

Duck Game

Manual del Proyecto

Integrantes	1
División de tareas	1
Herramientas	1
Puntos problemáticos	1
Objetivos alcanzados	1
Mejoras	2
Known Issues	2

Integrantes

Calvert de Bohun, Lucia	109441
Gerez, Facundo	109429
Osan, Martin	109179
Piccin, Candela	109760

División de tareas

La división de tareas no representa tan linealmente lo que hizo cada uno ya que sobre el final del trabajo cada uno se encargó de solucionar o implementar funcionalidades, detalles o corregir bugs en distintas partes del proyecto.

Calvert	Desarrollo en SDL: ventana del juego: construcción del motor gráfico; texturas, animaciones y efectos especiales Desarrollo en la lógica del juego: armas y armaduras. Corrección de bugs e integración con el cliente Armado de manual de usuario
Gerez	Desarrollo del protocolo. Desarrollo del monitor de lobbies, estructura de lobby e integración con el cliente en el servidor. Modo test del cliente. Desarrollo en SDL: zoom automático, construcción del motor de sonido, eventhandler.
Osan	Desarrollo total de la lógica del juego. Modo test y modo cheat del servidor.
Piccin	Desarrollo inicial de la conexión servidor-cliente y desarrollo del protocolo Desarrollo en Qt: menú principal y editor de niveles. Configuración de la compilación con cmake. Documentación y configuración del script de instalación.

Herramientas

El trabajo fue realizado en el lenguaje C++ (C++20) con el estándar POSIX 2008.

Para el desarrollo del juego se utilizaron las siguientes librerías:

- SDL2 y las extensiones SDL2_image (v2.6.3), SDL2_mixer (v2.6.3) y SDL2_ttf (v2.20.2) para el desarrollo de la ventana del juego, que incluye manejo de texturas, sonidos y texto.
- Qt (versión 5) para el desarrollo de la ventana del menú del juego y del editor de niveles.

- GTest (versión 1.15.2) para la generación de test de protocolo y lógica del juego.
- Shell scripting para los scripts de instalación y ejecución del programa.

Las herramientas de desarrollo utilizadas para debugging fueron GDB y para el manejo de leaks y errores de memoria, Valgrind. Utilizamos el IDE Visual Studio Code y el editor de texto Neovim.

Puntos problemáticos

- Concurrencia entre partidas
- Desarrollo inicial del protocolo
- Desarrollo del zoom

Objetivos alcanzados

Pudimos cumplir con que el juego soporta 1 o 2 jugadores en la misma máquina, además de poder jugar partidas de hasta 4 jugadores (el máximo de jugadores podría cambiar, no es necesariamente fijo). Se cumple con que cada jugador puede realizar todas las acciones posibles. Se pudieron implementar todas las armas y armaduras pedidas, y cajas que pueden ser destruidas con disparos, todo con sus respectivas animaciones y efectos de sonido. Y se cumple con poder jugar una partida que termina correctamente con un ganador.

Se cuenta con un zoom dinámico que muestra a todos los jugadores en todo momento haciendo un zoom in y zoom out automático.

Para el cliente se desarrollo un modo test y para el servidor un modo test y cheat (explicados en la Documentación técnica) y el editor de niveles.

Consideramos que se pudo cumplir con todos los requisitos de la consigna.

Mejoras

- Idear un protocolo menos rígido
- Agregar que el jugador pueda elegir su color de pato en el lobby.

Known Issues

- No se avisa cuando un jugador se desconecta de la partida, el juego sigue con un jugador menos.