscura. La narrativa girara entorno a un conjunto de zorros equipados con magia, armas y armaduras. Que buscan proteger su aldea de los monstruos que rondan su mundo mágico. Siendo desde espíritus (de diferentes elementos), No vivientes, osos, aves y otras criaturas mágicas que amenazan con destruir la aldea.

Ahora se expondrán los diferentes elementos del juego separado en Torres, enemigos, mapa, UI y arte.



(K13t2n, 2019)



(Fennec Fox Voxel Style, s. f.) pg. 8



Torretas

Zorro caballero (Costo: ----):

Esta tropa puede atacar a corta distancia a tropas terrestres (medio bloque a la redonda). Emplea una espada para dar cortes alrededor de él, con una animación de jiro. Teniendo un tiempo de 1.0s entre el comienzo de atacar y el siguiente. Infringiendo 10pts de daño.

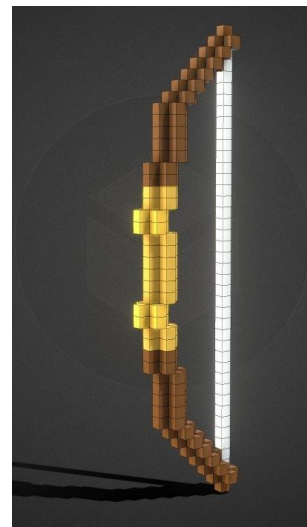


- **Mejoras:**
 - **(Maestría) Reducir tiempo de enfriamiento en un 10%**
 - Reducir tiempo de enfriamiento en un 15%
 - Remover tiempo de enfriamiento.
 - **(Filo de hierro) Incrementar daño de ataque en un 8%**
 - Incrementar daño de ataque 12%
 - Incrementar daño de ataque en un 15%
 - **(Espada larga) Incrementar distancia de ataque +1/3 (de lo inicial)**
 - Incrementar distancia de ataque +1/3 (de lo inicial)
 - Incrementar distancia de ataque +1/3 (de lo inicial)



Zorro arquero (Costo: ----):

Esta tropa puede atacar a media distancia a tropas aéreas o terrestres (un bloque a la redonda). Emplea un arco para disparar flechas (simples) alrededor de él, con una animación de jalar y soltar arco. Teniendo un tiempo de 0.5s entre el comienzo de atacar y el siguiente. Infringiendo 10pts de daño.



(Sketchfab, s. f.-b)

- **Mejoras:**

- **(Maestría) Reducir tiempo de enfriamiento en un 20%**
 - Reducir tiempo de enfriamiento en un 25%
 - Reducir tiempo de enfriamiento a 0.25s
- **Puntas especiales:**
 - **(Puntas de Fuego) Incrementar daño de ataque en un 10% + 1pts/s de daño por 3 segundos**
 - **(Puntas de Fuego II) Incrementar daño de ataque 20% + 3pts/s de daño por 5 segundos**
 - **(Bola de fuego) Incrementar daño de ataque 25% + 5pts de daño por 5 segundo**
 - **(Espigas de hielo) Incrementar daño de ataque en un 10% + congelación del enemigo por 3 segundos**
 - **(Espigas de hielo II) Incrementar daño de ataque en un 20% + congelación del enemigo por 5 segundos**
 - **(Bola de hielo) Incrementar daño de ataque en un 25% + congelación del enemigo por 8 segundos**
- **(Vista fina) Incrementar distancia de ataque +1/4 (de lo inicial)**
 - **(Binoculares) Incrementar distancia de ataque +1/4 (de lo inicial)**
 - **(Vista de Alcón) Incrementar distancia de ataque +1/4 (de lo inicial)**



Zorro cañonero (Costo: ----):

Esta tropa puede atacar a larga distancia a tropas aéreas o terrestres (dos y medio bloques a la redonda). Emplea un arco para disparar bolas de cañón (simples) alrededor de él, con una animación de explosión. Teniendo un tiempo de 1.5s entre el comienzo de atacar y el siguiente. Infringiendo 25pts de daño.



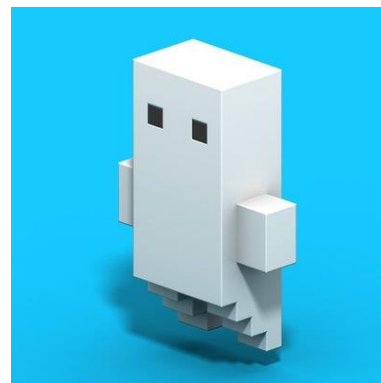
- **Mejoras:**
 - **(Maestría) Reducir tiempo de enfriamiento en un 10%**
 - Reducir tiempo de enfriamiento en un 15%
 - Reducir tiempo de enfriamiento a 1s
 - **Balas especiales:**
 - **(Bolas de Fuego) Incrementar daño de ataque 10% + 5pts de daño a las tropas que estén a medio bloque de distancia**
 - **(Bolas de Fuego II) Incrementar daño de ataque 15% + 8pts de daño a las tropas que estén a medio bloque de distancia**
 - **(Bombas) Incrementar daño de ataque 20% + 10pts de daño a las tropas que estén a medio bloque de distancia**
 - **(Bollas de hielo) Incrementar daño de ataque 10% + congelación del enemigo por 3 segundos**
 - **(Bolas de hielo II) Incrementar daño de ataque 15% + congelación del enemigo por 5 segundos**
 - **(Bolas de hielo III) Incrementar daño de ataque 20% + congelación del enemigo por 8 segundos**
 - **(Puntería fina) Incrementar distancia de ataque +10% (de lo inicial)**
 - **(Binoculares) Incrementar distancia de ataque +10% (de lo inicial)**
 - **(Vista de Alcón) Incrementar distancia de ataque +10% (de lo inicial)**



Enemigos

Espíritus (Terrestres)

- **Base (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .25bloques por segundo, contando con 10pts de vida.
- **Fuego (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 20pts de vida.
- **Agua (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 20pts de vida.
- **Piedra (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 25pts de vida.
- **Hielo (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 25pts de vida.



Esqueletos (Terrestre)

- **Base (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5bloques por segundo, contando con 15pts de vida.
- **Armadura (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .25 bloques por segundo, contando con 30pts de vida.





Araña (Terrestre)

- **Base (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de 1 bloques por segundo, contando con 10pts de vida.
- **Armadura (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 20pts de vida.



(Voxel Spider, s. f.)

Oso (Terrestre)

- **Base (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 35pts de vida.
- **Armadura (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .25 bloques por segundo, contando con 50pts de vida.
- **Alfa (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 50pts de vida.



Ada (Aéreo)

- **Base (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 20pts de vida.
- **Maestra (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de .5 bloques por segundo, contando con 40pts de vida.

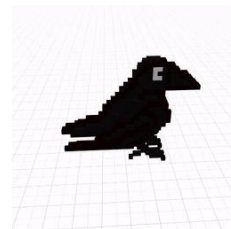


(Introduction To Voxel, 2017)
pg. 13



Cuervo (Aéreo)

- **Base (Valor: ---):**
 - Este enemigo recorre el camino a una velocidad de 1 bloques por segundo, contando con 20pts de vida.



(OpenSea, s. f.)

Mapa

Este se generará de forma procedural al comenzar cada partida y los enemigos recorrerán la ruta hasta el objetivo. El objetivo contara con 200puntos de vida, perdiendo uno por cada punto de vida que tengan los enemigos al alcanzar el objetivo.

Se colocará la entrada del lado izquierdo y el objetivo estará a la derecha, facilitando una visualización acorde con la lectura más popular. A la vez que se buscara mediante condiciones de generación tener una distancia de caminos semejante entre las interacciones, a la vez que siempre tener múltiples lugares donde quepan las torres.

Para mejor control de los espacios, se buscará un sistema de cuadrículas, el cual una tropa ocupará $\frac{1}{4}$ de cuadro o 1 cuadro para habitar, mientras que los enemigos ocuparan $\frac{1}{4}$ o $\frac{1}{2}$ de bloque para habitar.



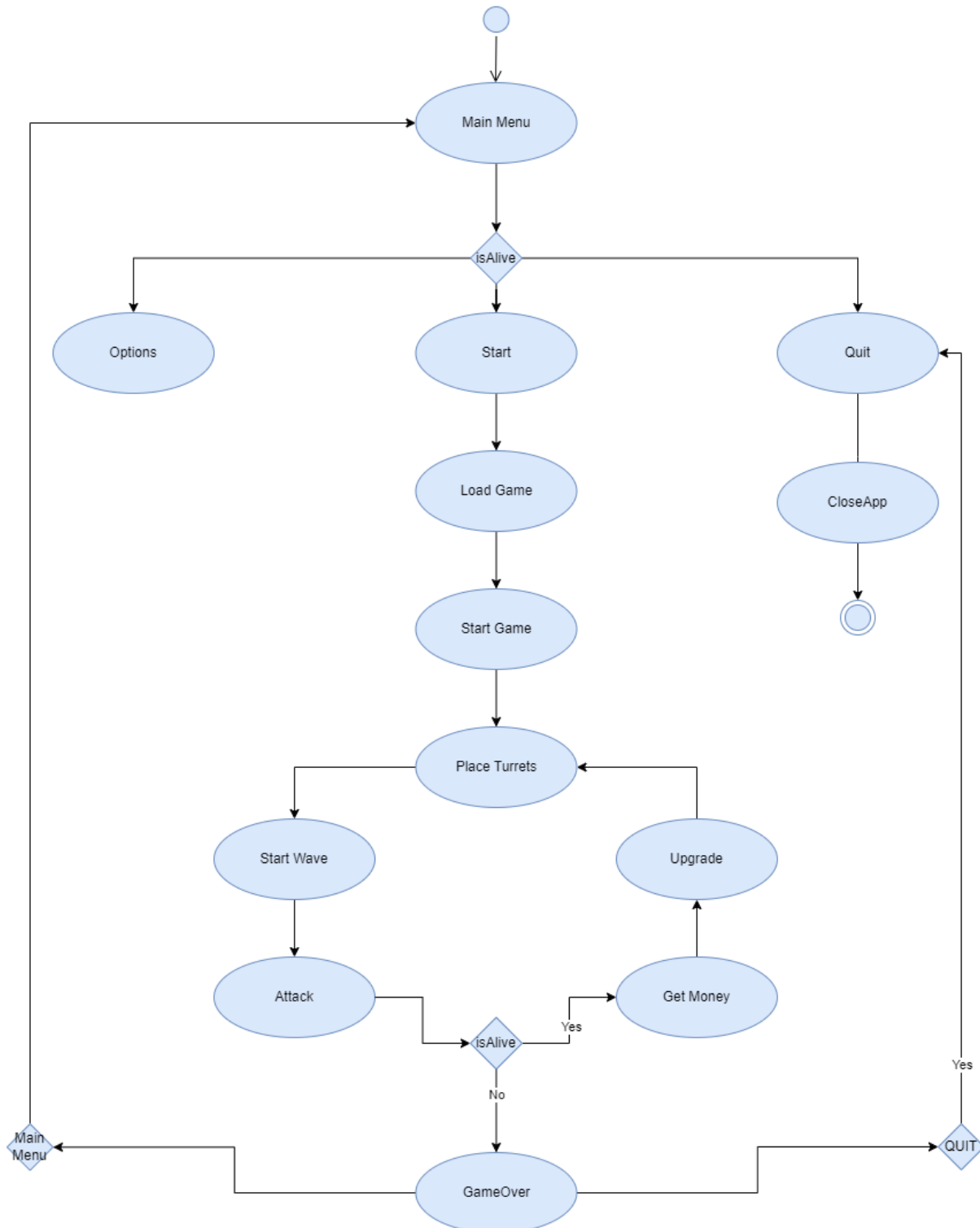
UI





Diagramas UML

GameLoop



Arte



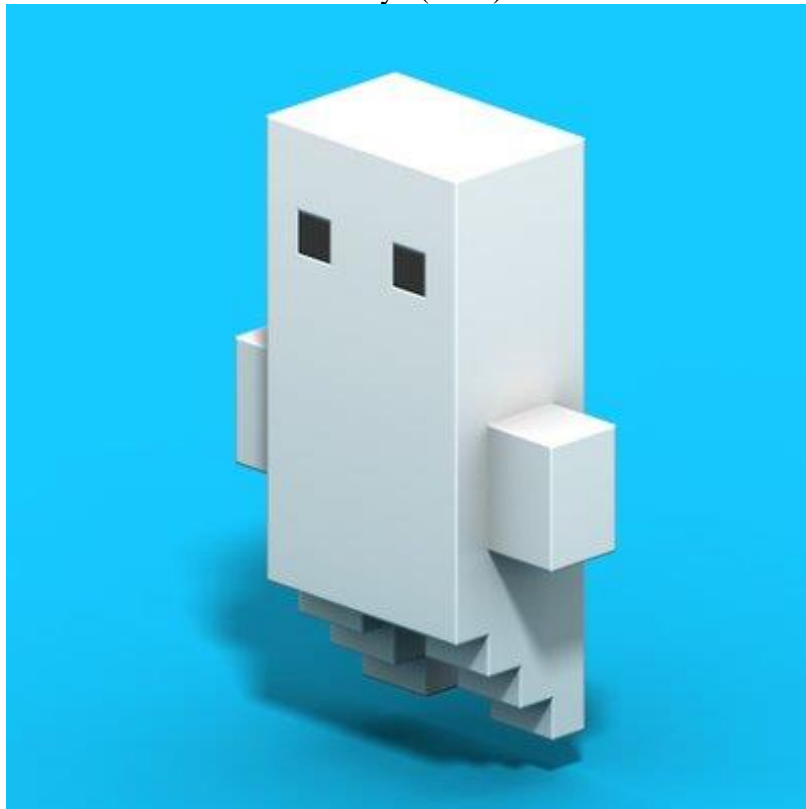
(Android Authority, 2015)



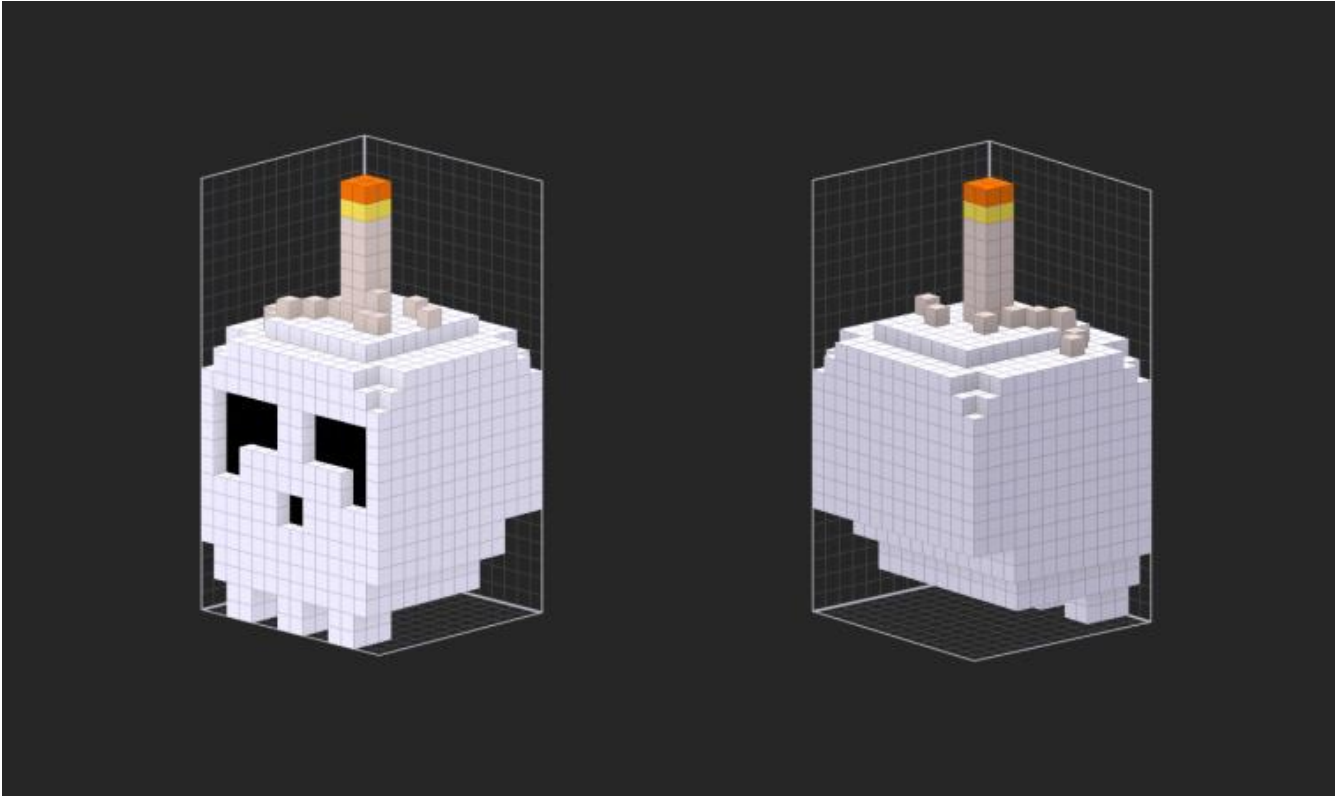
(Modelo 3d Soldier Voxel - TurboSquid 1962790, 2022)



Amaya (2020)



(Redirect Notice, s. f.)



(21957357 - Online Store, s. f.)



(Modelo 3d Cañón de Vóxel - TurboSquid 1951870, 2022)



(Modelo 3d Armas de Vóxeles - TurboSquid 1720786, 2021)



(Gönndyr Games, 2022)



(R/VOXEL, 2018b)



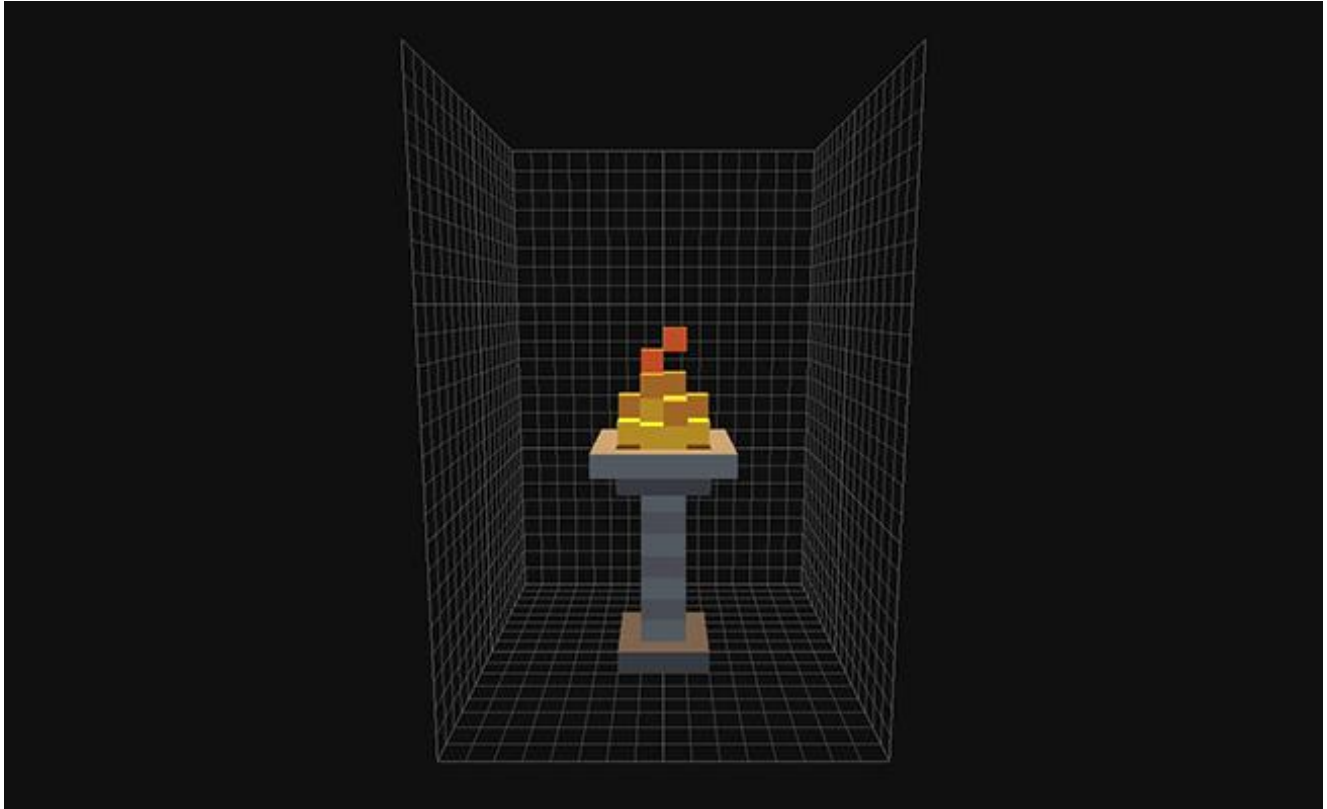
(Voxel Tavern, 2018)



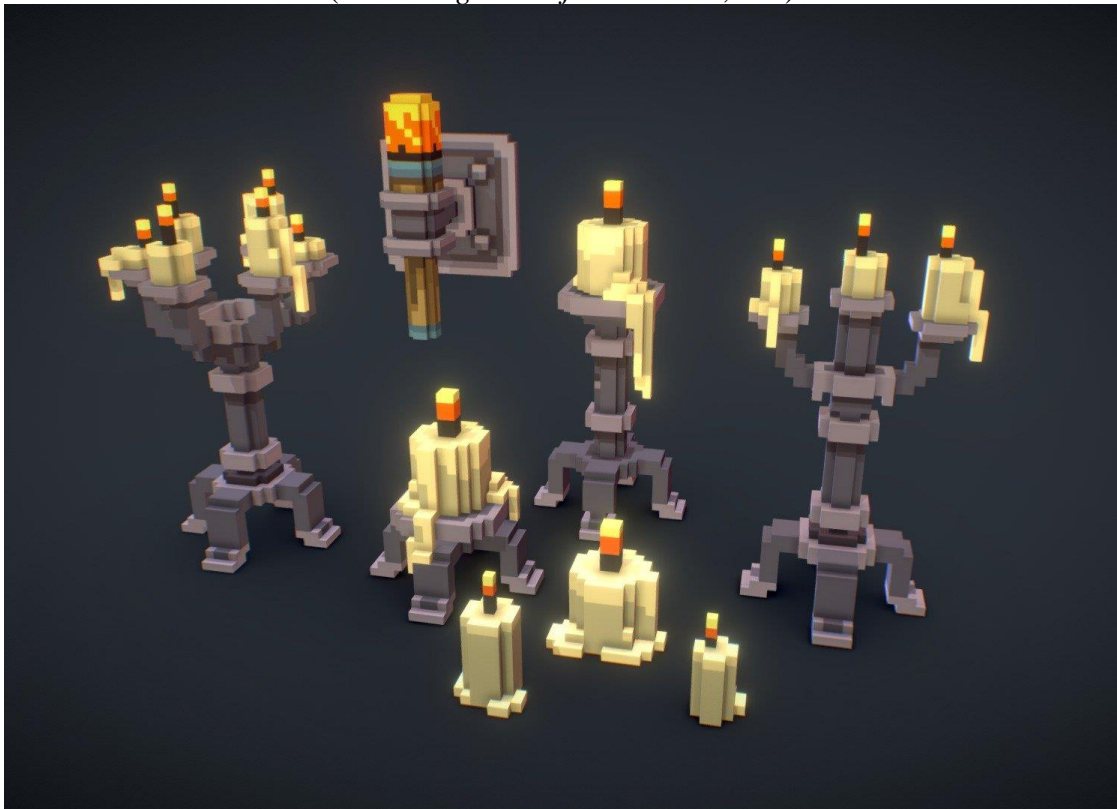
(*Voxel Tavern*, 2018)



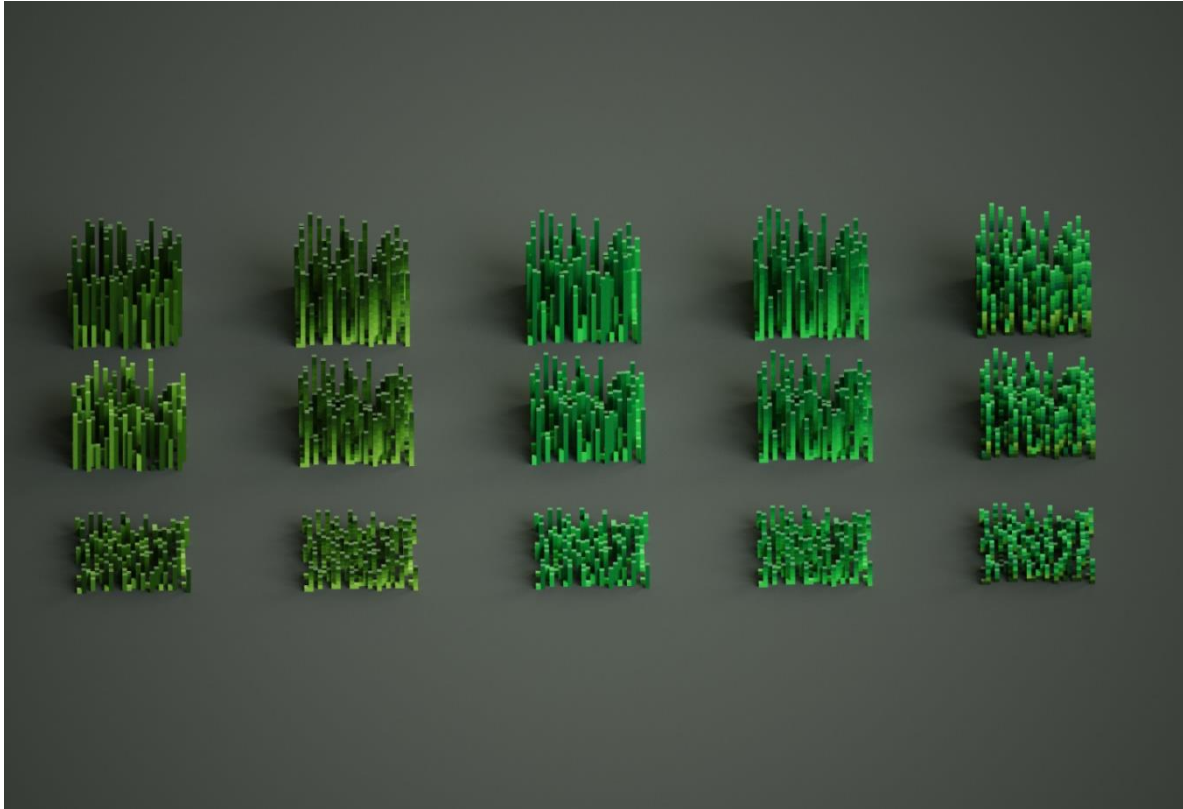
(*Voxel Sword Model*, Michal Mann, s. f.)



(Animating An Object Tutorial, s. f.)



(Sketchfab, s. f.)



(Llc, 2022)



(c) Alicia Yamashita

(35705954 - Online Store, s. f.)

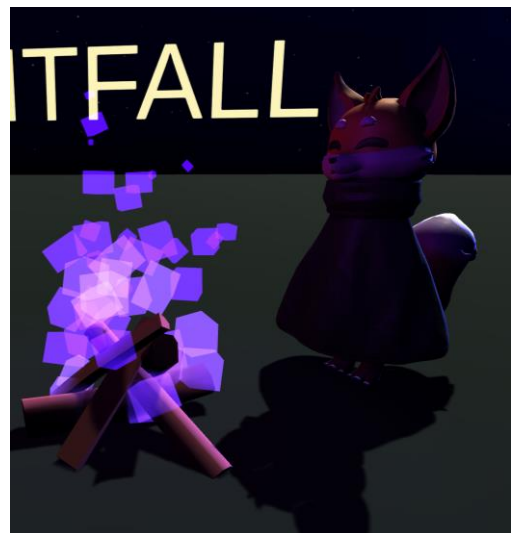
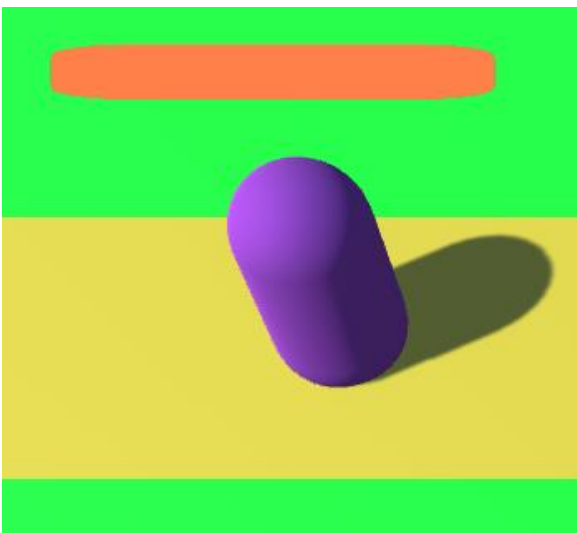
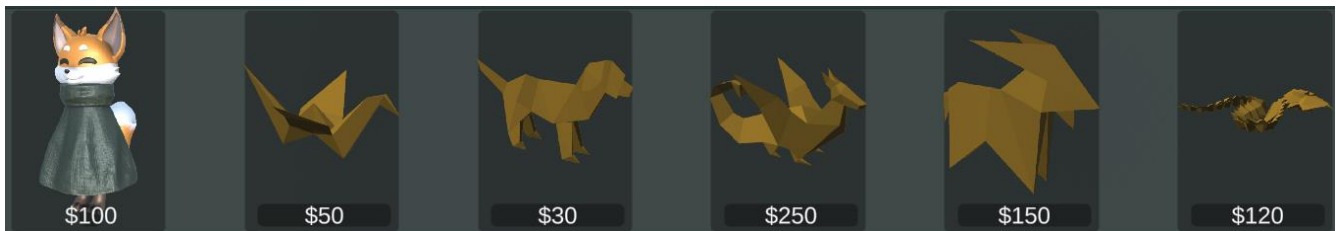


(Voxel Nature Pack / 3D Model, s. f.)



(Project Lowpoly, s. f.)

FinalArt:



Postmortem

Se empezó haciendo el juego pensando sería simple. Sería muy parecido a los proyectos pasados, pero sobre todo se subestimo la generación procedural. Lo que acabo de desplazar y cargar mucho del trabajo.



Se empezó por la generación de la lógica de movimiento de los enemigos empleando una lista de puntos que debe seguir el enemigo. Esto fue simple al emplear como base el proyecto de Barckey's en YouTube. Sin embargo, no era algo con lo que se podría contar para todo.

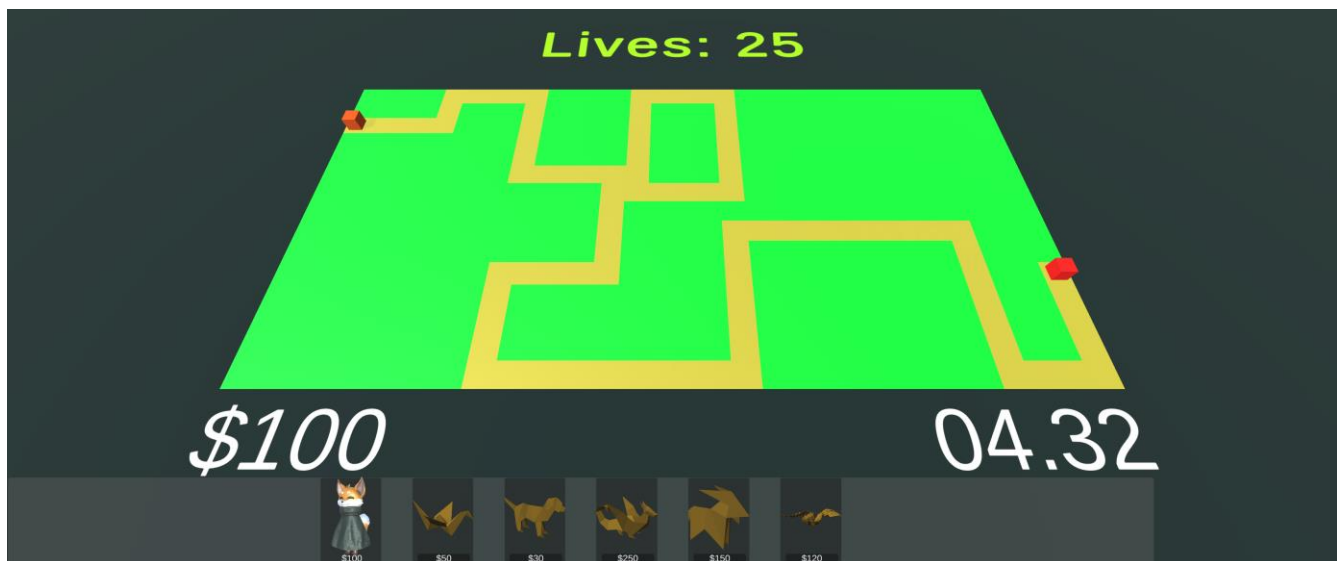
Fue en este punto donde surgieron los problemas. Empezamos con la generación del mapa, lo cual pensamos sería fácil. Solo decir, no salgas, gira de vez en vez y haz que el enemigo siga tu camino. Fácil, ¿no? Pues no fue así de fácil. Pues pese a lo procedural ser algo muy popular y main stream, no hay tanto papel ni lectura de inicio en dichos temas. Por lo que tras intentar implementar múltiples escritos que encontré, terminé por buscar y modificar el código de un sistema 2D, para generar mi generación.



Esta pequeña piedra, me tomo alrededor de 2 o un poco más de semanas, entre prueba y error.

Pero con un sistema de enemigos relativamente fuerte pude continuar con la estructura de las torretas.

Pero en este momento surgió un problema que haría mas lento todo el proceso, al tener en mente el desempeño, resulto en tener que hacer cimientos muy robustos y mucho más complejos, para así poder integrar las técnicas de optimización empleadas hasta ahora. Por lo que se tenía que generar un sistema de estadísticas generales para los enemigos y de las torretas, para poder hacer un pull de objetos y en especial de las balas.



Si bien para proyectos así de pequeños y sin mucho tiempo, se recurrió a las soluciones más rápidas y simples. I no siempre las mas optimas. Generando un poco de enredo entre códigos que se referencian y pasan y reciben valores entre ellos. Pero algo positivo de buscar generar sistemas tan robustos es que la implementación de elementos bajo las mismas líneas o del mismo tipo es mucho más rápido y simple.



Algo que se logro implementar al final del desarrollo gracias a los sistemas tan robustos que tiene el juego. Si bien no he logrado generar una formula de mejora que me resultara de mi agrado total, conseguir un resultado suficientemente satisfactorio. A la vez que poder implementar múltiples torretas, con sus diferentes niveles y skin (“Textura”).

El trabajo final ha sido algo que se puede ver como una Alfa, o un planteo de concepto. Si bien creo que seguiré trabajando en este proyecto pues me parece muy interesante y creo aun tener mucho por aportarle. A la vez que me sorprendió mucho algunas cosas que logre hacer, y los resultados que lograba con mis sistemas. Lo que ahora toca, es acabar de plantear algunas mecánicas y balanceo. Como para dar una revisión a los Requisitos del Previo GDD.

Upgrades:

Al final Se genero un sistema que multiplica los parámetros de forma regular para tener virtualmente mejoras infinitas en elementos infinitos.

```
1 reference
private void UpgradeStats(int level, int damage)
{
    _range = _range * (level / 1.5f);
    _damage = Mathf.RoundToInt(Mathf.Sqrt(level * damage))+1;
    _fireRate = _fireRate + (_fireRate * .20f);
    _value = Mathf.RoundToInt(_value + _upgradeCost);
    _saleCost = Mathf.RoundToInt(_value * .60f);
    _upgradeCost = Mathf.RoundToInt(_upgradeCost * level/1.5f);
    GetComponentInChildren<Renderer>().material = _materials[level--];
}
```

Esto genera un incremento de dificultad al progresar a la vez que genera estadísticas de mejora más claras. Y “automatizadas”.



Encuestas:



Preguntas Respuestas **6** Configuración

Se aceptan respuestas ☒

Resumen

Pregunta

Individual

¿Hubo algún elemento de que te confundiera o no entendieras para que servía?

6 respuestas

No, pero seria bueno poner las diferencias o una dwscripcion en cada una de las torretas

Las instrucciones estan un poco confusas cuando el juego es sencillo

No

no entendi bien, pero poco a poco fui deduciendolo

Pienso que es claro

ME costo entender un poco lo de las teclas ero a final bien

¿Las animaciones ayudaron a la estética del juego, o te molestaron?

6 respuestas

Si ayudaron

Ayudaron

Depende

si ayudaron

Me gustaron

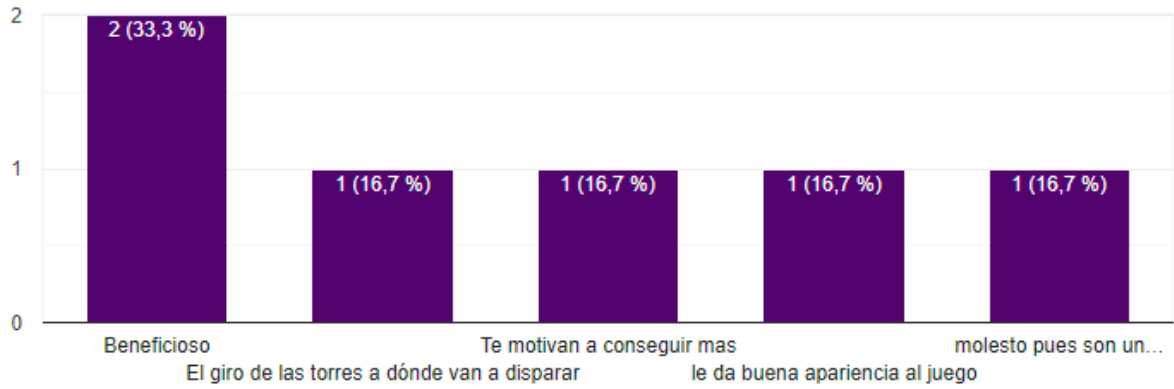
Si ayudan



[Copiar](#)

¿Qué te pareció molesto o beneficioso del uso de las animaciones?

6 respuestas



[Copiar](#)

¿Te pareció intuitivo la construcción de torres?

6 respuestas





¿Notaste algún problema o error al probar el juego? ¿Y cuál fue?

6 respuestas

Una vez no inicio el contador y no salían los malos

A veces al jugar de nuevo no reinicia

El sonido se traba cuando el juego está muy avanzado y tienes muchas torres

si, cuando intentas jugar de nuevo no avanzo el cronometro y en una ocasional reiniciar me aparecieron demasiados enemigos

Algunas veces no funcionó el reloj

retry a veces no funciona

¿Tuviste algun problema al jugar como espasmos u otros problemas de rendimiento?

6 respuestas

No

no

Me falta experiencia para los juegos

no, todo bien

¿Sentiste que la dificultad era correcta? ¿Por qué?

6 respuestas

Si.. esta bien. Aunque la velocidad de los malos podría ser un poco mas rapida.

Si. Creo que esta bien para sarle emoción

Me parece que la dificultad puede llegar a estar algo baja por el daño que hacen las torres a los enemigos

no me parece suficiente el \$ con el se iniciaicia

Si, pro que hay que estar atenta y hacer rápido los movimientos

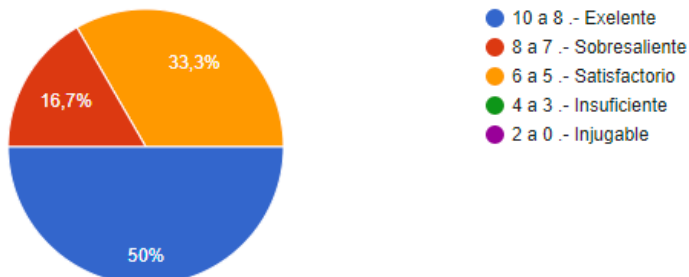
el 4 no me alcanzaba para mucho



 Copiar

¿Si tuvieras que dar una calificación al juego, cual sería?

6 respuestas



¿Cuáles crees son los puntos fuertes de este juego?

6 respuestas

Esta muy entretenido

Las animaciones y lo entretenido del juego

Lo fácil que es de entender el funcionamiento de el juego

entretenido

Son originales las animaciones

la idea y esta entretenido

¿Cuáles crees son los puntos más débiles de este juego?

6 respuestas

La velocidad de los malos

El que no tenga algun control para acelerar los fantasmas

Al avanzar en el juego y tener bastantes torres el juego se puede volver un poco aburrido

los errores antes comentados y el tamaño de algunas animaciones

Se puede mejorar

al principio cuesta entenderle



Análisis:

Si bien en general son muy positivas, creo se debe a que son personas próximas a mí y que no hacer sentir mal. A la vez que a lo largo de la carrera han visto el valor que tiene cada pequeño detalle en los juegos. Sin embargo, creo conseguir más información de ver como jugaron, por lo que para este análisis desbalijare un poco lo visto en las encuestas.

Upgrades:

En el caso de mi papá que nunca ha jugado el género de tower defense. Y algo de lo que mas note, es que no tenían una motivación hacia mejorar sus torretas. preferían poner otras como el Perro, al ser esta barata, a que mejor las que ya tenían. Por lo que habría que buscar una forma para impulsar al jugador a ver las mejoras y mostrarle lo que esta pagando previo a el pago; Esto con el fin de que puedan pensar y equilibrar intereses, costo sobre beneficio.

Para lograrlo podríamos empelar un menú lateral donde mostremos todos los parámetros de la torreta con sliders. A la ves que mostrar lo que se obtendría por mejorarla, mediante números en verde u otras técnicas para llamar la atención del usuario. Este menú debería abrirse solo colocar la torreta, para mostrar al usuario que es una opción a la que tiene acceso y facilitar el entendimiento del programa.



Time:

De forma general, se tiende a volver aburrido si ya se a colocado la torreta deseada y los enemigos son los que van lento. Por lo que el jugador solo observa, por lo que un dar la opción de un multiplicador de velocidad de tiempo sería una fácil y divertida solución.

Al permitir al jugar escoger el rimo de su partida, este se mantiene mas entretenido a la ves que se integra una variable mas con la que mantener al pendiente al usuario.

UI:

De forma general, se les complico entender El funcionamiento y significado de la interfaz. Pues no comprendían las diferencias entre las torretas, y se les debía explicar como mejorar las torretas y ver su rango. Por lo que tendría que ser un elemento mucho mas completo y explicito. Integrando simbología que indique es la tienda, a la ves que al seleccionar algún elemento de esta mostrar sus propiedades para así darle a conocer al usuario por que va a pagar dicho precio.

Otro elemento que note era confuso, era la colocación de las torretas. Pues muchos esperaban tener un drag and drop, o siquiera poder previsualizar el rango de sus torretas. Algo que veo como lógico e incluso esencial para una mejor experiencia y comprensión. Por lo que un paso lógico e importante, sería añadir un sistema de previsualización de rango o un drag and drop con el mismo sistema.



Enemigos:

Estos llegan a ser confusos para algunos usuarios. Pues sienten que tiene muy poca involucración en el desarrollo. Por lo que mas de una vez me preguntaron el cómo atacaban ellos a los enemigos, o que más debería hacer.

Esto se puede solventar con mas efectos que premien de mayor forma al usuario por sus buenas acciones generando algo de más integración en el juego y su desarrollo.

Dificultad:

En múltiples momentos note una curva de aprendizaje muy empinada. Pues o perdían muy rápido sin poder hacer nada, o lograban detener todo el ataque enemigo sin ninguna dificultad y un excedente de dinero Sobrante.

Como nivel introductorio o conceptual, me parece mas que completa. Sin embargo, para un público mas final y nuevo como lo es mi familia, se requieren sistemas mas pulidos a la vez que siento un mayor diseño del software.

Conclusiones:

Falta un mayor desarrollo del diseño de interfaz y un mejor manejo de los tiempos tanto de pruebas parciales como para poder dar mas maduración a los sistemas buscando mejores resultados.

En este caso se puede mejorar sobre todo en el aspecto del diseño e interfaz. Los demás sistemas son bastante sólidos, pero lo que es UI y su entendimiento es deficiente en el mejor de los casos.



Apa

At Night Fall:

21957357 - online store. (s. f.). https://limaloveet.pics/product_details/21957357.html

35705954 - online store. (s. f.). https://sjgoodsm.pics/product_details/35705954.html

55 Voxel Swords | 3D Weapons | Unity Asset Store. (s. f.). Unity Asset Store.
<https://marketplace.unity.com/packages/3d/props/weapons/55-voxel-swords-184172>

Amaya, G. V. (2020, 15 mayo). *Voxel Skeleton Character, Alex Garton*. Pinterest.
<https://www.pinterest.com.mx/pin/806074033294976611/>

Android Authority. (2015, 13 abril). *14 best Android tower defense games* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=uvCKyNbpFVo>

Animating an Object tutorial. (s. f.). IndieDB. <https://www.indiedb.com/tutorials/animating-an-object>

Gönndyr Games. (2022, 20 abril). *Canon Tower remake for «Towers “n” Dragons»*. X.
https://twitter.com/goenndyr_games/status/1516761782672580615

Llc, D. (2022, 3 mayo). Voxel Garden pack Gratis Modelo 3D in Otros 3DExport. 3DExport.
<https://es.3dexport.com/free-3dmodel-voxel-garden-pack-396605.htm>

modelo 3d Armas de vóxeles - TurboSquid 1720786. (2021, 17 abril). TurboSquid.
<https://www.turbosquid.com/es/3d-models/voxel-guns-model-1720786>

modelo 3d Cañón de vóxel - TurboSquid 1951870. (2022, 30 agosto). TurboSquid.
<https://www.turbosquid.com/es/3d-models/3d-voxel-cannon-1951870>

modelo 3d Soldier Voxel - TurboSquid 1962790. (2022, 22 septiembre). TurboSquid.
<https://www.turbosquid.com/es/3d-models/3d-soldier-voxel-model-1962790>



OpenSea. (s. f.). Black Raven - Voxels Wearables. OpenSea.

<https://opensea.io/es/assets/ethereum/0xa58b5224e2fd94020cb2837231b2b0e4247301a6/2138>

Project Lowpoly. (s. f.). itch.io. https://itch.io/queue/c/1815922/project-lowpoly?game_id=645897

R/VOXEL. (2018b). *Tower*. Redit. Recuperado 22 de abril de 2024, de <https://www.reddit.com/r/VOXEL/comments/8oeiep/tower/>

Sketchfab. (s. f.). *Dungeon Candles - Voxel Series - 3D model by BitGem*. <https://sketchfab.com/3d-models/dungeon-candles-voxel-series-78dc3267ab493faae61f6f3716faed>

Sketchfab. (s. f.-b). Voxel Style Fantasy Bow - Download Free 3D model by Ascii_Ape_III. <https://sketchfab.com/3d-models/voxel-style-fantasy-bow-4355787fc49746aeab1c67c74aeabae0>

Voxel Nature pack | 3D model. (s. f.). CGTrader. <https://www.cgtrader.com/3d-models/plant/other/voxel-nature-pack>

Voxel Spider. (s. f.). RenderU.com. <https://renderu.com/en/gallery/artwork/170370>

Voxel Sword model, Michal Mann. (s. f.). ArtStation. <https://www.artstation.com/artwork/xJ4E82>

Voxel Tavern. (2018). Redit. Recuperado 22 de abril de 2024, de https://www.reddit.com/r/isometric/comments/8ocgdo/voxel_tavern/?rdt=62665

Assets:

10 SkyBoxes Pack : Day - Night | 2D Sky | Unity Asset Store. (2016, 1 septiembre). Unity Asset Store. <https://assetstore.unity.com/packages/2d/textures-materials/sky/10-skyboxes-pack-day-night-32236>

10 SkyBoxes Pack : Day - Night | 2D Sky | Unity Asset Store. (2016b, septiembre 1). Unity Asset Store. <https://assetstore.unity.com/packages/2d/textures-materials/sky/10-skyboxes-pack-day-night-32236>

Distance Those icons Fill icon. (2017, 10 julio). Freepik. https://www.freepik.com/icon/distance_484140

Low health bar. (s. f.). Pixilart. <https://es.pixilart.com/art/low-health-bar-698d6a25ab4e333>



Origami Animals Pack / 3D Animals / Unity Asset Store. (2021, 15 febrero). Unity Asset Store.

<https://assetstore.unity.com/packages/3d/characters/animals/origami-animals-pack-123159>

Sketchfab. (s. f.-b). *Fox in a cape - Download Free 3D model by Bonfire SkylarFox (@bonfire.png).*

<https://sketchfab.com/3d-models/fox-in-a-cape-1b061329af8746b79267f56852b17204>

Audio:

Freesound - forrest and river.wav by barkenov. (s. f.).

<https://freesound.org/people/barkenov/sounds/406816/>

Freesound - Love Arrow.wav by ATG65. (s. f.). <https://freesound.org/people/ATG65/sounds/621470/>

Investigacion:

Arturo Nieto de Almeida, Alejandro García del Valle, & Diego Crespo Pereira. (s. f.). Elocons: un algoritmo de construcción de rutas eficiente para la pequeña y mediana empresa de distribución.

Recuperado 23 de abril de 2024, de https://www.gii.udc.es/img/gii/files/ELOCONS_Dyna_Enviada.pdf

Brownlee, J. (2023, 10 octubre). A Tour of Machine Learning Algorithms.

MachineLearningMastery.com. <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>

colaboradores de Wikipedia. (2023a, mayo 12). *Algoritmo de búsqueda A**. Wikipedia, la Enciclopedia Libre.

[https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*](https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A%2A)

colaboradores de Wikipedia. (2024b, enero 10). Generación por procedimientos. Wikipedia, la Enciclopedia Libre.

https://es.wikipedia.org/wiki/Generaci%C3%B3n_por_procedimientos

colaboradores de Wikipedia. (2024f, abril 8). Búsqueda de ruta. Wikipedia, la Enciclopedia Libre.

https://es.wikipedia.org/wiki/B%C3%BAsqueda_de_ruta

colaboradores de Wikipedia. (2024i, abril 17). Algoritmo. Wikipedia, la Enciclopedia Libre.

<https://es.wikipedia.org/wiki/Algoritmo>



David. (2016, 9 junio). Como se definen paths en un juego. Aprende Game Maker.

<https://www.aprendegamemaker.com/paths/>

De la Empresa, U. (2021, 6 julio). ¿Qué entendemos por algoritmo? UDE Universidad de la Empresa.

<https://ude.edu.uy/que-son-algoritmos/>

Ixel Games. (2021, 6 enero). Tutorial Construct 3, crear rutas (path) de forma fácil. [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=myH5XxGMrIs>

Jiménez-Carrión, M., Jiménez-Panta, A. B., & Coaquira-Velásquez, M. A. (2023). Algoritmo evolutivo generador de rutas eficientes para el transporte público. *Información Tecnológica*, 34(1), 71-88.

<https://doi.org/10.4067/s0718-07642023000100071>

Ramirez, A. (2021, 13 diciembre). Máquinas de estado. DEV Community.

<https://dev.to/gara501/maquinas-de-estado-4ma5>

Ruiz-Azuaga, V. B. (2016, 14 marzo). La ciencia de los videojuegos: IA (III). ULUM. <https://ulum.es/la-ciencia-de-los-videojuegos-ia-iii/>

Santiago Fiorino. (2024, 9 febrero). *Comparando Algoritmos: A* vs Dijkstra, en el mapa de la ciudad*.

[Vídeo]. YouTube. <https://www.youtube.com/watch?v=oMgfGkFSgI0>

Santiago Fiorino. (2024b, febrero 9). *Comparando Algoritmos: A* vs Dijkstra, en el mapa de la ciudad*.

[Vídeo]. YouTube. <https://www.youtube.com/watch?v=oMgfGkFSgI0>

RecursosCodigo:

Brackeys. (2016, 6 julio). *How to make a Tower Defense Game (E01) - Unity Tutorial* [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=beuoNuK2tbk>

Episode 9: Procedural Path Generation. (s. f.). Gist.

<https://gist.github.com/goldennoodles/8eb753b4c221dd6ad03ef5ef55edbf07>



GarnetKane. (s. f.). *IsometricPathGenerator/Assets/Scripts/MapGenerator.cs* at main ·

GarnetKane99/IsometricPathGenerator.

GitHub.

<https://github.com/GarnetKane99/IsometricPathGenerator/blob/main/Assets/Scripts/MapGenerator.cs>