

Duck Game

Índice

[Introducción](#)

[Descripción](#)

[Controles](#)

[Movimientos y acciones](#)

[Armas](#)

[Granada](#)

[Banana](#)

[Pew-Pew Laser](#)

[Laser Rifle](#)

[AK-47](#)

[Pistola de Duelos](#)

[Pistola Cowboy](#)

[Magnum](#)

[Escopeta](#)

[Sniper](#)

[Armaduras y Cascos](#)

[Escenario](#)

[Cámara](#)

[Animaciones](#)

[Identificación del jugador](#)

[Sonidos](#)

[Musica ambiente](#)

[Configuración](#)

[Cheat](#)

[Editor de niveles \(para grupos de 4 integrantes\)](#)

[Condición de victoria](#)

[Lobby](#)

[Entregables](#)

[Aplicaciones Requeridas](#)

[Unit tests](#)

[Restricciones](#)

[Referencias](#)

Introducción

En este TP vamos a recrear una versión simplificada del Duck Game, creado por Landon Podbielski en 2014 [1].

Descripción

Controles

El cliente gráfico deberá soportar 1 o 2 jugadores en la misma máquina, controlando los patos por vía del teclado.

Estas son las posibles acciones:

- Moverse a la izquierda
- Moverse a la derecha
- Salta/aletear
- Tomar/soltar un arma/caso/armadura
- Disparar
- Apuntar hacia arriba
- Tirarse al piso

Movimientos y acciones

Los patos pueden moverse libremente por el escenario. También pueden saltar y estando en el aire, pueden aletear y caer más despacio si se presiona repetidamente el botón de salto.

Los patos no sufren daño por caídas salvo si caen fuera del escenario que en cuyo caso, mueren.

Los patos pueden disparar en la dirección en la que están mirando; pueden disparar hacia arriba si apuntan hacia allí. La dirección precisa de los proyectiles depende del tipo de arma.

Inicialmente los patos no tienen armas/armaduras/cascos y deben tomarlas de las que estén en el escenario. Si lo desean, pueden soltarla también.

También pueden tirarse al piso, un movimiento peculiar, para simular estar muertos o tal vez tratar de esquivar algún disparo.

Armas

Granada

Mecánica: Cuando el jugador presiona el botón de disparo, le está sacando el seguro a la granada. A partir de ahí, tiene 4 segundos antes de que explote. Presionando luego el botón para tirar el arma, el pato lanza la granada.

Al explotar, lanza fragmentos con un alcance de 5 tiles.

Alcance: medio

Municiones: 1 granada

Banana

Mecánica: funciona igual que la Granada solo que no hay ninguna explosión. Cuando se lanza la banana esta queda en el piso esperando a que un pato la pise, se resbale, caiga y salga patinando por el suelo. Ideal ponerlo en lugares cerca de precipicios!

Alcance: medio

Municiones: 1 banana

Pew-Pew Laser

Mecánica: al presionar el botón de disparo, la Pew-Pew Laser dispara una rafaga de 3 rayos, todos con la misma dirección principal aunque con algo de dispersión (Si el pato dispara horizontalmente, los 3 rayos salen horizontalmente pero uno va un poco más inclinado hacia arriba y otro más inclinado hacia abajo). Para hacer múltiples disparos, el jugador debe presionar repetidamente el botón de disparo. Aun así, todos los rayos siguen una línea recta.

Municiones: 12 rafagas

Alcance: 35 a 38 tiles

Laser Rifle

Mecánica: al presionar el botón de disparo, el Laser Rifle dispara un rayo láser; manteniendo presionado el botón, continuar disparando rayos. Los rayos salen con una inclinación de 45 grados hacia abajo; múltiples disparos consecutivos hacen que tenga algo de variación esta inclinación.

Los rayos rebotan varias veces sobre superficies hasta impactar en un pato o desaparecer.

Municiones: 10 rayos

Alcance: 30 tiles

AK-47

Mecánica: igual que el Láser Rifle, el jugador puede realizar múltiples disparos manteniendo el botón de disparo. Las balas salen en línea recta pero con una dispersión (variación en el eje ortogonal al disparo) considerable, incrementando si se mantiene el botón de disparo presionado (o sea, se pierde precisión). El disparo de la AK-47 genera un retroceso que hace que el pato se mueva un poco en la dirección opuesta al disparo.

Municiones: 30 balas

Alcance: 13 tiles

Pistola de Duelos

Mecánica: arma de un solo tiro, con una dispersión (variación en el eje ortogonal al disparo) considerable.

Municiones: 1 bala

Alcance: 5 tiles

Pistola Cowboy

Mecánica: se requiere presionar el botón de disparo por cada tiro; arma sin dispersión, ni retroceso, ni tiempo de enfriamiento o recarga.

Municiones: 6 balas

Alcance: 20 tiles

Magnum

Mecánica: igual que la Pistola Cowboy pero con un poco de dispersión y retroceso.

Municiones: 6 balas

Alcance: 20 tiles

Escopeta

Mecánica: se requiere presionar el botón de disparo por cada tiro y luego de un tiro hay que presionar una segunda vez para recargar el arma.

En cada disparo se lanzan 6 perdigones con una gran dispersión.

Municiones: 2 disparos

Alcance: 7 a 9 tiles

Sniper

Mecánica: se requiere presionar el botón de disparo por cada tiro; arma sin dispersión, ni retroceso. Tiene un tiempo de recarga corto.

Municiones: 3 balas

Alcance: 64 tiles

Armaduras y Cascos

Mecánica: se requiere presionar el botón de disparo ponersela. Una vez puesta, una armadura o casco puede absorber 1 fragmento/bala/rayo sin producirle daño al pato aunque desapareciendo la armadura/casco en el proceso.

Un pato puede tener hasta 1 armadura y 1 casco a la vez.

—

Vease [5] y [6].

Escenario

Son una serie de plataformas sobre las cuales los patos podrán caminar y/o saltar. A diferencia del juego original no habrá puertas ni piscinas de agua o fuego.

Lo que sí habrá serán los spawn places: lugares en donde aparecen armas/armaduras/cascos al azar (que dependerá del escenario a jugar). Cuando un arma es tomada y el spawn place queda libre, habrá un tiempo mínimo más un tiempo al azar para que este vuelva a hacer aparecer un arma.

En los escenarios habrá cajas que podrán ser destruidas con disparos. Algunas cajas no tendrán nada, otras tendrán armas/armaduras/cascos y otras explosivos que explotarán como si fuesen una granada, todo elegido al azar.

Las cajas no reaparecen.

Cámara

La cámara de todos los jugadores debe mostrar a todos los patos vivos en **todo** momento. Dado que en algunos momentos los patos estarán más cerca o más lejos entre ellos, la cámara deberá hacer un zoom in o zoom out automático.

Animaciones

El juego no debe mostrar imágenes estáticas sino pequeñas animaciones para darle mayor realismo:

- Los personajes tienen una animación para correr, saltar, disparar...
- El movimiento de los proyectiles
- Las explosiones.

Identificación del jugador

En los sprites disponibles [2] hay solo 2 colores de pato, sin embargo es requerido que cada jugador tenga un color distinto.

Se pueden modificar los sprites manualmente o en runtime.

Sonidos

Como todo juego se debe reproducir sonidos para darle realismo a los eventos y acciones que suceden:

- Cuando hay disparos.
- Cuando hay una explosión.

Si la cantidad de eventos que suceden es muy grande, algunos sonidos pueden ser evitados para no saturar al jugador con tanta información.

Música ambiente

El juego debe reproducir una música ambiente [4], con un volumen relativamente bajo.

Configuración

Todos los atributos en este enunciado son a modo de ejemplo y deben poderse cambiar vía un archivo de configuración YAML.

Por ejemplo se debe poder cambiar el daño de un arma; el radio de una explosión; etc.

La idea es que no haya valores hardcodeados en el código: el archivo de configuración te permitirá ir

tuneando el juego para hacerlo más fácil o difícil y le permitirá al docente probar rápidamente una funcionalidad sin tener que jugar mucho tiempo.

Cheat

El juego debe tener cheats (trucos) que permitan **probar más fácilmente el trabajo**. Cuales y como ejecutarlos están definido por ustedes (se recomienda ser pragmáticos y buscar la facilidad)

Editor de niveles (para grupos de 4 integrantes)

Se deberá implementar un editor de niveles (tanto crear nuevos como editar preexistentes). Se deberá poder posicionar los spawns (puntos de aparición) de los patos, las armas/armaduras/cascos y cajas así como también de los pisos y paredes que componen el nivel.

Condición de victoria

En una ronda se elige un mapa al azar y gana el último pato que siga con vida. Tras 5 rondas se debe mostrar un marcador que liste los jugadores y la cantidad de rondas ganadas totales.

Una partida es una serie grupos de 5 rondas: tras 5 rondas se evalúa si al menos 1 pato logró tener 10 o más victorias. Si es así, la partida finaliza siendo el pato con mayor rondas ganadas el pato ganador.

Si tras 5 rondas nadie logró llegar a las 10 victorias se vuelven a jugar otras 5 rondas.

Si para el puesto de ganador de la partida hay empate, se adicionan 5 rondas más hasta que pueda declararse un único ganador.

Habiéndose declarado ganador, se debe mostrar una pantalla de victoria.

Lobby

Los jugadores podrán optar por crear una nueva partida o unirse a una pre-existente.

Al crearse una partida, el jugador podrá elegir cuándo darle comienzo a esta. Jugadores que no se hayan sumado hasta ese momento no podrán hacerlo.

Al momento de crear o de unirse a una partida, se deberá indicar si en el cliente gráfico se usará por 1 o por 2 jugadores.

Antes de comenzar la partida se debe indicar claramente qué color de pato le fue asignado a cada jugador. Dicha asignación debe mantenerse a lo largo de todas las rondas de una misma partida.

Entregables

- Documentación (véase los detalles en la [página web oficial de la materia](#)):
 - Manual del usuario
 - Documentación técnica

- Manual de proyecto
- Instalador (sea un makefile, cmake o un script the shell) que debe:
 - Descargar las dependencias del proyecto e instalarlas (por ejemplo SDL)
 - Compilar el proyecto
 - Correr los test unitarios
 - Copiar/mover binarios a /usr/bin, los assets (como imágenes, audios) a /var/NAME/ y los archivos de configuración a /etc/NAME/, donde NAME es el nombre del TP que el grupo decida darle.
Nota: dentro de /var/NAME/ o de /etc/NAME/ pueden haber tantas subcarpetas como lo necesiten/deseen.
- Repositorio en Github con el código, el instalador, documentación y todo el material necesario para la compilación y ejecución.

Aplicaciones Requeridas

El proyecto consta de 3 aplicaciones:

- Cliente gráfico
- Servidor
- Editor gráfico

El editor no tiene por qué conectarse al servidor: los niveles editados pueden ser manualmente movidos al servidor. No así el cliente que debe descargarse del servidor el/los niveles a jugar.

El proyecto debe poderse compilar y ejecutar en un Ubuntu 20.04 Focal Fossa (o en su versión más ligera Xubuntu 20.04).

Este es el mismo entorno que fue usado durante la cursada por el Sercom.

Unit tests

El protocolo de comunicación debe estar testeado con GoogleTests; los tests pueden mockear los sockets o no.

Restricciones

La siguiente es una lista de restricciones técnicas exigidas por el cliente:

1. El sistema se debe realizar en C++ (C++20) con el estándar POSIX 2008 utilizando librerías SDL y Qt y se permite el uso de QtCreator.
2. El protocolo de comunicación debe ser binario.
3. Los archivos de configuración deben ser en YAML.
4. Todo socket utilizado en este TP debe ser bloqueante (es el comportamiento por defecto).

Referencias

- [1] https://en.wikipedia.org/wiki/Duck_Game
- [2] Sprites: https://www.sprites-resource.com/pc_computer/duckgame/
- [3] Sonidos (tendran que ser algo creativos aqui): <https://www.youtube.com/watch?v=EA4h8l2zZ1g>
- [4] Musica:
https://www.youtube.com/watch?v=Dj4pEg9S_Pk&list=PLfMR6-vRLLbKe_rXu6Ad25_x9DKz9dpla&index=1
- [5] Armas: <https://duckgame.fandom.com/wiki/Guns>
- [6] Armaduras y cascos: <https://duckgame.fandom.com/wiki/Equipment>