Trabajo Práctico Final - Worms

Edson Justo Narcizo, *Padrón Nro. 97775*
justo.edson@gmail.com
Franco Liberali , *Padrón Nro. 99491*
franco.liberali@gmail.com
Bautista Canavese, *Padrón Nro. 99714*
bauti-canavese@hotmail.com

Taller de Programación I - Cuatrimestre I, 2018
Facultad de Ingeniería, Universidad de Buenos Aires

12 de junio de 2018

Índice

1. Manual de Proyecto

1.1. División de tareas

La division de tareas fue seguida como indica la sugerencia en el enunciado del tp.
Franco: Servidor. Edson: Cliente. Bautista: Editor y varias implementaciones graficas.

1.2. Herramientas

En general se utilizaron las herramientas de compilación de gcc y gdb como debugger, además de cmake para la fabricación de makefile para luego, lo que permitió ser mas flexible en cuanto a el linkeo de librerias, etc.

1.2.1. Cliente

El cliente, que desarrolla sobre una GUI (interfaz de usuario gráfica), se utilizo la librería `sdl` y `gtk` para la implementación de menus y selección de mapas.

1.2.2. Servidor

El servidor, debe realizar todas las operaciones en cuanto al movimiento de los objetos y la física del mismo; por lo tanto se utilizo `Box2D` para todas las físicas.

1.2.3. Editor

En cuanto al editor, al ser estático en cuanto a las animaciones de las imágenes se decidió implementar con `qt`, además de la utilización de el editor grafico `qt editor`.

2. Documentación Técnica

2.1. Requerimientos de software

SO: Linux.
Libs: yaml, sdl, sdl mixer, Box2D, qt5.
Compile: cmake, makefile.

2.2. Descripción general

El proyecto se compone de tres módulos:

1. Servidor.
2. Cliente.
3. Editor.

2.3. Modulo Editor

Permite la edición de un escenario para el juego

- Elegir el fondo de pantalla.
- Colocar las vigas y gusanos.
- Definir qué armas y herramientas pueden usarse y la cantidad de municiones de ellas.
- Configurar la vida inicial de los gusanos.

2.3.1. Objetos

Implementados como un item de la escena padre (donde se encuentran dibujados), asignándole una imagen dependiendo si es una viga (chica o grande) o un gusano (del equipo: 1,2,3,4).

2.3.2. Ventanas

- Ventana principal: con los botones de acceso a la creación de objetos y guardado de mapa.
- Ventana armas: permite configurar las armas, cambiando su munición y seleccionando si pueden usarse o no.
- Ventana de gusanos: permite la configuración de la vida de los gusanos.

2.3.3. Guardado de mapa

Dentro de la clase `Registro` se irán almacenando los cambios en tiempo real sobre el editor, finalmente cuando el usuario seleccione `archivo/guardar como`, se dispondrá el lugar y el nombre en el donde se quiere guardar. Esto genera un archivo *yaml* con las siguientes características:

Luego de esta sentencia se indicaran todos los objetos que formaran el mapa, tanto los gusanos como las vigas.

```
-objetos:
```

Los objetos dispondrán del siguiente formato:

```
-objetos:
 -tipo: string<nombre>
 -angulo: float<radianes>
 -x: float<metros>
 -y: float<metros>
 -equipo: int<equipo>
```

Luego de esta sentencia se indicaran todas las armas que forman el mapa.

```
-armas:
```

Las armas dispondrán del siguiente formato:

```
-armas:
 -tipo: string<nombre>
 -disponible: string<si,no>
 -municion: int<municion>
```

2.4. Protocolo Servidor

```
todos los int son mandados en 4 bytes big endian
todas las coordenadas se encuentran en MILIMETROS y los
angulos en GRADOS de 0 a 360.

Strings:
int largo del string
el string en si sin el '\0' del final

ASIGNACIÓN DE ID DE JUGADOR(se manda solo al jugador que corresponde):
char en 0.
int con numero de jugador.

CONECCION DE UN JUGADOR (se manda a todos los jugadores. En caso de que
un jugador se desconecte y se conecte otro la idea seria volver a mandar
 este mensaje con el mismo numero de jugador y otro nombre, para
 sobreescribir el anterior):
char en 1.
int numero de jugador.
string nombre del jugador.

CREACIÓN DE UNA VIGA EN EL MAPA:
char en 2.
int x.
```

```
int y.
int angle.

CREACION DEL AGUA EN EL MAPA:
char en 3,
a definir segun editor como se va a guardar en el yalm.

CREACION DE UN GUSANO EN EL MAPA
char en 4
int id gusano
int con el numero de jugador que es el dueño
int x
int y
int direccion (-1: izquierda, 1: derecha)
int angle

Es decir, el id de cada gusano esta conformado por un tupla
(jugador_dueño, numero) con el que se va a identificar a ese
gusano en cada uno de los próximos mensajes
La idea es que los gusanos de cada jugador se muestren de un
color distinto.

INICIO DE TURNO
char en 5
int id de jugador actual
int id gusano

POSICIÓN DE UN GUSANO CAMBIA:
char en 6
int id gusano
int x
int y
int direccion (-1: izquierda, 1: derecha)
int angle

CAMBIO DE ESTADO DE UN GUSANO
char en 7
int id gusano
int con numero de nuevo estado
estados: 0 = inactivo, 1 = moviendose, 2 = en el aire, 3 = en el agua,
 4 = muerto.

POSICION DE UN PROYECTIL EN EL MAPA
char en 8
int id de proyectil
int numero de arma: 1=bazooka, 2=mortero, ... continua segun orden en
consigna del tp, 11 = fragmento de proyectil
int x
int y
int angulo

EXPLOCION DE UN PROYECTIL EN EL MAPA
char en 9
int id de proyectil
int x
int y

EL GUSANO ACTUAL SACO UN ARMA
char en 10
int numero de arma: 1=bazooka, 2=mortero, ... continua segun orden en consigna del tp
```

```
EL GUSANO ACTUAL CAMBIA DE ÁNGULO DE APUNTADO
char en 11
int cambio de angulo: 1 = arriba, -1 = abajo

UN GUSANO CAMBIO SU VIDA
char en 12
int id gusano
int nueva vida

UN JUGADOR SE HA DESCONECTADO
char en 13
int numero de jugador que se desconecto

Un JUGADOR HA GANADO LA PARTIDA
char en 14
int numero de jugador ganador
```

2.5. Protocolo Cliente

```
LO QUE MANDA EL CLIENT:
todos los int son mandados en 4 bytes big endian
todas las coordenadas se encuentran en METROS y los angulos en GRADOS.

Las coordenadas refieren al CENTRO del objeto.

Las string se mandan:
int largo del string
el string en si sin el '\0' del final

AVISO DE NOMBRE DEL JUGADOR:
char en 0
int numero de jugador
string con nombre del jugador

AVISO DE MAPA A JUGAR Y CANTIDAD MAXIMA DE JUGADORES:
char en 1
int con map_id
int con cantidad maxima de jugadores
este mensaje debe ser mandado solo por el jugador que reciba el numero
de jugador 1, como se explica en "inicio_de_partida.txt"

El sistema de movimiento del gusano es el siguiente: al inicio de cada
turno todos los jugadores reciben el mensaje "INICIO DE TURNO"(mirar
protocolo_server.txt) en donde son informados del jugador y gusano actual.
 Los mensajes que afectan a los gusanos envian siempre el numero de jugador
  que lo envia, y sera el server quien chequee si el jugador que envia el
  mensaje tiene derecho o no a mover el gusano del turno actual.
MOVER UN GUSANO:
char en 2
int numero de jugador que envia el mensaje
int con direccion: -1= izquierda, 1 = derecha

SALTO DE UN GUSANO
char en 3
int numero de jugador que envia el mensaje

SALTO ATRAS DE UN GUSANO
char en 4
int numero de jugador que envia el mensaje

GUSANO SACA ARMA
```

```
char en 5
int numero de jugador
int numero de arma: 1=bazooka, 2=mortero, etc. segun consigna del tp

GUSANO CAMBIO ANGULO DE APUNTADO
char en 6
int numero de jugador
int cambio de angulo: 1= arriba, -1=abajo

GUSANO CAMBIA TIEMPO DE CUENTA REGRESIVA DE ARMA:
char en 7
int numero de jugador
int nueva cuenta regresiva: 1,2,3,4,5

GUSANO CARGA BARRA DE PODER:
char en 8
int numero de jugador

PARAR DE CARGA BARRA DE PODER Y DISPARAR:
char en 9
int numero de jugador
```