Propuesta Proyecto Final

videojuego informática II

Diego Zuluaga
Julian Salazar

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Diciembre de 2020

Índice

1. Motivación.

- Poner a prueba las habilidades adquiridas a lo largo del curso programando con el lenguaje C++ y el paradigma de la programación orientada a objetos en el entorno de desarrollo integrado Qt Creator.

-La posibilidad de crear un videojuego por sí mismo; ver que uno tiene la capacidad y el control de crear objetos reales y simular sistemas físicos.

-Las destrezas adquiridas durante el curso, ser capas de posteriormente aplicarlas a proyectos venideros, que contribuyan en el crecimiento profesional y personales de los implicados en el videojuego.

-Desarrollar un juego entretenido, que se caracterice por su diseño y jugabilidad.

2. Objetivos.

-Plasmar en el videojuego lo aprendido en informática II durante el semestre.

-Cumplir los requisitos que se piden del proyecto final.

-Ser capaces de implementar sistemas físicos.

-Usar repositorios para para facilitar el trabajo en grupo.

-Hacer un buen manejo de la información al realizar documentación del proyecto.

-Lograr un videojuego que sobresalga por su jugabilidad, diseño y originalidad.

3. Metodología.

Fase	Tema	Descripción
1	Análisis	Dados los requisitos para la realización del proyecto final, proponer ideas de juegos con el fin de revisar si es viable para realizar en el tiempo estimando y que cumpla los objetivos (requisitos) propuestos.
2	Propuesta	Una vez se tenga la idea del juego, se pasará a dar una descripción del videojuego (índice 5) y hacer una propuesta de las clases (Índice 6) a usar, con sus métodos y atributos (que pueden variar según las necesidades y dificultades que se presenten).
3	Ejecución	Implementar en qt Creator las clases escogidas durante la propuesta, y desarrollar el videojuego siguiendo el cronograma.
4	Documentación	Documentar el código a medida que se progrese en el videojuego.
5	Seguimiento	A medida de ir avanzando en la ejecución del proyecto, se deben realizar pruebas, para corregir errores que se presenten y verificar que se esté cumpliendo lo estipulado en el cronograma, en caso de retrasos adaptar el cronograma para finalizar el proyecto en el tiempo de manera exitosa.
6	Entrega	Generar ejecutable y tráiler del proyecto final, además el vídeo de explicación de la implementación del videojuego.

4. Cronograma.

Tarea	Descripción	Fecha
1	Crear repositorio.	18 de diciembre
2	-Crear la clase Character (Personaje principal) y Bullet. -Programar instancia de la clase Character para que con ciertas teclas se pueda mover hacia arriba, hacia abajo y a los dados. - Programar el poder generar disparos por parte del personaje principal al presionar una tecla.	21, 22, 23 de diciembre
3	-Adelantar en el diseño de los personajes. -Crear la clase Enemy (enemigos) y Obstacle (obstáculos). -Programar los obstáculos para que aparezcan el mapa de forma aleatoria cada x tiempo. -Programar enemigos para que salgan en la parte superior de del mapa, y darles movimiento para que lleguen hasta la parte inferior del mapa y desaparezcan.	25 de diciembre- 2 de enero
4	-Implementar la vida y puntaje del personaje principal (class Character). -Programar interacciones entre los enemigos, los obstáculos, el personaje principal y las balas generadas por el personaje principal. -Crear la clase Bonus. -Generar Bonus (ayudas) cada cierto tiempo en el mapa.	3,4,5 de enero
5	-Inicio de sesión y registro de usuario. -Interfaz de menú principal.	6,7 de enero
6	-Crear niveles, noción de dificultad y transición para jefe final (pasar a vista horizontal).	8,9 de enero
7	-Crear clase boss (jefe final). -Crear la clase Power (poderes). -Darles movimiento a los jefes finales para cada nivel. -Implementar físicas con los poderes y obstáculos (de adorno si hace falta).	10-15 de enero
8	-Programar multijugador.	16,17 de enero
9	-Guardar y cargar partidas.	18-19 de enero
10	-Agregar sonido, música, efectos especial ...	20-21 de enero
11	-En lo posible mejorar gráficas y arreglo de errores del juego (pulir el videojuego).	22-23 de enero
12	-Revisiones finales del juego (mirar que todo funcione correctamente).	24-25 de enero
13	-Generar ejecutable.	25 de enero

5. Descripción del juego.

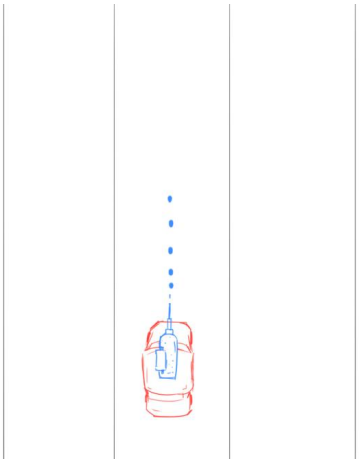
Nombre del juego: Encroachment.

Descripción: El juego consiste en defender la tierra de una invasión extraterrestre, para esto el personaje que la defenderá sera un automóvil.

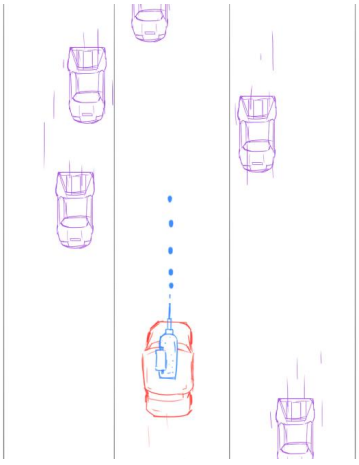
El juego tendrá tres mapas (en donde el nivel de dificultad irá aumentando por cada mapa).

Cada nivel tendrá un jefe final, para poder enfrentarlo hay que pasar por una serie de obstáculos y enemigos que trataran de eliminar al jugador.

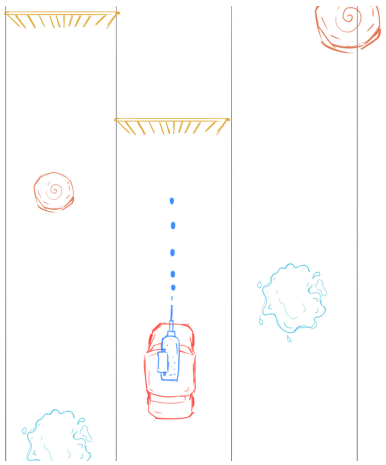
El auto tiene la capacidad de disparar.



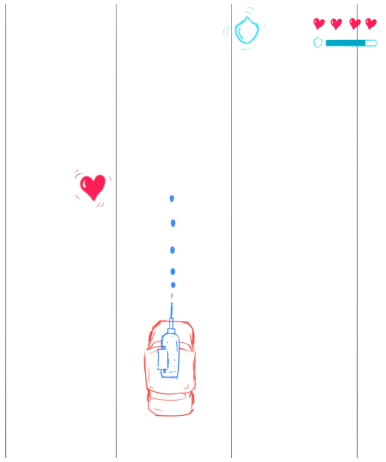
Por el camino vendrán enemigos que querrán eliminar al jugador principal.



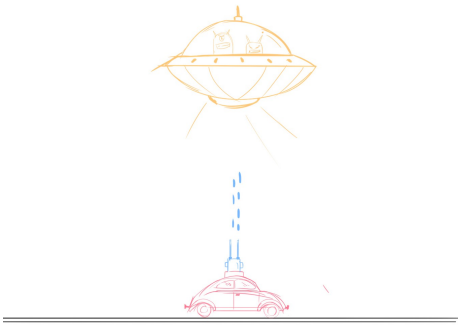
También habrá obstáculos que impidiran avanzar al carro y hay que intentar que esquivar, en caso de chocar podrá bajar la vida del auto, o efectos como ralentizarlo durante un tiempo.



Por el camino también podrán aparecer ayudas, que pueden potenciar el auto.



Al final para poder pasar el nivel hay que enfrentar un jefe final.



Una vez sea derrotado el jefe se desbloqueará el siguiente nivel, o si ya es el último nivel habremos terminado el juego.

EL personaje principal tendrá una barra de vida, que en caso de llegar a 0 perderá y tendrá que empezar el nivel de nuevo.

6. Definición de objetos.

6.1. Balas

/*Generar balas en el juego.*/

Class Bullet:

Atributos:

int damege; daño que genera la bala sobre un cuerpo al impactar.

Métodos:

Bullet() //constructor

-Bullet() //destructor

void Move() //mover la bala en cierta dirección

6.2. Personaje Principal

/*Personaje principal: Automóvil que tendrá la capacidad de generar disparos y contara con una barra de vida. Será manejado por un usuario.*/

Class Character:

Atributos:

float posx, posy //posición (x,y) en el mapa.

int health //Representa la salud del personaje

Métodos:

Character() //constructor

-Character() //destructor

void Shoot() //Generar disparos

void Move() //Moverse por mapa

void keyPressEvent(QKeyEvent *evento); //reconocer entrada del teclado

6.3. Enemigos

/*Enemigos: Automóviles que disparan contra el personaje con el fin de poner eliminarlo o que llegue con la menor vida al final.*/

Class Enemy:

Atributos:

int health //Representa la salud

Métodos:

Enemy() //constructor

-Enemy() //destructor

void Shoot() //Generar disparos

void Move() //Moverse en el mapa

6.4. Obstáculos

/*Obstáculos: Objetos que se encuentran en el camino del personaje principal con el fin de generarle algún tipo de daño*/

Class Obstacle:

Atributos:

float posx,posy //Posición (x,y) en el mapa

Métodos:

Obstaculos() //constructor

Obstaculos() //destructor

void generateDamage() //En caso de colisionar con el personaje principal este método generara un daño sobre el auto. el daño dependerá del obstáculo con el que se haya colisionado el personaje principal.

6.5. Poderes

/*Poderes que los jefe de fin de nivel podrán lanzar para intentar acabar con el personaje principal*/

Class Power:

Atributos:

int damege; daño que genera el poder sobre un cuerpo al impactar.

Métodos:

Power() //constructor

-Power() //destructor

void Move() //mover el poder del enemigo hacia el personaje principal

6.6. Jefe final

/*Enemigo del final de nivel, que el personaje principal debe derrotar para desbloquear el siguiente nivel o si es el último nivel para terminar el juego.*/

Class Boss:

Atributos:

float posx, posy //posción (x,y) en el mapa.

int health //Representa la salud.

Métodos: Boss() //constructor

-Boss() //destructor

void Shoot() //Generar disparos.

void Move() //Moverse en el mapa.

void throwPower() //Lanzar poderes.

6.7. Ventajas

/*Ventajas: Objetos que se encuentran en el camino del personaje principal con el fin de generarle algún tipo de beneficio.*/

Class Bonus:

Atributos:

float posx,posy //Posición (x,y) en el mapa

Métodos:

Bonus() //constructor

Bonus() //destructor

void generateBenefit() //En caso de colisionar con el personaje principal este método generara una ayuda sobre el auto, la ayuda dependerá del obstáculo con el que se haya colisionado el personaje principal

6.8. Otros

/*Objetos a conocer como hacer que funcionen*/

Class SignIn //Iniciar sección.

Class SignUn //Nueva cuenta.

Class DeleteAccount //Borrar cuenta.

Class NewGame //Guardar juego.

Class SaveGame //Guardar juego

Class LoadGame //Nuevo juego

Class DeleteGame //Borrar progreso del juego

Class Multiplayer()//poder jugar en modo multijugador

(POsiblemente) Class Users()//administar información de los usuarios

6.9. Nota.

-Con la adquisición de nuevos conocimientos los métodos y atributos de los objetos pueden variar.

-Además de los métodos y atributos ya descritos, también se incluirán los necesarios para poder graficar las clases en pantalla.