

Modelamiento del Proyecto

Angel Graciano Graciano Espitia
Anderson Giraldo Arboleda

Informatica 2
Profesores: Augusto Enrique Salazar - Jonathan Ferney
Gomez
6 de abril de 2022

Índice

1. Introducción	3
2. Historia del juego	3
3. Descripción	3
3.1. Jugador	3
3.2. Escenario	3
3.3. Enemigos	4
3.4. Jefes	4
3.5. Proyectiles	5
4. Modelamiento Físico	5
5. Programación	5
5.1. clases	5

1. Introducción

Es importante la adquisición de conocimientos para la vida, y ello se basa en aplicarlos, para esto es este proyecto, para que todo lo que aprendimos en el curso podamos utilizarlo como una herramienta para la realización del proyecto y como no, para la vida profesional. En este escrito se relatara el planteamiento de nuestro proyecto.

2. Historia del juego

En un futuro lejano, hay dos imperios que gobiernan la humanidad, logos y zebes. En nuestro caso es debemos defender el imperio de logos con una nave llamada Airgear. Donde nos enfrentaremos a tropas aéreas del imperio de zebes por varios escenarios.

3. Descripción

3.1. Jugador

El como se explica en la historia nosotros seremos una nave del imperio de logos, el jugador podrá interactuar con la nave por medio del teclado utilizando las letras A, W, S y D en el cual nos podremos mover libremente en un escenario cuadrado limitado por barreras invisibles. lo que el jugador podrá hacer es mover y disparar provisionalmente 2 proyectiles para derribar a sus enemigos. el jugador contara con cierta cantidad de vida, que esta pensado para ser una barra verde o también sera números, depende bastante como se de el desarrollo.

3.2. Escenario

Los escenarios que constituirán el juego son 3. Estos serán en el mar con especie de islas montañas. La otra sera en una especie de caverna. Y por ultimo el escenario final sera en el cielo de una ciudad.

Cada escenario va tener sus enemigos y obstáculos, y también su dificultad, en otras palabras cada ves que el jugador avance de escenario la dificultad incrementara.

Estos escenarios se tenían planeado para que los objetos del fondo fueran items gráficos que se fueran desplazando y regresaran a cierta posición como si fuera un ciclo para crea la ilusión que la nave se estaba moviendo. Pero dependerá mucho de que escenario se este para implementarlo.

3.3. Enemigos

Lo tipos de enemigos serán 3 sin contar a los jefes de cada fase, también se tiene planeado hacer un obstáculo pero depende de la progresión del desarrollo del juego. Estos enemigos se destruirán cuando reciban un disparo de la nave del jugador. El primer tipo de enemigo sera un zepelín el cual este no hará mas que avanzar hacia delante y estorbar para el jugador haciéndole daño si este lo golpea.

El segundo sera otra nave como la que controla el jugador pero esta dispara proyectiles rectilíneos o parabólicos, esta también se moverá en movimiento rectilíneo, cabe recalcar que el tipo que de proyectiles de este enemigo varia con el escenario para implementar la dificultad, este también nos hará daño si colisionamos con el.

El tercer tipo de enemigo sera un misil que tendrá un movimiento oscilatorio, el cual si colisiona con el jugador le hará daño, este digamos que es el enemigo que hará mas daño al jugador.

También hemos decidido agregar un obstáculo provisional el cual sera una columna vertical que si golpea al jugador este le hará daño. Pero dependiendo de como vaya el desarrollo del proyecto este se implementara

3.4. Jefes

Este apartado va a ser excepcional en el sentido que no hemos definido las formas de los jefes o como se van ver, ya que lo que sabemos es que son grandes y se nos puede dificultar la búsqueda de algún sprite y dependiendo de que podamos encontrar o hacer serán de una forma o otra. Pero si sabemos en que consistirán.

El primer jefe sera básico porque solo lanzara proyectiles rectilíneos, pero serán los suficientes para poner a prueba el jugador. el jefe tendrá una vida especifica el cual el jugador podrá vencer con un determinado numero de disparos para asi ganar la fase.

El segundo jefe también contara de proyectiles rectilíneos además de un tendrá proyectil que sera un movimiento circular. también este contara con una vida especifica sera un poco mayor al del anterior jefe.

El tercer jefe contara con proyectiles rectilíneos y circular. además también tendrá un proyectil que sera una bola que ira haciendo colisiones elásticas en el escenario. Este jefe también tendrá cierta vida y igual que el anterior tendrá mas salud.

3.5. Projectiles

El jugador y algunos enemigos constaran de proyectiles, no todos lo proyectiles estarán en el jugador o cada una de los enemigos, solo serán algunos y estos principalmente serán 4. Antes que todo, la descripción gráfica es provisional porque todavía no tenemos los sprites a la mano, el cual si se nos dificulta la obtención de algún sprite, este se vera a cambios.

El primero sera un disparo con forma de un rayo láser que ira rectamente, el cual lo tendrá el jugador, el enemigo que también es una nave como el jugador y el primer jefe.

El segundo proyectil sera una bomba que tendrá el jugador(provisional) y el enemigo anteriormente mencionado. este sera un movimiento parabólico.

EL tercer proyectil tomara la forma de una bola de fuego, el cual el segundo jefe tendría. este le daremos un movimiento circular uniforme.

El cuarto proyectil sera una roca que solamente el ultimo jefe tendría y este colisionaría con las paredes del escenario haciendo colisiones elásticas.

4. Modelamiento Físico

Todos los modelos fisicos van a estar albergados en una clase, y prácticamente sera una de las principales ya que esta se va a heredar a casi cualquier cosa que represente un movimiento.

Esta constara de varios modelos fisicos como.

- Movimiento Rectilíneo Uniforme
- Movimiento Parabólico
- Movimiento Oscilatorio
- Colisiones elásticas
- Movimiento Circular Uniforme

5. Programación

5.1. clases

Como se había mencionado antes, se va a contar con una clase llamada física que es la que se va a encargar de la parte. Contara con atributos fisicos: posiciones en X y Y, velocidades en X y Y, diferenciales de X (provisional). también contara con un atributo

que con la ayuda de condicionales en un método que vamos a mencionar después, el cual decidirá que modelo físico utilizar.

En cuanto a métodos Por ahora se pensó en Getter and Setter para los atributos físicos. Y también otro donde vamos al almacenar todos los modelos físicos del juego.

primero se había decidido hacer una clase de enemigos, otra de proyectiles, pero creemos que al final se va a escoger una clase por cada enemigo, en estas heredaremos la clase de física y denominado la clase como objeto gráfico vamos a poder darle una imagen.

Para cada una de las clases que se van a mostrar en el escenario va a ser así, pero no todas las clases van a contener los mismo atributos ya que recordemos que los jefes tendrán diferentes características, el jugador una y los enemigos otras; por ejemplo, la vida.

También se tendrá clase para cada uno de los proyectiles pero aun no sabemos si esa clase se la vamos a heredar a los objetos que lanzan proyectiles. puede que por problemas técnicos no tengamos el resultado que esperamos. por eso cuando se este desarrollando el proyecto se vera con mas certeza.

También planeábamos tener varios métodos en la clase mainwindow que se aplicaran a cada uno de los escenarios o también podría darse el caso de tenerlo en una clase.

Para esta parte de implementación todavía estamos muy verdes y necesitamos contextualizarnos de varios fenómenos para poder determinar con claridad lo que es el código.

Por lo que se ha visto vamos a necesitar de varios timers para hacer la implementación del movimiento autónomo de los enemigos y de los proyectiles y necesitamos planear este tema bien para el tema de sincronidad del vídeo juego.