

Left 4 Dead

Índice

[Introducción](#)

[Descripción](#)

[Infectados](#)

[Jumper](#)

[Wich](#)

[Spear](#)

[Venom](#)

[Jugador](#)

[IDF](#)

[P90](#)

[Scout](#)

[Granadas](#)

[Bombardeo aereo](#)

[Puntos de vida, derribo y muerte](#)

[Ataque y colisiones](#)

[Escenario](#)

[Finalización del juego](#)

[Interfaz gráfica](#)

[Sonidos](#)

[Música](#)

[Animaciones](#)

[Configuración](#)

[Cheat](#)

[Aplicaciones Requeridas](#)

[Servidor](#)

[Cliente](#)

[Restricciones](#)

[Referencias](#)

Introducción

El trabajo consiste en implementar un juego 2D multijugador inspirado en el famoso Left 4 Dead.

En él los jugadores representarán soldados en un mundo post-apocalíptico donde la mayoría de la población ha sido infectada por un patógeno desconocido.

Descripción

Infectados

Los infectados **caminan** o permanecen **quietos** en las calles si no son molestados pero la mayoría empezará a caminar hacia los jugadores en cuanto estén a la vista para matarlos (sea a través de golpes, zarpazos o mordidas).



Algunos incluso llegan a **correr** y a **saltar**. Esto es característico de todos los infectados sin embargo algunos han evolucionado y han adquirido destrezas específicas.

Jumper

Este infectado tiene la particularidad de **saltar** sobre sus víctimas y producirles daño al caer sobre ellas. El jugador puede ver que el infectado está por saltar por su característica pose en el piso. Si es lo suficientemente rápido de reflejos, podrá correr y evitar el ataque.



Witch

Esta infectada **ocasionalmente** se pone a gritar **atrayendo a más infectados** que llegaran corriendo a la zona desde todas las direcciones. Los jugadores deberían eliminarla lo antes posible.



Spear

A pesar de que la mayoría de los infectados muestran un nivel cognitivo básico, algunos han llegado a evolucionar a tal punto de ser capaces de usar armas como lanzas.



Venom

Con una apariencia totalmente humana, Venom lanza un líquido/gas corrosivo. Puede lanzarlo como un **spray desde corta distancia** (como se muestra en la figura) así como también puede **lanzar bolas de este líquido desde largas distancias**.

El spray daña a todo jugador que esté en contacto con él.



Jugador

Los jugadores podrán elegir entre 3 tipos de soldados que se distinguen por las armas que llevan: IDF, P90 y Scout.

Todas las armas tienen munición infinita, no obstante luego de vaciar un cargador el soldado deberá hacer una recarga.

Con la excepción del jugador que porta el P90, el resto de los soldados pueden lanzar granadas tanto explosivas como de humo.

IDF

Este rifle de asalto produce una rafaga de 20 balas con un daño considerable a corta distancia pero uno mucho menor a larga distancia reflejando su imprecisión.

Debe recargar el arma cada 50 rafagas.



P90

Con rafagas de 10 balas, es un rifle más balanceado con una performance más consistente en largas distancias.

Debe recargar el arma cada 30 rafagas.



Scout

A diferencia del IDF y P90, el Scout es un rifle que dispara de a 1 bala. No obstante cada bala daña a todo infectado que se encuentre en su trayectoria (aunque el daño de cada impacto se verá reducido conforme la bala pasa por más infectados).

Se recarga cada 20 balas.



Granadas

Las granadas tienen munición infinita no obstante no pueden ser lanzadas consecutivamente (hay un tiempo de espera mínimo entre lanzamientos).

Las granadas convencionales producen una explosion q daña a todos los infectados de la zona. Cuidado que la explosion puede dañar a otros jugadores.



Las granadas de humo no producen daño pero los infectados quedan aturdidos y dejan de correr y saltar por cierto tiempo aunque siguen caminando. Las granadas de humo son una forma efectiva de ralentizar una

horda de infectados y ganar tiempo.



Bombardeo aereo

El jugador con la P90 no puede lanzar granadas pero puede iniciar un bombardeo aéreo. El bombardeo dejará caer granadas por todo el escenario salvo la zona cercana al jugador.

Tiene munición infinita pero tiene un tiempo mínimo de espera entre bombardeos mucho mayor que el tiempo de espera de las granadas de los otros soldados.



Puntos de vida, derribo y muerte

Cuando un jugador recibe demasiado daño y sus puntos de vida llegan a 0 este es **derribado**. Otro jugador puede asistirlo y levantarlo.

El jugador caído se levanta recuperando la mitad de sus puntos de vida.

Un jugador que es derribado por tercera vez **cae muerto**. Un jugador derribado que no es asistido por un compañero en el plazo de 30 segundos muere también.



Ataque y colisiones

Para determinar si un soldado está efectivamente disparándole a un infectado o viceversa, si un infectado está efectivamente dañando a un soldado hay que determinar si ambos están alineados.

Como el juego es 2D en un escenario pseudo-3D definiremos para cada infectado/soldado una región rectangular que va de los pies a un poco más de las rodillas.

Habrà alineación (y por ende habrá daño) si estas regiones rectangulares se solapan en el eje vertical. En la siguiente imagen, en solo los 2 primeros casos hay alineación.



Tanto los infectados como los soldados pueden colisionar entre sí. Es decir, no pueden “ocupar” el mismo espacio y que uno no puede avanzar por encima del otro.

Usando la misma región regular podemos definir que hay colisión si las regiones colisionan.

En la figura anterior, en los 2 primeros casos, no hay colisión pero si el infectado continuará moviéndose hacia la derecha, eventualmente colisionará con el soldado.

En el caso 5 no hay colisión pero el infectado no podría moverse hacia abajo por que colisionara con el soldado

Escenario

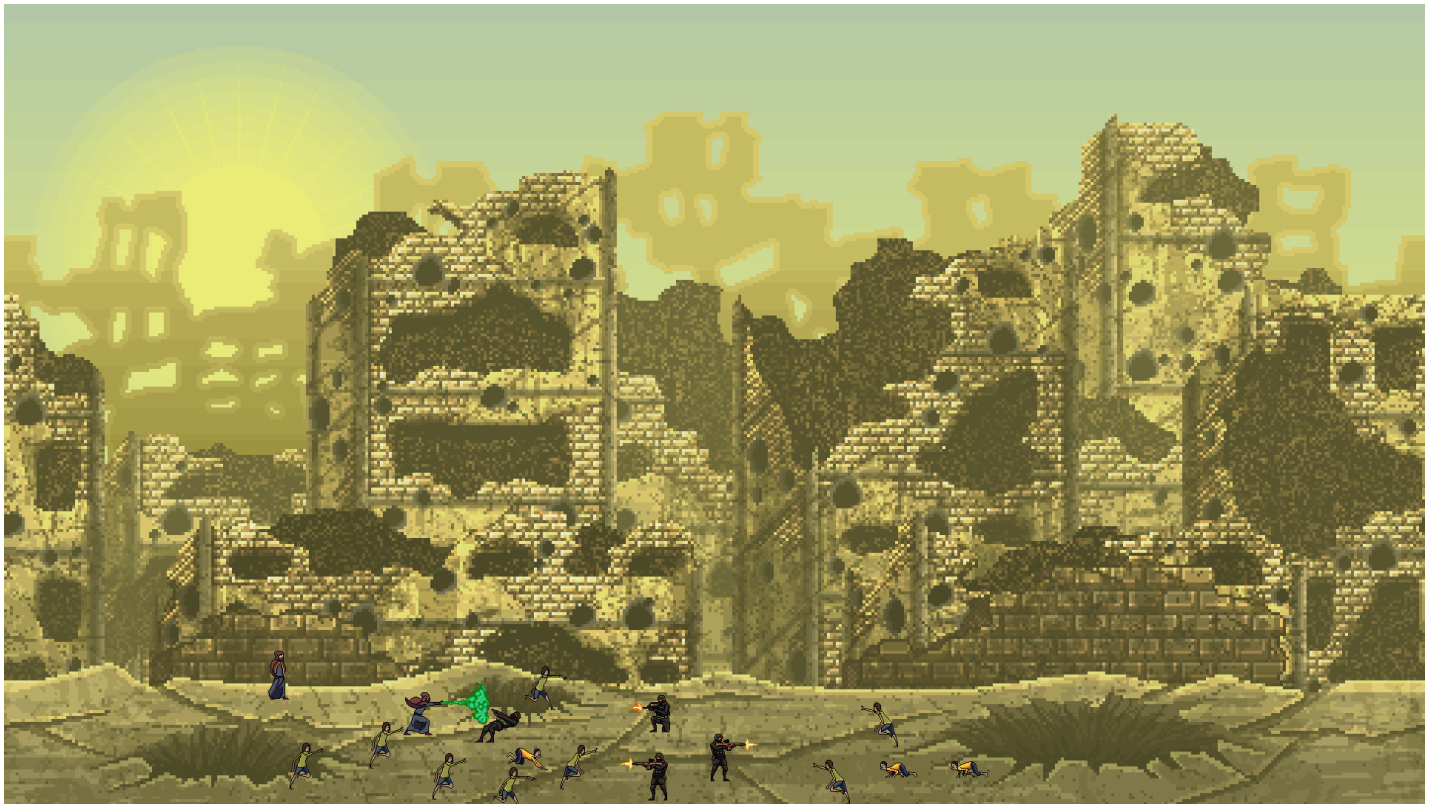
Este consta de un terreno en el cual tanto los jugadores como los infectados pueden moverse. En él hay cráteres y grietas gigantes que **obstaculizan** el paso.



Lo que se muestra en pantalla es solo una parte del escenario: la **cámara** sigue al grupo de jugadores y se mueve con ellos.

Los jugadores pueden desplazarse tanto a la derecha como a la izquierda para ir avanzando en el mapa. Cuando se llega a un extremo el mapa “continúa” en el opuesto (como si fuese un *buffer circular*) lo que da la ilusión de un mapa infinito.

El fondo de los escenarios está compuesto por múltiples fondos que se desplazan a diferentes velocidades para dar la ilusión de **paralaje**.



Ejemplo del escenario. La cámara sigue a los jugadores y trata de mantenerlos en el centro de la imagen. Si el grupo de jugadores se acerca lo suficiente a uno de los bordes, la cámara se mueve revelando nuevas partes del escenario.

Finalización del juego

Hay 2 modos de juegos: modo “**clear the zone**” y “**survival**”.

En “**clear the zone**” la cantidad de infectados en un escenario es **fija** y los jugadores ganan cuando eliminan a todos ellos.

En el modo “**survival**”, a medida que pasa el tiempo aparecen **más y más infectados** y estos se van haciendo **progresivamente más fuertes**, resistentes y veloces.

No hay condición de victoria: la muerte es inevitable y solo se compite por ver quién resiste más tiempo.

Al finalizar una partida se computa la cantidad de infectados eliminados, la cantidad de balas disparadas y el tiempo de la partida.

En el modo “**survival**” estas estadísticas se guardan en el servidor y muestran a los jugadores en que ranking han quedado (hay 3 rankings, uno por cada una de las estadísticas).

Interfaz gráfica

Además de la vista del escenario con los soldados e infectados, se debe mostrar la vida de cada soldado (con barras de vida), la cantidad de rafagas restantes antes de recargar y el tiempo restante para poder lanzar una nueva granada.

También se debe mostrar un contador de balas disparadas en la partida, infectados eliminados y del tiempo transcurrido.

Controles

El jugador controla al soldado vía el teclado sea para moverlo como para disparar y recargar. En el caso de la recarga esta sucede automáticamente si ya no hay más balas o si el jugador presiona la tecla correspondiente.

Para el lanzamiento de las granadas, que tan lejos se lanza dependerá de por cuánto tiempo se presione la tecla: un tiempo corto, la granada alcanzara una distancia corta (pero lo suficientemente lejos para que no dañe al jugador); un tiempo largo, la granada alcanzara una distancia larga.

Sonidos

Como todo juego se debe reproducir sonidos para darle realismo a los eventos y acciones que suceden:

- Cuando hay disparos.
- Cuando hay una explosión.

Si la cantidad de eventos que suceden es muy grande, algunos sonidos deben ser evitados para no saturar al jugador con tanta información.

Música

Se debe reproducir una música de fondo.

Animaciones

Las unidades no son mostradas con imágenes estáticas sino con pequeñas animaciones: debe mostrarse cómo se mueven, atacan, recargan, lanzan granadas, son derribadas o son destruidas.

Configuración

Todos los atributos en este enunciado son a modo de ejemplo y deben poderse cambiar vía un archivo de configuración.

Por ejemplo se debe poder cambiar el daño por bala por arma; la cantidad de balas en cada rafaga; la cantidad de rafagas antes de cada recarga; la velocidad de cada infectado; la probabilidad de que una witch grite; etc.

La idea es que no haya valores hardcoded en el código: el archivo de configuración te permitirá ir tuneando el juego para hacerlo más fácil o difícil y le permitirá al docente probar rápidamente una funcionalidad sin tener que jugar mucho tiempo (por ejemplo al setear la vida de los soldados en 1 es trivial probar el derribo/muerte).

Cheat

El juego debe tener cheats (trucos) que permitan **probar más fácilmente el trabajo**. Cuales y como ejecutarlos están definido por ustedes (se recomienda ser pragmáticos y buscar la facilidad)

Testing

Testear de forma automática todos los componentes de un juego en general es complicadísimo y en muchos casos imposible de hacer en su totalidad.

Sin embargo hay partes que son fácilmente testeables y que deberán presentar:

- Tests automáticos de todo el protocolo de comunicación entre el cliente y servidor.
- Tests automáticos de la persistencia y carga de las estadísticas.
- Tests automáticos del sistema de colisiones.

Nota: para el testeo del protocolo puedes usar sockets reales y testear con tiburoncín o puedes usar sockets fake. Ambas opciones tienen sus pros y contras. No te cases con ninguna y hace una PoC para ver cual te conviene.

Aplicaciones Requeridas

Servidor

El juego a implementar es multijugador con una arquitectura cliente-servidor. El servidor deberá recibir por parámetro la ruta a un archivo de configuración que contendrá:

- Puerto donde escuchar
- Rutas a los archivos que contienen los diferentes mapas.
- Parámetros del juego: el daño de cada unidad, las velocidades, todos los valores numéricos descritos en el presente enunciado deben ser configurables desde un archivo de texto. Esto es **esencial** para optimizar la jugabilidad y balancear las fortalezas y debilidades de las unidades sin tener que recompilar el código.

El servidor deberá manejar persistencia para mantener un registro de las estadísticas del juego.

Cliente

Es el elemento central que interactúa con el jugador. Debe poder mostrarle una pantalla de login para definir a qué servidor quiere conectarse, qué mapa desea jugar, con qué soldado y el modo. El servidor le generará un número de partida único para que otros jugadores se puedan unir.

Ya iniciado el juego, debe ofrecer una interfaz rica de acuerdo a los requerimientos pedidos y al finalizar cada partida debe presentar las estadísticas finales y si el modo de juego fue **"survival"**, el ranking de estas.

Restricciones

La siguiente es una lista de restricciones técnicas exigidas:

1. El sistema se debe realizar en ISO C++17 utilizando librerías QT y/o SDL.
2. La información de configuración debe ser almacenada en formato YAML. No está permitido utilizar una implementación propia de lectura y escritura de YAML: deben usar una librería.
3. Es condición necesaria para la aprobación del trabajo práctico la entrega de la documentación

mínima exigida (consultar sitio de la cátedra). Es importante recordar que cualquier elemento faltante o de dudosa calidad pone en riesgo la aprobación del ejercicio.

Referencias

[1] https://en.wikipedia.org/wiki/Left_4_Dead