Manual de Proyecto

Índice

Organización semanal

Semana a semana nos fuimos organizando con el objetivo de ir llevando al dia los objetivos propuestos por la cátedra tal cual están descriptos en el enunciado.

Semana 1

- Pruebas de concepto de SDL, Qt y Box2D.

Semana 2

- Inicio del armado del protocolo de comunicación.
- Armado de la estructura cliente-servidor, colas de entrada y salida, hilos de enviado y recepción.

- Implementación de la estructura cliente-servidor para 1 partida.
- Implementación de envios, colisiones y renderización de vigas.
- Implementación de envios, colisiones y renderización de un gusano o más.
- Implementación de envios, colisiones y renderización de movimiento de un gusano o más.

Semana 3

- Implementación de multiples partidas.
- Conectar múltiples clientes a una misma partida.
- Manejo de turnos de gusanos.
- Implementación basica del menu de partidas.
- Implementación de golpe de gusano con bate.
- Creacion de equipos por cliente.
- Implementación de mover gusano en vigas con inclinación.

Semana 4

- Implementación del salto de los gusanos.
- Ejecución de mas de una partida en simultaneo.
- Implementación básica de la cámara en la vista.
- Implementación de temporizador.
- Implementación ataque con granada Verde.
- Implementación ataque con granada Santa.
- Implementación ataque con granada Banana.
- Implementación ataque con Dinamita.
- Implementación ataque con Bazooka.
- Implementación de test automatizados para la comunicación.
- Cambio de turno con lapso de tiempo luego de ataque.

Semana 5

- Implementación de la herramienta teletransportación.
- Implementación de salto hacia atrás.
- Refactorización de comunicación de enviado de Proyectiles (enviado de múltiples proyectiles).
- Implementación de Ataque aereo.
- Implementación de Mortero.
- Implementación de Granada Roja.
- Arreglo de vista de animación en ambos clientes al ejecutar ataque.
- Boceto MANUAL DE USUARIO.
- Creación de instalador.

Semana 6

- Correcciones en el Ataque Aereo.
- Implementación de fragmentos de Granada Roja y Mortero.
- Cambio de renderización del agua.
- Agregado de pantallas de victoria y derrota.
- Implementación completa de Menú de partidas.
- Renderización de salto para atras.
- Implementación de muerte al caer al vacio.
- Implementación de vista de temporizador de las granadas.
- Boceto Manual de Proyecto.
- Diagramas de estructura del proyecto.
- Implementación de visualización de vida total por equipo.
- Cámara sigue a un misil y worm golpeado

Semana 7

- Reducción del consumo de la CPU de los clientes.
- Arreglo de bugs generales.
- En partidas múltiples, el abandono de un cliente tiene como consecuencia la muerte de sus Worms y la partida continúa para los demás.
- Documentación terminada.
- Diagramas de clases.
- Diagramas de secuencia.
- Música ambiente y sonido.
- Feedback de armas.
- Implementación completa de la cámara en la vista. Seguimiento de Worms golpeados, seguimiento de proyectiles y movimiento de la cámara con el mouse.
- Pruebas manuales del juego.
- Mejora del menú.

Repartición de tareas

El proyecto fue dividido en tres categorias:

- Vista
- Lógica del juego y físicas
- Comunicación, Concurrencia y Estructura del juego

El alumno responsable en llevar a cabo la lógica del juego y parte física fue **Mateo Julián Rico**. En particular se encargó de:

- Implementar la lógica del juego (movimientos, ataques, turnos, etc).
- Modelar el movimiento de los gusanos y los proyectiles mediante el uso de Box2D.

El alumno responsable en llevar a cabo la lógica de la Comunicación, Concurrencia y Estructura del juego fue **Federico Solari Vazquez.** En particular se encargó de:

- Implementar los protocolos del comunicación del servidor.
- Implementar los protocolos del comunicación del cliente.
- Implementar la lógica de armado de múltiples partidas.
- Implementar la estructura de el cliente y el servidor (colas, threads…).
- Implementar los cierres ordenados al terminar la partida o cuando hay un error.
- Hacer diagramas de estructura del proyecto.
- Implementación de test automáticos de la comunicación haciendo mocks de socket.

El alumno responsable en llevar a cabo la lógica de la vista fue **Alan Valdevenito**. En particular se encargó de:

- Implementar la vista del juego mediante SDL2pp. Esta implementación incluye el manejo de eventos, el renderizado y el sonido.
- Implementar el menu principal mediante QT. Esta implementación incluye un menu de partidas donde el cliente puede unirse a partidas que ya fueron creadas y un menu para crear nuevas partidas.

Herramientas utilizadas

- Google Test.
- Box2D.
- SDL2pp.
- QT.
- Visual Studio Code.
- Live Share (Visual Studio Code).

Documentacion requerida para utilizacion de las herramientas

- Google test: https://github.com/google/googletest

- Box2D: https://box2d.org/documentation/ https://www.iforce2d.net/b2dtut/
- SLD2pp: https://github.com/libSDL2pp/libSDL2pp
- QT: https://doc.qt.io/

Documentación adicional

- Link a tablero de tickets: https://github.com/users/AlanValdevenito/projects/1

Información Adicional

¿Cuales fueron los puntos más problemáticos?

- Cambio de enviado de un proyectil a múltiples proyectiles.
- Cierres ordenados de partidas.
- Ensamble de Google Test con el proyecto.
- Disponibilidad de canales de sonido. Al querer reproducir muchos sonidos simultáneamente, no habían canales disponibles para ello y generaba un error de SDL Mix. Explicación en el commit `a5f8aafe4b14bf786503f3751405024f960e170c`.
- Renderizado de objetos que sólo son visibles en la cámara, para mejorar la performance del CPU de los clientes.
- Cerrar partida de un cliente y que todos sus gusanos mueran.

¿Pudieron llegar con todo?

- Feature de las provisiones. Debido al tiempo que nos quedaba para la entrega decidimos no implementarlo para invertir tiempo en otras funcionalidades.
- Feature del viento para las armas. Debido al tiempo que nos quedaba para la entrega decidimos no implementarlo para invertir tiempo en otras funcionalidades.

¿Hay errores conocidos?

- En ocasiones el misil de la bazooka y el mortero rebota contra las vigas, cuando en realidad debería explotar en el momento en el que colisiona.

- Si se cierra una ventana del menu principal, el lobby se bloquea esperando la seleccion del cliente y no puede unirse ningun otro.
- Algunas ventanas del menú principal no son responsive.
- Si la posicion de algun Worm es negativa al cliente no le llegan de forma correcta las posiciones.

Si volvieran hacer el proyecto, que cambiarían (a nivel código o a nivel organizacional).

- En cuanto a organización no cambiariamos nada, creemos que nos organizamos bien, se puede ver reflejado en el tablero realizado en GitHub.
- Se hizo un buen slicing de los features a tratar de modo tal que trabajamos de a un feature a la vez y este era relativamente corto.
- Refactorizaríamos el código de la clase Game que por una decision de invertir tiempo en otras funcionalidades no pudimos hacerlo.