

Organizacion semanal

Semana a semana nos fuimos organizando con el objetivo de ir llevando al dia los objetivos propuestos por la catedra 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 comunicacion.
- Armado de la estructura cliente-servidor, colas de entrada y salida, hilos de enviado y recepcion.
- Implementación de la estructura cliente-servidor para 1 partida.
- Implementacion de envios, colisiones y renderizacion de vigas.
- Implementacion de envios, colisiones y renderizacion de un gusano o mas.
- Implementacion de envios, colisiones y renderizacion de movimiento de un gusano o mas.

Semana 3

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

Semana 4

- Implementacion del salto de los gusanos.
- Ejecucion de mas de una partida en simultaneo.
- Implementacion basica de la camara en la vista.
- Implementacion de temporizador.
- Implementacion ataque con granada Verde.
- Implementacion ataque con granada Santa.
- Implementacion ataque con granada Banana.

- Implementacion ataque con Dinamita.
- Implementacion ataque con Bazooka.
- Implementacion de test automatizados para la comunicacion.
- Cambio de turno con lapso de tiempo luego de ataque.

Semana 5

- Implementacion de la herramienta teletransportacion.
- Implementacion de salto hacia atras.
- Refactorizacion de comunicacion de enviado de Projectiles (enviado de multiples proyectiles).
- Implementacion de Ataque aereo.
- Implementacion de Mortero.
- Implementacion de Granada Roja.
- Arreglo de vista de animacion en ambos clientes al ejecutar ataque.
- Boceto MANUAL DE USUARIO.
- Creacion de instalador.

Semana 6

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

Semana 7

- Reducción del consumo de la CPU de los clientes.
- Arreglo de bugs generales.
- En partidas multiples, el abandono de un cliente tiene como consecuencia la muerte de sus Worms y la partida continua para los demas.
- Documentacion terminada

- Diagramas de clases
- Musica ambiente y sonido
- Feedback de armas
- Implementacion completa de la camara en la vista. Seguimiento de Worms golpeados, seguimiento de proyectiles . movimiento de la camara con el mouse
- Pruebas manuales del juego
- Mejora del menú.

Reparticion de tareas

El proyecto fue dividido en tres categorias:

- Vista
- Logica del juego y fisicas
- Comunicacion, Concurrencia y Estructura del juego

El alumno responsable en llevar a cabo la logica del juego y parte fisica fue Mateo Julian Rico. En particular

se encargo de:

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

El alumno responsable en llevar a cabo la logica de la Comunicacion, Concurrencia y Estructura del juego fue Federico Solari Vazquez. En particular

se encargo de:

- Implementar los protocolos del comunicacion del servidor.
- Implementar los protocolos del comunicacion del cliente.
- Implementar la logica de armado de multiples 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.
- Implementacion de test automaticos de la comunicacion haciendo mocks de socket.

El alumno responsable en llevar a cabo la logica de la vista fue Alan Valdevenito. En particular se encargo de:

- Implementar la vista del juego mediante SDL2pp. Esta implementacion incluye el manejo de eventos, el renderizado y el sonido.
- Implementar el menu principal mediante QT. Esta implementacion 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 multiples proyectiles.
- Cierres ordenados de partidas.
- Ensamble de google test con el proyecto.

- Disponibilidad de canales de sonido (al querer reproducir muchos sonidos simultaneamente, no habian canales disponibles para ello y generaba un error de SDLMix)

explicacion en commit `a5f8aafe4b14bf786503f3751405024f960e170c`

- Renderizado de objetos que solo son visibles en la camara, para mejorar la performance del CPU de los clientes.
- Cerrar partida de un cliente y que todos sus gusanos mueran.

¿Pudieron llegar con todo?

- Quedo pendiente implementar el feature de las provisiones. 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 menu 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 cambiaríamos nada, creemos que nos organizamos bien, se puede ver reflejado en el tablero realizada en GitHub.
- Se hizo un buen slicing de los features a tratar de modo tal que trabajabamos de a un feature y este era relativamente corto.
- Refactorizaríamos el codigo de la clase Game que por una decision de invertir tiempo en otras funcionalidades no pudimos hacerlo.