Taller de programación

Curso Veiga

Manual de proyecto
Duck game

3 de diciembre de 2024

Matias Besmedrisnik 110487

Lourdes Ramirez Almada 105900

Ramirez Ignacio 111167

Ezequiel Aragon 110643

Índice

1. Introducción

Les presentamos el manual de proyecto del juego *DuckGame*, realizado para la materia de Taller de Programación, Cátedra Veiga, de la Facultad de Ingeniería de la Universidad de Buenos Aires. Este manual reflejará cómo abordamos el proyecto: desde los primeros días, con un poco conocimiento sobre las nuevas tecnologías, hasta el desafío de conocernos como equipo y enfrentar el trabajo de la manera más rápida y dinámica posible.

2. Reporte semanal de tareas

2.1. Primera semana (15/10 - 22/10)

En nuestra opinión, esta fue una de las semanas más importantes del proyecto, ya que nos permitió conocernos como equipo y definir una dinámica de trabajo que resultó esencial para el desarrollo del mismo.

Durante el transcurso de esta semana, nos dedicamos a comprender en profundidad el enunciado, identificando los requisitos principales y estableciendo metas tanto a corto como a largo plazo. Además, comenzamos a familiarizarnos con las nuevas herramientas y su instalación, como `SDL` y `CMake`. Estas actividades fueron fundamentales para encontrar un punto de partida para el proyecto.

Integrante	Tareas realizadas
Matias	Planteo de protocolo y diseño base de comunicación entre servidor y cliente.
Lourdes	Búsqueda de recursos (Sprites). Prueba de concepto de QT.
Nacho	Creo todas las texturas principales e Implemento una demo de la sdl con el movimiento de los patos
Ezequiel	Planteo de protocolo y diseño base de comunicación entre servidor y cliente.

Cuadro 1: Tareas realizadas por los integrantes en la Semana 1.

2.2. Segunda semana (22/10 - 29/10)

Integrante	Tareas realizadas
Matias	Movimientos de pato, saltos y aleteo, del lado del servidor. Implementaciones primitivas del disparo y su lógica. Implementación de colas e hilos del lado del cliente.
Lourdes	Editor de níveles: Lógica de fondos y plataformas.
Nacho	Ya puedo enviar y recibir comandos del server y dejo por default un arma y armadura al pato
Ezequiel	Planteo de armamento básico y su lógica. Desplazamiento de las balas.

Cuadro 2: Tareas realizadas por los integrantes en la Semana 2.

Integrante	Tareas realizadas
Matias	Lectura de archivo .YAML para los mapas, implementación de boxes y colisiones del pato. Lógica de agregar color a los patos y doble jugador en un mismo cliente.
Lourdes	Editor de níveles: Lógica del resto de elementos. Implementación de guardar archivos mapas en YAML.
Nacho	Ya dibujo las balas cuando se dispara y el handler ya esta casi completo, comienzo con el zoom
Ezequiel	Lógica de armas de dificultad media, dispersión y retroceso. Desplazamiento de múltiples balas.

Cuadro 3: Tareas realizadas por los integrantes en la Semana 3.

2.3. Tercera semana (29/10 - 05/11)

2.4. Cuarta semana (05/11 - 12/11)

Integrante	Tareas realizadas
Matias	Implementación de PickUp, LeaveGun, factory weapons y escudos. Implementación del LobbyPartidas, donde se pueden jugar múltiples partidas en el mismo servidor y multicliente. Cambios en el protocolo de comunicación.
Lourdes	Menu: Se agrega la parte visual y lógica de conexión con el cliente. Editor: Se agrega un menú y la posibilidad de editar un mapa.
Nacho	Agrego movilidad los patos fluida al hacer cualquier movimiento, se actualizan constantemente los objetos con un redenderer, sigo avanzando con el zoom
Ezequiel	Lógica de armas de alta dificultad, ráfaga de disparos y primeras colisiones.

Cuadro 4: Tareas realizadas por los integrantes en la Semana 4.

2.5. Quinta semana (12/11 - 19/11)

Integrante	Tareas realizadas
Matias	Arreglo de colisiones, cierre de conexiones, Manual de Usuario, lectura de configuración desde config.yaml.
Lourdes	Editor: Se agrega un YAML de configuración. Instalador.
Nacho	Implento el sonido a las armas, la musica de fondo y comienzo con los tests.
Ezequiel	Perfeccionamiento de colisiones, lógica de cajas y sus recompensas. Ejecución de resbalada de banana y explosión de granada.

Cuadro 5: Tareas realizadas por los integrantes en la Semana 5.

2.6. Sexta semana (19/11 - 26/11)

Integrante	Tareas realizadas
Matias	Continuo con coalisiones, infromes, manuales, chequeos de ingreso del menu y actualizacion del protocolo
Lourdes	Instalador: El juego se puede correr desde cualquier ubicación. Editor de níveles: Se pueden eliminar objetos y aparecen donde se los llamo.
Nacho	Implemento las scenas de end of round, victory, colores de los patos. Arreglo el cierre del lado del cliente, ya dejo andando por completo el zoom y termino los tests.
Ezequiel	Corrección de colisiones y recompensas de las cajas. Desarrollo de documentación.

Cuadro 6: Tareas realizadas por los integrantes en la Semana 6.

2.7. Séptima semana (26/11 - 03/12)

Integrante	Tareas realizadas
Matias	Arreglo conexiones de forma definitiva, rediseño de pantallas, nuevas animaciones para disparo y muerte. Arreglo el instaler.
Lourdes	Menu: Funcionalidad del boton Quit e implementación de otras correciones. Instalador.
Nacho	Cambio por completo el manejo de las texturas, ya que primero se descargan todas y luego se usan por referencia, arreglo los cierres de las correcciones que nos dieron y con esto bajo el consumo de 60 porciento a 4 porciento del CPU.
Ezequiel	Corrección de colisiones. Implementación de colisión de balas con pato acostado. Implementación para soltar armadura y casco. Desarrollo de documentación.

Cuadro 7: Tareas realizadas por los integrantes en la Semana 7.

3. Herramientas

3.1. IDEs

Para realizar el proyecto, utilizamos diferentes entornos de desarrollo integrados (IDEs) según las preferencias y posibilidades de cada integrante. En general, `CLion` fue la herramienta principal, especialmente en los momentos en que necesitábamos trabajar de manera simultanea a través de `CodeWithMe`, y fue la opción preferida por la mayoría de los integrantes. Complementada con `GDB`, nos permitió depurar el código de forma fácil y cómoda. Además, algunos integrantes utilizaron `Visual Studio Code`, una IDE más liviana, que se adaptó bien a tareas específicas del proyecto. Por otro lado, `QtCreator` fue utilizado en particular para el diseño del lobby y el editor, aprovechando sus capacidades para trabajar con `Qt`.

3.2. Linters

Para un codigo de mejor calidad, utilizamos diferentes linters que vinimos utilizando a lo largo de la cursada. Estas herramientas nos ayudaron a mejorar la legibilidad y prevenir errores comunes.

- **cppcheck**: Utilizado para detectar errores potenciales.

- **cpplint**: Verificar que el codigo cumpla con las convenciones de estilo de `C++`.
- **clang-format**: Para asegurar un estilo uniforme en todo el proyecto.

4. Documentación

4.1. Sockets y threads

Es importante mencionar que este trabajo práctico se desarrolló con conocimientos previos, como el uso de **Threads**, **Sockets** y otras herramientas vistas a lo largo de la materia. Para repasar estos temas puede ser util los siguientes links:

- Hands-on-sockets
- Hands-on-threads

4.2. SDL Y QT

A lo largo de este proyecto, nos enfrentamos a diferentes situaciones que requirieron aprender nuevas tecnologías.

Para aprender y trabajar con **SDL** y **Qt**, utilizamos material de clases anteriores, especialmente ejemplos prácticos realizados durante la pandemia. Además, complementamos este conocimiento con la documentación oficial y tutoriales como:

- Tutoriales de Lazy Foo: Una excelente referencia paso a paso para aprender SDL desde conceptos básicos hasta avanzados.
- Tutoriales de la catedra: Tutorial de SDL del curso.
- QT: Documentacion oficial de QT.

4.3. CMake

Para trabajar con **CMake**, recurrimos a los siguiente repositorio propuesto por la catedra:

- Proyect template: Base de como podría estar organizado el proyecto.
- Tutorial CMake: Tutorial del dipa de CMake.

5. Feedback

A nuestro grupo le parecieron muy interesantes los contenidos de las clases, y valoramos la claridad con la que se explicaron los temas. Quizás nos habría gustado que se dedicara un poco más de tiempo a explorar el uso de SDL y Qt con ejemplos prácticos. También creemos que hubiera sido útil contar con una estructura básica de un juego como referencia para guiarnos en nuestro proyecto.