

Documentacion Zombie Defense *

Gerson Ramirez

Maximo Agrazal

INTRODUCCIÓN

Este proyecto se trata de diseñar y programar un juego de estrategia con batallas por turnos. Los juegos de estrategia son aquellos juegos en los que, el factor de la inteligencia, habilidades técnicas, planificación y despliegue, pueden hacer predominar o impulsar al jugador hacia la victoria del juego, un ejemplo es el ajedrez. El objetivo del juego es defender una base del ataque continuo de zombies, usando personajes con diferentes habilidades y armas. En cada turno se pueden tomar acciones para mover a los personajes, atacar y usar ítems. Deben existir al menos 3 personajes diferentes con habilidades y características especiales. En el juego existe un punto de respawn (Respawn significa la regeneración de algún elemento dentro del juego), y la base a proteger por los jugadores. La aplicación contara con una interfaz sencilla que muestra el mapa y las posiciones de los personajes, zombies, puntos de respawn ,ítems.

Planteamiento del problema

Para empezar a desarrollar este proyecto, primero se hizo un diagrama de clases, en el que se formularon las clases, atributos y operaciones de cada una. Al momento de comenzar a escribir código, se empezó con un paquete modelo, el cual contiene las clases que se van a implementar para usar en la interfaz gráfica, para desarrollar este juego se utilizaron casillas en las que los personajes pueden moverse mientras no tengan un obstáculo. Los personajes tienen distintas habilidades y estadísticas (vida, ataque). Durante cada turno, los zombies se deberán mover y si tienen a un jugador cerca, lo atacarán. De igual forma con los jugadores hacia los zombies, también deberán poder obtener y consumir Ítems que los ayudaran a poder sobrevivir a las oleadas de zombies. Al final de cada turno, se sube de nivel y los stats del jugador deberán mejorar.

DISEÑO

Dado que en este formato, al colocar las imágenes se ven muy pequeñas y los diagramas necesitan ser grandes para que sean legibles, se adjuntarán junto con este pdf .

A. Cronograma

B. División de actividades

C. Diagrama de clases

D. Diagrama de casos de uso

E. Diagrama de Secuencias

I. MÉTODOS

Después de definir las clases con sus atributos y funciones, e implementar una interfaz gráfica para el proyecto hicieron falta métodos que pudiesen permitirnos aplicar todos los requerimientos que este trabajo tenía. Estos métodos eran necesarios para cada clase. Primero, detallaremos los métodos necesarios para el jugador, entre los principales estaba el de movimiento, el cual se logra haciendo otro método que le diga las coordenadas sobre espacios que estén desocupados, y a partir de ahí el jugador podrá elegir la casilla a la cual moverse. El método de movimiento se recicla con los zombies, el único detalle es que los zombies se mueven en ciertas condiciones.. El método para poder agarrar los ítems que se encuentran en algunas casillas, al pasar sobre una que tenga un ítem, este se agrega a un inventario y de ahí el jugador lo podrá consumir. Otro método fundamental es el de atacar, sirve tanto para zombies como para el jugador, si un zombie esta al alcance de un personaje, este usara algún arma que tenga, para poder eliminar al zombie y poder sobrevivir. Ahora, los zombies cuentan con varios métodos que ya han sido mencionados, el mas exclusivos de ellos son el respawn, o reaparecimiento, lo que les permite invadir de forma continua la base del jugador. Otras clases como las casillas, cuentan con pequeñas funciones para diferenciarlas entre ellas mismas, una casilla puede tener un obstáculo y por ende estar obstruida, o ser un punto de respawn, o una base del jugador. Esto va a depender mas que nada del diseño del mapa. También existen los ítems, estos objetos ayudan al jugador, por lo que al ser meramente objetos en el juego las funciones necesarias van a ser pocas. Para concluir con esta sección se puede decir que al fin y al cabo todos son necesarios para poder hacer un buen proyecto.

CONCLUSIÓN

El desarrollo de este proyecto nos ayudó a mejorar nuestro entendimiento y manejo sobre los paradigmas de herencia; dado que existían distintos tipos de enemigos o zombies, y también existían distintos tipos de personajes, armas e incluso de ítems. Para poder solucionar el proyecto también fue necesario el uso de hilos, los cuales nos permiten hacer múltiples

acciones al mismo tiempo durante el juego. Este proyecto fue bastante complicado, pero nos sirvió para aprender bastante sobre el funcionamiento de juegos de estrategia en turnos, y también para tener mas experiencia con Java, probablemente de los lenguajes que actualmente se siguen usando bastante. Lo cual nos es bastante útil para ser mejores programadores y dominar mas lenguajes.

REFERENCIAS

[1] T. JButton, "Transparencia de un JButton", Stack Overflow en español, 2020. [Online]. Available: <https://es.stackoverflow.com/questions/127986/transparencia-de-un-jbutton>. [Accessed: 19- Nov- 2020].

[2] C. JLabel?, E. Robertson and E. Lang, "Can I add an action listener to a JLabel?", Stack Overflow, 2020. [Online]. Available: <https://stackoverflow.com/questions/5260462/can-i-add-an-action-listener-to-a-jlabel>. [Accessed: 19- Nov- 2020].

[3]J. Sensei, "Ajustar una imagen al tamaño de un JLabel en Java", Jedicerocool.blogspot.com, 2020. [Online]. Available: <http://jedicerocool.blogspot.com/2013/01/ajustar-una-imagen-al-tamano-de-un.html>. [Accessed: 19- Nov- 2020].

[4]R. Torres et al., "¿Como generar números aleatorios dentro de un rango de valores?", Stack Overflow en español, 2020. [Online]. Available: <https://es.stackoverflow.com/questions/5390/como-generar-n>

[5]F. Ureña Gómez, "Eventos y Listeners en Java — Disco Duro de Roer", Discoduroderoer.es, 2020. [Online]. Available: <https://www.discoduroderoer.es/eventos-y-listeners-en-java/>. [Accessed: 19- Nov- 2020].

Enlace al repositorio: <https://github.com/MaxAndre22/Proyecto-2-POO>