

UNIVERSIDAD TECNOLÓGICA DE SANTIAGO, UTESA

SISTEMA CORPORATIVO

Facultad Arquitectura e Ingeniería

Carrera de Ingeniería en Sistemas Computacionales



PROGRAMACION DE VIDEOJUEGOS

PROYECTO FINAL: CAPITULO I

Profesor:

Iván Mendoza

Integrantes:

Guillermo Santos 2-18-0494

31 de marzo, 2022

Santiago de los Caballeros,

Rep. Dominicana

INDICE

INDICE	2
CAPITULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO	4
1.1 Descripción	4
1.2 Motivación	4
1.2.1 Originalidad de la Idea	4
1.2.2 Estado del Arte.....	4
1.3 Objetivo general	6
1.4 Objetivos específicos	6
1.5 Escenario.....	6
1.5.1 Menú principal	7
1.5.2 Selector de niveles	7
1.5.3 Plantilla de escenario de niveles.....	8
1.6 Contenidos	8
1.6.1 Ambiente	8
1.6.2 Torres	8
1.6.2.1 Torres defensivas.....	8
1.6.2.2 Torres estratégicas.	9
1.6.3 Enemigos.....	9
1.6.3.1 Rail	9
1.6.3.2 Coloso	9
1.6.3.3 Flash	9
1.6.3.4 Trickster	9
1.6.3.5 Variante jefe	10
1.7 Metodología.....	10
1.7.1 Incremento I: Plantillas para el desarrollo.....	10
1.7.2 Incremento II: Torretas, enemigos, componentes interactivos y mejora de las mecánicas bases	10
1.7.3 Incremento III: Diseño de niveles y mecánica “Upgrades”	10
1.8 Arquitectura de la aplicación	10
1.9 Herramientas de desarrollo	10
REFERENCIAS BIBLIOGRAFICAS.....	12
LINK DEL PROYECTO:.....	12
LINK DE CAPITULOS:.....	12

CAPITULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO

1.1 Descripción

Este juego se llama “Defend The Core”, se trata de un tower defense de estilo sci-fi con componentes modernos, donde el jugador se verá envuelto en la necesidad de construir torres y otros tipos de estructuras para defender el “CORE”, núcleo de la central minera de la empresa SANCTUSCORE, de los ataques de monstruos provenientes de CORES no purificados en el planeta.

1.2 Motivación

El subgénero de estrategia llamado tower defense siempre ha sido uno de mis géneros favoritos de los videojuegos, y con la observación del juego ROGUE TOWER, un juego tower defense que salió hace unos meses, y sus mecánicas pensaba en cómo sería un tower defense que incorpore ciertas de sus mecánicas, pero que a la vez las reestructure para una mayor dificultad.

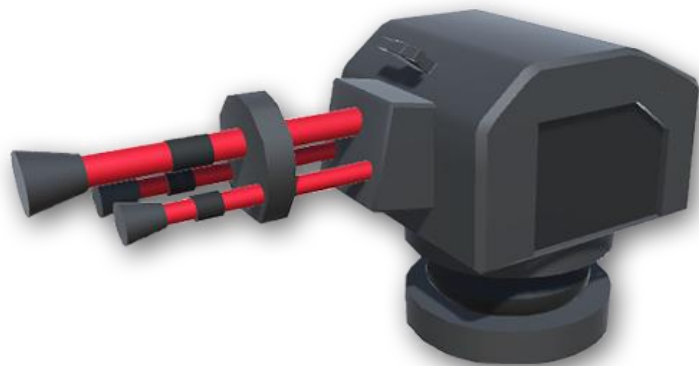
Por lo que cuando me vi en la necesidad de pensar en que desarrollar como proyecto final dije “¿Porque no?” y desarrollé esta idea que tenía aún más.

1.2.1 Originalidad de la Idea

Primeramente, remplazamos la mecánica de uso de dinero por una de recolección de recursos en el mapa en tiempo real, y el “Mana” por “Energía” para que este más acorde con la temática Sci-fi del juego y no con la temática de fantasía del juego que use como inspiración. También se cambió por completo la transición entre oleadas, de una sesión de estrategia donde podías esperar hasta que le des clic al botón de listo para iniciar la oleada, a una cuenta regresiva que te obliga a pensar constantemente fuera y durante las oleadas como reestructurar las defensas para el CORE y las otras estructuras del mapa. Para esto surgió otra idea original, y es el agregado de un tipo de monstruos cuyo comportamiento es diferente al resto, en lugar de seguir la ruta para llegar al CORE y destruirlo, este se sale del camino y se dirige a las estructuras de producción de recursos más cercanas para destruirlas, lo cual te obliga a crear defensas descentralizadas y a no construir sin un plan en mente.

1.2.2 Estado del Arte

Debido a mi inexperiencia con herramientas de diseño gráfico y el modelado 3D, se utilizarán 3 paquetes de assets distintos para el desarrollo del juego, uno usado en la serie de video tutoriales usado para crear la plantilla base del juego mostrada en tareas anteriores, y las otras 2 provenientes de la “Assets Store de Unity”, aquí mostro algunas capturas:





1.3 Objetivo general

Crear un tower defense de alta dificultad con mecánicas únicas y otras utilizadas en el juego que sirve como inspiración.

1.4 Objetivos específicos

- Crear un juego multiplataforma.
- Crear un tower defense entretenido.
- Crear mecánicas desafiantes que obliguen a pensar al jugador.
- Diseñar un juego de estilo Sci-Fi futurista.

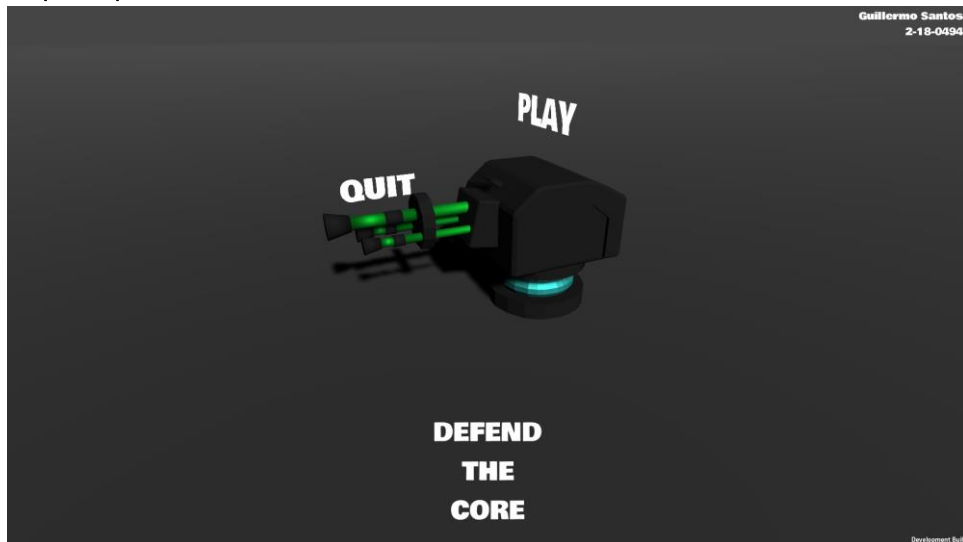
1.5 Escenario

El juego constara de 4 tipos de escenas distintas, la escena del menú principal donde se muestran las opciones de “QUIT” para salir del juego “JUGAR” para entrar al menú de nivel juego, y “UPGRADES” para realizar ciertas mejoras al CORE y desbloquear estructuras (Actualmente en desarrollo).

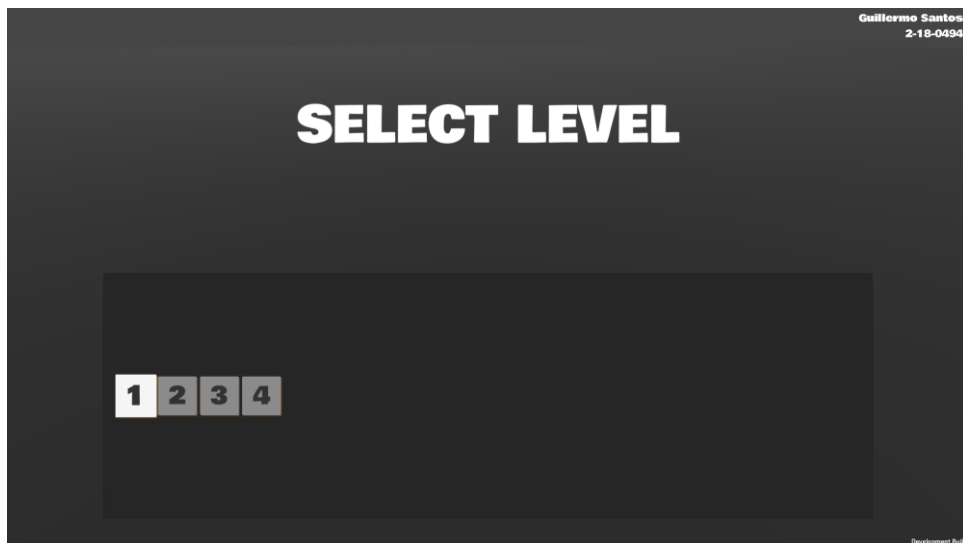
El escenario de niveles será un cuadrado con varias alturas y recursos dispersos en el mapa, siendo esto distinto para cada nivel, aunque esta parte aún sigue en desarrollo por lo que no se tiene visual de esta todavía.

Solo se tiene visual de 2 de estas escenas actualmente + una visual de la base utilizada en el escenario de niveles.

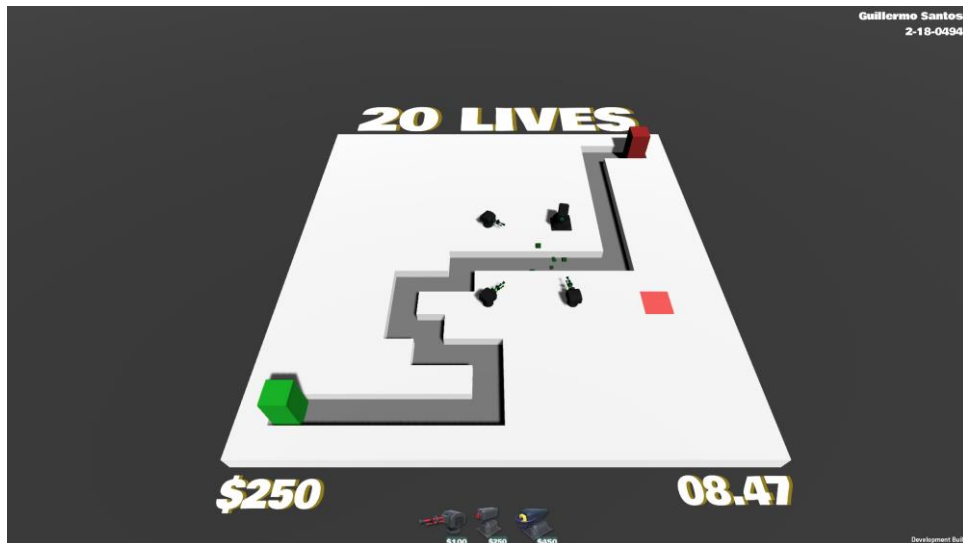
1.5.1 Menú principal



1.5.2 Selector de niveles



1.5.3 Plantilla de escenario de niveles



1.6 Contenidos

El contenido del juego se divide en los distintos componentes que dividen el nivel:

1.6.1 Ambiente

El ambiente está compuesto por 4 elementos principales, el fondo del nivel, que por lo general es un gris cercano a negro, el camino a seguir de los enemigos, el CORE, el EnemySpawner de donde se originan los enemigos y finalmente los nodos.

Existen 2 tipos de nodos:

- **Los construibles:** que son aquellos donde podrás colocar las torres, estos tienen una mecánica única que incrementa el rango de las torres según su altura.
- **Los no construibles:** son los nodos donde no podrás colocar torres, sirven de obstáculos en ciertos niveles para reducir el área en que tus torres son capaces de disparar y son el lugar donde se posicionan los recursos que deberás recolectar en el mapa.

1.6.2 Torres

Algo que nunca falta en los tower defense, las torres, estas están divididas en 2 grupos, las defensivas y las estratégicas.

1.6.2.1 Torres defensivas.

Son las torres principales a construir para defender el CORE, están compuestas por las torres que atacan a los enemigos o tienen algún tipo de efecto sobre ellas. Estas son:

- **StandardTurret:** La torre básica para la defensa de la base.
- **MissileLauncher:** Una torre de defensa con daño de área.
- **LaserBeamer:** Una torre que hace daño constante, además de ralentizar los enemigos.
- **ElectricTower:** Una torre que disminuye la velocidad de enemigos dentro de su área.

1.6.2.2 Torres estratégicas.

Son torres incapaces de atacar enemigos pero que cumplen una función específica, se necesita al menos una de estas para superar un nivel del juego y estas son:

- **HealerTurret:** Una torreta capaz de curar al CORE y otras torres estratégicas dentro de su rango.
- **EnergyTurret:** Una torreta que recolecta energía de los almacenes energéticos en el mapa, solo si uno de estos está en su rango.
- **MiningTurret:** Una torre que se encarga de la recolecta de recursos en el mapa, sin estas los recursos de construcción se acabarán rápidamente.

1.6.3 Enemigos

Los enemigos son una lista variada de los oponentes que aparecerán en las oleadas, Estos constan de 4 tipos y una variante llamada "BOSS" de cada uno de estos.

1.6.3.1 Rail

Es el enemigo base, y cuenta con las estadísticas estándar para los enemigos, se verá presente en todos los niveles.

1.6.3.2 Coloso

Una variante más lenta, pero con mayor vida y tamaño que la variante "Rail", su dureza hace que se requieran varias torres para abatirlos y si se juntan varios pueden ser un dolor de cabeza para el jugador.

1.6.3.3 Flash

Una variante de "Rail" con menor vida, pero con una velocidad muy superior, dependiendo de la estructura de la defensa, estas pueden convertirse en algo mortal para el jugador.

1.6.3.4 Trickster

Este se puede considerar como el más peligroso entre los ya mencionados, a diferencia que el resto, comparte las funciones de ataque de las torres (En lugar de morir luego de chocar con el CORE, este le ataca a la distancia), pero lo que resulta verdadera mente problemático es que en lugar de seguir el camino junto con las demás torres, esta ataca a las torres estratégicas hasta que no queden ninguna, solo así se dirigirá al CORE, tiene ligeramente más vida y mayor velocidad que la variante "Rail" lo que sumado a su particular comportamiento lo vuelve un gran dolor de cabeza y un factor del cual hay que cuidarse en varios niveles, pues pueden arruinar completamente la partida.

1.6.3.5 Variante jefe

Cada uno de los enemigos antes listados tienen una variante jefe, con estadísticas muy superiores, son el enemigo final en varios niveles. El hecho de que no se limitan a solo uno por nivel más, la posibilidad de la variable Trickster, lo vuelven un enemigo a temer.

1.7 Metodología

La metodología seguida para el desarrollo del juego ha sido la incremental, donde cada incremento está compuesto por una serie de funcionalidades del juego y sus componentes, en total serían 3 incrementos:

1.7.1 Incremento I: Plantillas para el desarrollo

Este es el primer y más pequeño incremento, consta de la creación de las bases para el diseño de los niveles y la composición del juego base o plantilla ya presentados en tareas anteriores.

1.7.2 Incremento II: Torretas, enemigos, componentes interactivos y mejora de las mecánicas bases

La creación de las torretas de defensa, las torretas estratégicas, los distintos enemigos y los recursos del mapa, así como la interacción entre todos ellos, además de cambios en las mecánicas bases del juego, como el sistema de oleadas, sistema de objetivos y pathfinding de enemigos, y condiciones de victoria o derrota de los niveles.

1.7.3 Incremento III: Diseño de niveles y mecánica "Upgrades"

Con los componentes de los incrementos anteriores se diseña cada nivel que tendrá el juego, además se plantea la creación del sistema de mejoras que se utilizará para desbloquear las torretas y realizar mejoras, como la cantidad de recursos iniciales, vida del CORE y reducciones de costos.

1.8 Arquitectura de la aplicación

La arquitectura de desarrollo de la aplicación será la arquitectura en capas, donde el juego estará dividido en distintas capas que representan un componente en específico, como los materiales usados para dar color a los modelos, audio y audiomixer para controlar los volúmenes de los audios, Prefabs para los Prefabs que componen los objetos del juego, Scripts para la lógica y controladores del juego y ScriptableObjects que representan la base para el comportamiento de los objetos del juego.

1.9 Herramientas de desarrollo

Se utilizará una serie de herramientas para el desarrollo del juego, como son:

- Motor gráfico: Unity

- IDE: Visual Studio 2022
- Fuente de audios: <https://elements.envato.com>
- Unity Assets Store

REFERENCIAS BIBLIOGRAFICAS

- Juego inspirado en: [Rogue Tower\(Steam\)](#)
- Video tutoriales, juego plantilla y assets para 3 de las torres de defensa: [Serie](#)
- Assets Store, componente 1, para torres estratégicas: [Robo's turret \(free sample\) | 3D Sci-Fi | Unity Asset Store](#)
- Assets Store, componente 2, para la torre eléctrica: [Electric Turret | 3D Characters | Unity Asset Store](#)
- Fuente de audio: [Envato Elements: vídeos, música, fotos y gráficos de archivo ilimitados](#)

LINK DEL PROYECTO: [Guillermo-Santos/DefendTheCore \(github.com\)](#)

LINK DE CAPITULOS: [DefendTheCore/Capitulos del proyecto at santo · Guillermo-Santos/DefendTheCore \(github.com\)](#)