

Proyecto 1 - Zombi Defense

Yuen Law

Semestre I, 2020

Objetivos

1. Aplicar conceptos de ingeniería de software.
2. Utilizar herencia para el diseño de clases.

Descripción

El proyecto se trata de diseñar y programar un juego de estrategia con batallas por turnos. El objetivo del juego es defender una base del ataque continuo de zombis, usando personajes con diferentes habilidades y armas. En cada turno se pueden tomar acciones para mover a los personajes, atacar y usar items.

Personajes Deben existir al menos 3 personajes diferentes con habilidades y características especiales. Por ejemplo, un personaje puede estar especializado en exploración y tener un rango de movimiento superior a los demás y ser capaz de escalar obstáculos. Cada personaje debe tener al menos 3 habilidades o características que se van obteniendo conforme avanza el juego. Los personajes tienen vida y pueden morir.

Ejemplos de habilidades:

- Una acción más por turno
- Brincar/escalar obstáculos
- Mayor rango de ataque con armas de rango
- Mayor rango de movimiento
- Posibilidad de evadir ataques
- Posibilidad de duplicar el daño si se ataca con arma de corto alcance
- Posibilidad de duplicar el efecto de curación de items
- Posibilidad de duplicar experiencia
- Algún tipo de arma inicial
- Doble capacidad de cargar items
- Doble arma

Enemigos Los enemigos del juego son controlados por el computador y son zombis que actúan con un set de reglas sencillo:

1. Si puede atacar, ataca.
2. Si ve a un personaje, avanza hacia él.
3. Si escucha un ruido, se mueve hacia el ruido.
4. Si no se cumple ninguna de las anteriores se mueve hacia la base.

Todos los zombis tienen un rango de visión. Deben existir también 3 tipos de zombis con habilidades especiales.

Ejemplos de habilidades zombi:

- Una acción más por turno
- Doble vida o defensa
- Mayor rango de visión
- Chance de venir acompañado

Armas Existen 2 tipos de armas: de largo y de corto alcance. Las armas de largo alcance deben tener un rango de alcance máximo. Además cada arma debe tener un nivel de ruido cuando se usa. Los zombis, si no ven ningún personaje se moverán hacia el ruido más alto. El ruido se resetea después de cada turno. El juego debe contar con al menos 2 armas de cada tipo con niveles de ruido diferente.

Items Cada vez que se elimina un zombi, este tiene un chance de soltar un ítem o arma. Deben existir al menos 3 tipos de ítems, por ejemplo, ítems para curar, para dar experiencia (aumentar de nivel), armas, etc.

Mapa El mapa del juego es un tablero. A un extremo del tablero se encuentra la base por defender. Esta base puede estar representada por ejemplo por una casilla (la entrada a la base) y una vez que algún zombi logra llegar, se pierde el juego. Cada casilla del mapa puede tener un obstáculo, representar una plataforma alta, tener una pared o puerta, etc.

Spawning points En cada turno hay un chance de que aparezcan más zombis en el mapa, en casillas especiales designadas como spawning points. Conforme avanza el juego deben aparecer nuevos spawning points en el mapa.

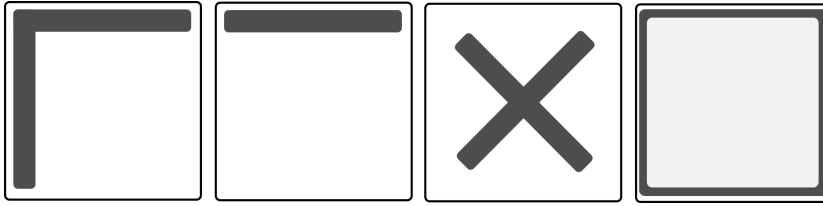


Figura 1: Ejemplo de casillas del mapa: de izquierda a derecha: con 2 paredes, con una pared, con obstáculo, plataforma elevada.

Resumen del turno

1. Cada personaje realiza sus 3 acciones: moverse, atacar, usar ítem, en cualquier orden.
2. Cada zombi realiza su acción
3. Se activan los spawning points
4. Se resetea el nivel de ruido

Requerimientos funcionales

- La aplicación debe contar con una interfaz sencilla que muestre el mapa y las posiciones de los personajes, zombis, spawning points, ítems, etc.
- Se debe proporcionar la interfaz adecuada para realizar las acciones.
- Se debe proporcionar la información necesaria para ver los resultados de las acciones.

Otros requerimientos

- Se debe programar usando el paradigma orientado a objetos. Debe usar herencia.
- Debe implementar una interfaz gráfica.
- Para el control de desarrollo se debe usar la pizarra de proyecto en Github y el sistema kanban para registrar el progreso de las actividades realizadas.
- Como parte de la documentación debe entregar un diagrama de clases UML completo, un diagrama de casos de uso y un diagrama de secuencia para el caso de uso jugar turno.

Entregables

1. Documentación PDF, en el formato IEEE para conferencias (máximo 8 páginas) con las siguientes secciones: Introducción (planteamiento del problema), Diseño (Diagrama de clases, Diagrama de casos de uso, diagramas de secuencia (para el caso de uso: jugar turno), división actividades y cronograma), métodos (descripción detallada de la solución), Conclusiones y Referencias.
2. Código en un repositorio de Git. El último commit realizado debe ser antes de la fecha y hora límite de entrega.

Aspectos administrativos

1. Debe completar una reunión de avance al menos una semana antes de la entrega para un 5 % de los 30 que vale el proyecto.
2. Fecha de entrega: viernes 13 de noviembre, hora: 12 media noche
3. El pdf debe ser enviado por medio de Teams
4. El proyecto puede ser desarrollado en Java.
5. Aplican los puntos del documento "Aspectos Administrativos" entregado al inicio del curso.

Evaluación

Rubro	Valor
Documento PDF	(30)
- Introducción	2.5
- Diseño	10
- Métodos	10
- Conclusión	2.5
- Referencias	5
Código	(60)
- GUI	10
- Lógica de juego (turnos)	10
- Reglas de zombis	10
- Personajes/zombis	10
- Armas/items	10
- Mapa/casillas	10
Workflow	(10)
- Uso de git	5
- uso del board de github	5
Total	100