FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

TALLER DE PROGRAMACIÓN I

1º CUATRIMESTRE 2021

Trabajo Práctico Final - Manual de Proyecto

AUTORES

Cai, Ana Maria - #102 150

Giampieri Mutti, Leonardo - #102 358

Vazquez Fernandez, Francisco Manuel - #104 128

Índice

1. Enunciado

El enunciado completo se encuentra en el siguiente archivo adjunto enunciado.pdf.

2. División de tareas

Al igual que el proyecto, se dividió generalizadamente las tareas en tres categorías: servidor, cliente y editor.

El encargado de implementar el servidor fue el alumno Francisco Manuel Vazquez Fernandez. En particular este se encargó de:

- Servidor: algunas tareas a destacar son:
  - Carga de mapas.
  - Lógica de movimiento de los personajes y colisión.
  - Integración completa con Box2D.
  - Lógica de las partidas.
  - Lógica de ataque.
- Documentación.

El encargado de implementar el cliente fue el alumno Leonardo Giampieri Mutti. En particular este se encargó de:

- Protocolo de comunicación y sus respectivas pruebas.
- Servidor:
  - Recepción de clientes.
  - Creación de partidas y unirse a partidas existentes.
  - Comunicación clientes-servidor dentro de una partida.
- Cliente completo: algunas tareas a destacar son:
  - Stencil.
  - Cámara.
  - Mostrar todo el mapa, incluyendo varios tipos de terrenos distintos, objetos y unidades.
  - Interacción por parte del usuario.
- Documentación.

El encargado de implementar el editor fue el alumno Ana Maria Cai. En particular este se encargó de:

- Editor completo.
- Estructura de archivos YAML para la creación de mapas.
- Pantalla de login y de partidas del cliente.
- Documentación.

3. Evolución del proyecto

A continuación se describen las actividades llevadas a cabo cada semana del proyecto para ser completado.

- Semana 1: Comienzo del proyecto
  - Familiarización con herramientas a utilizar.
  - Primeros pasos con box2d.
  - Programa básico con SDL: se muestra una imagen, una animación y esta animación se puede desplazar por la pantalla.
  - Primeros pasos con CMAKE.
- Semana 2
  - Implementación de esquema básico del esqueleto cliente-servidor multicliente.
  - Primeros pasos en la estructura de hilos del cliente.
  - Animaciones con algoritmo "drop and rest"para evitar desincronización.
  - Sistema de comunicación (cliente - servidor): protocolo y sus respectivas pruebas.
  - Programa demostración de movimiento de objetos con box2d. Unimos física de box2d con visuales de sdl
- Semana 3
  - Se completa esquema básico de esqueleto cliente servidor.
  - "Juego aburrido": observamos la comunicación entre varios clientes dentro de una partida.
  - Se familiariza con la herramienta qt y se implementa ventanas básicas y draft de 'drag and drop' básico.
  - Se intenta implementar stencil.
- Semana 4
  - Servidor:
    - Creación de múltiples partidas.
    - Se realiza un avance sobre la lógica del juego. Se agregan las armas y la lógica de disparo
    - Se agrega archivo de configuración.
  - Cliente:
    - Se empiezan a dibujar los mapas a partir del formato de los archivos del editor.

        - Se dibujan elementos visuales del Hud pertinentes al jugador, como la plata, las balas, y los puntos de vida.
        - Inicio de la pantalla de Login.
        - Inclusión de sonidos.
        - Se agrega archivo de configuración.
    - Editor:
        - Se lee y modifica archivo yaml que representan mapas.
        - Se crea una pantalla de inicio.
        - Se implementa 'point and click' para agregar elementos al mapa.

- Semana 5: Entrega primera
    - Servidor:
        - Lógica del juego casi completa.
        - Múltiples partidas checkeadas.
        - Armado de mapas completos a partir de un archivo yaml del editor.
    - Cliente:
        - Se implementa interfaz gráfica básica de LogIn.
    - Editor:
        - Se une compilación de editor con el resto del proyecto
        - Se agrega funcionalidad de scrolling.
        - Se completa 'drag and drop'.

- Semana 6
    - Servidor:
        - Se refactoriza buena parte de la lógica del juego.
        - Se completan los features faltantes.
    - Cliente:
        - Se termina de realizar configurable al stencil.
        - Se arregla el problema del sonido con la distancia.
        - Se agrega una pantalla de scores al final de la partida.
        - Correciones de la primera entrega.
    - Editor:
        - Correcciones primer entrega.
        - Agrega feature de validar existencia de zonas.
        - Se sacan algunas armas del editor que no deberian aparecer como drops en el mapa.

- Creación de instalador con bash y cmake.
- Documentación: manual de usuario.
- Documentación: manual técnico: cliente y servidor.

- Semana 7: Entrega final
  - Se completan detalles sobres features.
  - Se realizan más refactors de la lógica del juego.
  - Mejoras instalador, se deshardcodean todos los paths a archivos y se genera un directorio propio del juego.
  - Documentación: manual de proyecto.
  - Documentación: manual de usuario continuado.
  - Documentación: manual técnico: editor.

4. Inconvenientes encontrados

A continuación se listan un conjunto de inconvenientes encontrados:

- Un inconveniente encontrado fue la integración del 'editor de escenarios' con el resto del proyecto. En un principio, el 'editor' se desarrollaba y compilaba el un archivo 'cmake' aislado al resto del proyecto. A la hora de intentar unir la compilación de todo el proyecto se consumió mucho tiempo intentado resolver varios conflictos de librerías ya que se intentó crear un subdirectorio con todas las clases del editor que requerías qt. Esto no fue posible y se resolvió alternamente agregando los archivos de las clases del 'editor' sin la creación de subdirectorios. Además, hubo conflictos relacionados a rutas de los archivos utilizados.
- Otro inconveniente relacionado al 'editor de escenarios' fue la implementación de un feature complicado como fue 'drag and drop' para una clase de qt distinta a la que fue finalmente utilizada.
- Encontramos también problemas al integrar box2d al proyecto. En principio queríamos generar un compilado y linkearlo con el resto del proyecto pero terminamos forkeando el repositorio de box2d al repositorio del trabajo.
- Un feature complicado fue la transformación de ángulos de sdl a box2d. Al tener el eje y reflejado y medir ángulos clockwise, se tuvo que realizar transformaciones para poder tener un sistema de referencia común entre el cliente y el servidor.
- Al desacoplar al protocolo del servidor y del cliente mediante callbacks, generamos una estructura tangencial a ambos programas pero que sin embargo es funcional. Lo que fue complicado al principio fue entender como sería la estructura, qué métodos se iban a necesitar para la comunicación, y a que objetos pertenecerían los callbacks. Además, al tener tantos mensajes, el protocolo se fue volviendo más complejo y a veces complicado de entender del todo.
- Fue complicado lograr la rotación de los personajes respecto de la posición del mouse. Nuevamente se encontraron problemas con el cambio de base de SDL respecto a la base 'normal'.
- Nos encontramos inconvenientes a la hora de debuggear con el juego ya casi completo. Para algunos bugs, se debía levantar varias partidas, llegar al mismo setting en el cual recordabamos haber visto ese bug y poder reproducirlo. Esto en general nos llevó mucho tiempo.

5. Análisis de puntos pendientes

5.1. Servidor

Si bien se terminaron todos los features, enumeramos algunos puntos que creemos que son mejorables.

- Mandar un mensaje verboso al fallar en el login, sea por haber tratado de crear una partida con un nombre ya existente o por haber tratado de unirse a una partida que ya estaba llena.
- Arreglar bug de movimiento que no permite movilizarse (a veces) en el juego debido a la lógica de la recepción de eventos de movimiento.

5.2. Cliente

Al igual que con el servidor, todos los features pedidos por el enunciado están terminados, pero algunos podrían ser pulidos. Los enunciamos.

- Acomodar algunas texturas que se corren de lugar al cambiar resoluciones.
- Zoomear la cámara ante un cambio de resolución.
- Beautify algunas texturas del hud.

5.3. Editor

Se logró implementar un editor de mapas de niveles o escenarios del juego. A continuación se listan características o features de este último:

- Se utiliza la librería Qt.
- Se valida que se tengan dos zonas de inicio para los equipos, y que cuente con al menos una zona para plantar la bomba.
- Se puede abrir y guardar mapas que se guardan en formato YAML.
- Los mapas más grandes que las pantallas soportan 'scrolling'.
- Se pueden crear mapas desde cero que se guardan en formato YAML.
- Se implementó 'point and click' para colorcar objectos en el mapa.
- Se implementó 'drag and drop' para mover objetos en el mapa.
- Se puede colocar en un mapa paredes, armas, cambiar el fondo y colocar las tres zonas.

5.4. Restricciones

A continuación se listan las restricciones cumplidas a la hora de la implementación del trabajo práctico.

- el trabajo practico se resuelve en POSIX 2008 C++11 utilizando SDL y Qt.
- Se usa una librería de parsing YAML llamada yaml-cpp y no se implementa un parser propio.
- Se logró completar la documentación mínima exigida.
- Se implementó un instalador y desinstalador a base de un script bash.

6. Herramientas

A continuación se listan las principales herramientas utilizadas:

- C++11: lenguaje de programación utilizado para implementar el trabajo práctico.
- SDL2: librerías utilizadas para la implementación de la interfaz gráfica.
- QT5: librerías utilizadas para la implementación de la interfaz gráfica.
- box2d: librería para manejar la física en juegos 2D.
- cmake y make: herramientas utilizadas para la compilación.
- yaml-cpp: librería para parsear archivos en formato YAML
- GitHub: forja para alojar proyectos utilizando el sistema de control de versiones Git.

7. Conclusiones

Como conclusiones, podemos decir los 3 integrantes que aprendimos un montón de ambos stacks, back y frontend, ya que es el trabajo más grande que hicimos hasta el momento y además el más complejo. Nos encontramos dificultades frente a lo grande del trabajo y en cuanto a la organización y la división de tareas, pero se pudo llevar adelante para obtener un producto que cumple con los requerimientos pedidos en el tiempo acordado. Pudimos hacer sesiones de a grupo en donde los 3 programabamos secciones distintas de la aplicación y al mismo tiempo nos íbamos ayudando frente a los inconvenientes que encontrábamos. También, pudimos aprender más sobre el uso de herramientas de control de versiones, como lo fue git, para no pisarnos con los features que cada uno iba desarrollando, y con frameworks que hasta el momento no conociamos como lo era box2d y sdl2.