

Zheng, Alexandre - 776093
del Rincón de la Villa, Alonso - 783252
Gallén Recio, Antonio - 735184
Plo Andrés, Juan - 795105

0

Índice

1. Concepto del juego	2
2. Reparto de roles	3
3. Cuestiones de implementación	4
4. Estructura	5
5. Bucle del juego	6
6. Lógica del juego	7
Máquina de estados	7
Pantallas	8
7. Gráficos	12
8. Unidades, estructuras y recursos	13
9. Cámara	14
10. Tareas restantes	15

1. Concepto del juego

Se trata de un juego de estrategia en tiempo real(RTS) con vista top-down, de 1 vs 1. El modo de juego contra la IA tiene como objetivo destruir la estructura principal del enemigo ganando y terminando así la partida

Para ganar tendrás que armar un ejército que supere al del enemigo, o bien tendrás que idear alguna estrategia para despistar a los enemigos y hacerte con la victoria, para ello podrás aprovechar los recursos que se encuentran en el mapa, con los que poder mejorar tus unidades y aumentar tus tropas

Podrás seleccionar hasta 3 tipos de dificultad para probarte a ti mismo, donde la IA no te lo pondrá fácil para ganar, o quizás más probablemente, te ganará ella. Las partidas serán guardadas y podrás retomarlas en cualquier momento

Además, durante el transcurso de la partida, podrás acceder a un menú de opciones desde donde podrás cambiar los controles de tu teclado para mejorar tu experiencia y comodidad

2. Reparto de roles

Debido a que todos los integrantes compartíamos un conocimiento limitado con respecto al desarrollo de videojuegos, todos creamos la estructura del videojuego de una manera organizada y gracias a un diagrama de clases aprovechando la herencia, pudimos repartirnos las tareas y desarrollar en paralelo los distintos aspectos del juego:

Juan: Desarrollo de las unidades, comandos de las unidades, cálculo de caminos, estructura del mapa, algunos menús, carga de partida

Antonio: IA, desarrollo de las unidades, utilidades del código

Alonso: *Rip* del juego original, cámara, niebla de guerra, algunos menús, guardado y carga de partida

Alex: Desarrollo de las estructuras y sus funcionalidades, interfaces(GUI, ratón, menús, etc.), efectos de sonido y carga del mapa de juego

3. Cuestiones de implementación

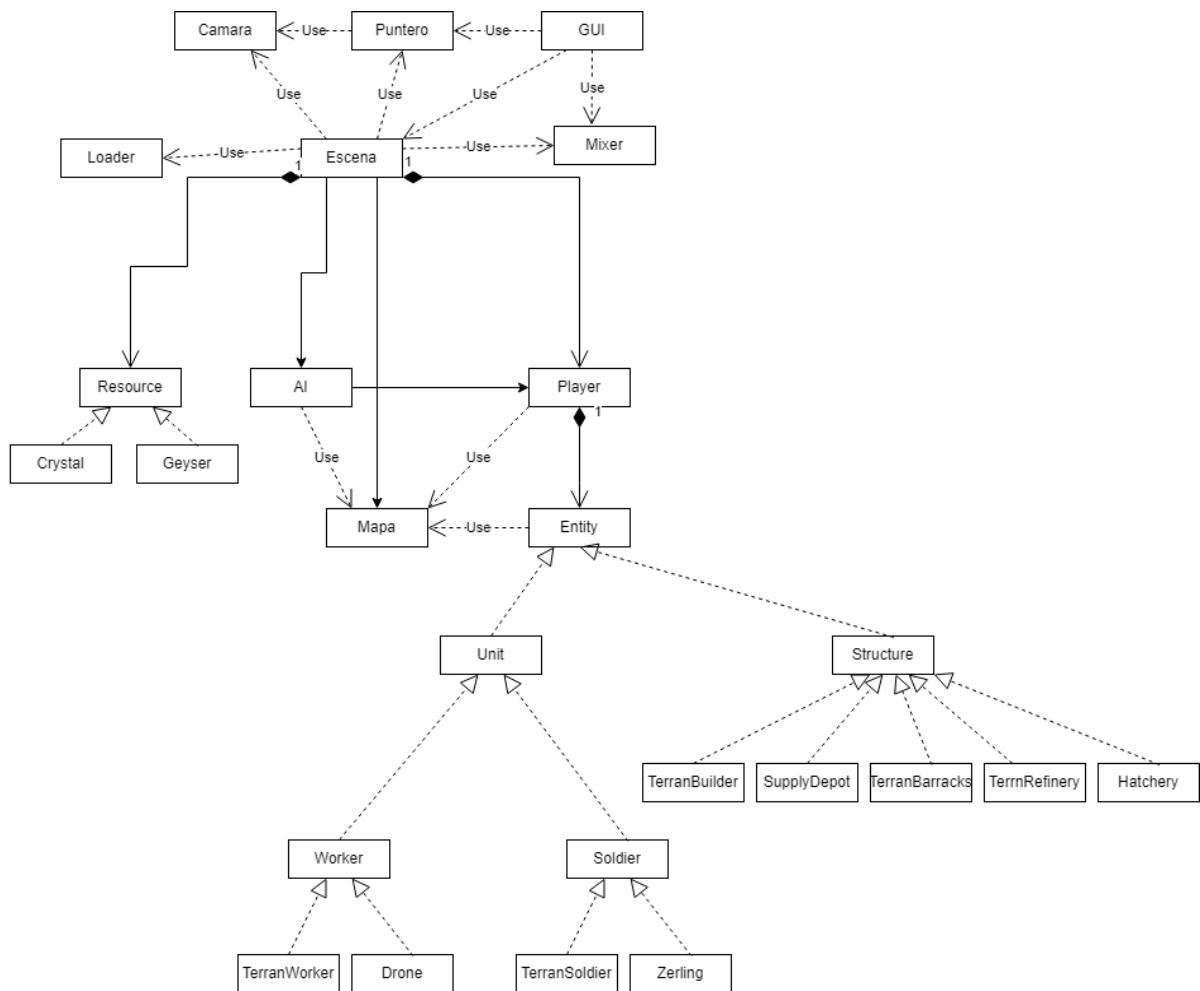
El juego ha sido desarrollado en PC, para ser jugado en Windows. La implementación se ha realizado en Python puesto que encontramos muy cómoda la inferencia de tipos y el tratamiento de listas del lenguaje

El juego funciona alrededor de una frecuencia de 60 frames por segundo, según las acciones que realiza el jugador. La resolución de la pantalla del juego es de 1025 x 770 fijo como el juego original, no se permite reajuste de pantalla, y no está pensado para jugar en pantalla completa

4. Estructura

Para el trabajo paralelo del equipo, la implementación está basada en el siguiente diagrama de clases

Diagrama de clases:



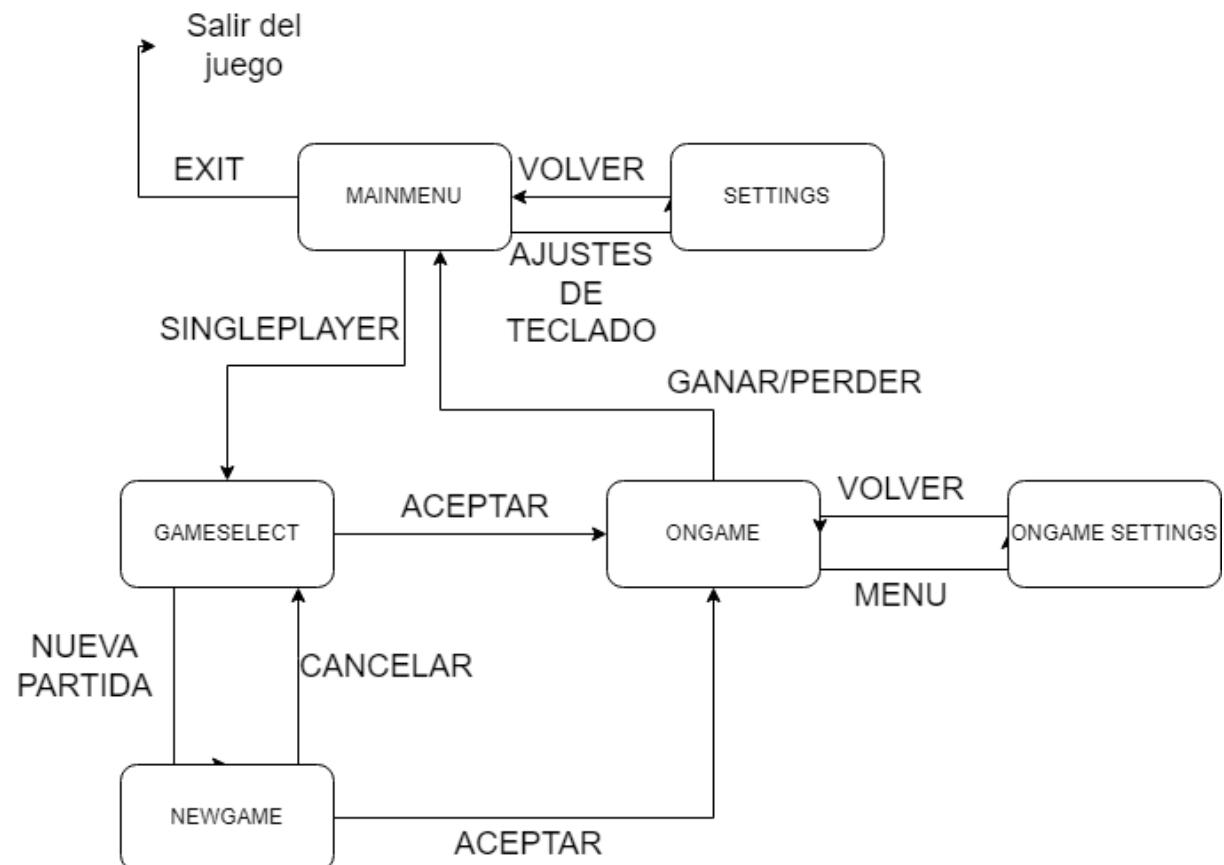
5. Bucle del juego

El programa principal se basa en primero procesar el primer evento de la cola de eventos del usuario(Procesar Inputs), actualizar el estado del juego en base al input, actualizar estado general del juego una vez aplicado el input y renderizar

6. Lógica del juego

El juego se basa en una máquina de estados, de cada estado depende cómo afectan los inputs y qué se actualiza y qué se renderiza

Máquina de estados



Pantallas

MAINMENU: Menú principal con opciones de Single Player para jugar, Exit para salir del juego y Ajustes



GAMESELECT: Pantalla para elegir jugar una partida guardada o crear una nueva partida.
(No disponible en la Beta)



NEWGAME: Pantalla para crear una nueva partida eligiendo Mapa, Dificultad y Raza.
(No disponible en la Beta)



INGAME: Pantalla donde se lleva acabo la partida, con información de los recursos en la parte superior derecha, la partida abarcando $\frac{3}{4}$ de la pantalla, un minimapa abajo a la izquierda, información de la unidad seleccionada en el centro inferior y los comandos de la estructura seleccionada en la parte inferior derecha



7. Gráficos

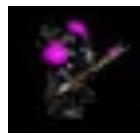
Los sprites utilizados para el desarrollo del juego han sido sacados tanto de la pagina: https://www.spriters-resource.com/pc_computer/starcraft/, como del propio juego, *ripped* gracias a las siguientes herramientas:

<http://www.zezula.net/en/mpq/download.html>
<https://sfsrealm.hopto.org/downloads/SFGrpConv.html>
<https://github.com/poiuyqwert/PyMS>

Todas las entidades y los recursos tienen un spritesheet en forma de columna y contienen todas las animaciones

8. Unidades, estructuras y recursos

- Trabajador terrano (SCV / Space Construction Vehicle): es la unidad principal, su función principal es minar y extraer gas, pero también sirve para el combate aunque dispone de poco daño solo puede atacar cuerpo a cuerpo
- Soldado terrano (Marine): la unidad de ataque, tu ejército se compondrá de soldados primariamente. Aunque siempre se pueden usar trabajadores como primera línea, ya que los soldados, pese a tener un rango más amplio y gran daño, tienen menos vida
- Base terrana (Command center): es la estructura central la raza terrana y la base de operaciones desde la cual se pueden decidir construir otros edificios, adquirir mejoras para el ejército del usuario y reclutar nuevos trabajadores



- Barracas (Barracks): estructura que permite el reclutamiento de soldados



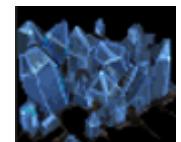
- Depósito de recursos (Supply Depot): estructura que amplía la cantidad de unidades que la colonia puede mantener simultáneamente



- Refinería de gas vespeno (Refinery): colocada sobre un géiser de gas vespeno permite la extracción de gas de éste por parte de los trabajadores



- Depósito de minerales: estructura neutral de las que cualquier trabajador de cualquier bando puede extraer minerales para almacenarlos en su base. La cantidad de mineral extraíble de cada depósito es limitada. El mineral es necesario tanto para la construcción de estructuras como para el reclutamiento de unidades y su mejora
- Géiser de gas vespeno: similares a los depósitos de minerales, permiten la extracción de gas si se ha construido una estructura que lo permita sobre ellos. El gas extraíble es limitado y es vital para la mejora de las unidades



9. Cámara

La cámara es una estructura de datos compuesta de 3 valores: sus coordenadas en el espacio del mundo, su ancho y su largo. En cada iteración del bucle del juego se comprueban las teclas asociadas a su movimiento, sus coordenadas se actualizan en base a la tecla pulsada

La cámara interviene en el render de todos los objetos, pues sólo se renderizan aquellos objetos que se encuentren dentro de los límites del rectángulo que constituye la cámara, transformando sus coordenadas a coordenadas en el espacio de la cámara, de forma que objetos que, incluso si se dibujaban, no aparecían en la pantalla por estar fuera de sus límites

Además permite la conversión de las coordenadas de los eventos de ratón, que vienen dadas en el espacio de la pantalla/cámara, al espacio del mundo

10. Tareas restantes

Existen distintas características interesantes que aún no están completamente desarrolladas y que se esperan completar para la versión final del videojuego:

- Añadir unidades que inciten el llevar a cabo una estrategia de organización del ejército
- Terminar la segunda raza (los Zerg) y añadirla para que el usuario disponga más opciones de juego, usable tanto por el propio jugador como para la IA
- Añadir un modo historia con misiones
- Rutas para los invasores de la IA precisas