



TALLER DE PROGRAMACIÓN  
CURSO VEIGA

# Manual de usuario Duck game



19 de noviembre de 2024

Matias Besmedrisnik 110487

Lourdes Ramirez Almada 105900

Ramirez Ignacio 111167

Ezequiel Aragon 110643



# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Requisitos del sistema</b>	<b>3</b>
<b>3. Instalación</b>	<b>3</b>
<b>4. Duck Game</b>	<b>3</b>
4.1. Ejecucion . . . . .	3
4.2. Crear/unirse partida . . . . .	4
4.3. Crear partida . . . . .	4
4.4. Unirse a partida . . . . .	6
4.5. Colores . . . . .	7
4.6. Controles . . . . .	8
4.7. Respawn points . . . . .	9
4.8. End of round/Victory . . . . .	9
<b>5. Configuración</b>	<b>11</b>
<b>6. Editor</b>	<b>12</b>
6.1. Ejecución . . . . .	12
6.2. Funcionamiento . . . . .	12
<b>7. Cheats</b>	<b>13</b>

## 1. Introducción

Les presentamos el manual de usuario 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 de usuario ayudará a instalar y entender todo lo que sea necesario para poder disfrutar de la mejor experiencia de juego.

## 2. Requisitos del sistema

Para ejecutar el juego correctamente, el sistema debe cumplir con los siguientes requisitos:

- **Sistema Operativo:** Ubuntu 20.04 Focal Fossa o su versión ligera (Xubuntu 20.04).
- **Dependencias:**
  - **yaml-cpp:** Biblioteca para manejar archivos YAML.
  - **SDL2PP:** Biblioteca para SDL2.
  - **Qt Framework:** Framework para desarrollar interfaces gráficas y manejar eventos.
  - **Google Test (gtest):** Biblioteca para la implementación y ejecución de pruebas, utilizadas para el test del protocolo.

## 3. Instalación

Para la instalación deberás seguir los siguientes pasos:

- **Clonar el repositorio:**

```
$ git clone <URL_DEL_REPOSITORIO>
$ cd <NOMBRE_DEL_REPOSITORIO>
```

- **Descargar las dependencias:** Esto nos permitirá instalar las dependencias necesarias para el correcto funcionamiento del juego.

```
$ chmod +x download.sh
$ ./download.sh
```

- **Utilizar el instalador:** El instalador nos permitirá compilar el proyecto, correr los tests y mover archivos necesarios.

```
$ chmod +x install.sh
$ ./install.sh
```

## 4. Duck Game

### 4.1. Ejecución

Si ya te encuentras en el directorio build, puedes pasar directamente al siguiente paso. De lo contrario, deberás ingresar:

```
$ cd build
```

Primero se debe ejecutar el servidor para esto es necesario abrir una terminal e ingresar lo siguiente:

```
$ ./taller_server <Puerto>
```

Y luego iniciamos el cliente en otra terminal con la siguiente entrada:

```
$ ./taller_client localhost <Puerto>
```

Por ejemplo, si queremos utilizar el puerto 8080 se debera hacer lo siguiente:

```
$ ./taller_server 8080
```

Y luego desde otra terminal:

```
$ ./taller_client localhost <Puerto>
```

## 4.2. Crear/unirse partida

Una vez que se haya ejecutado tanto el servidor como el cliente le propondra si crear o unirse a una partida.

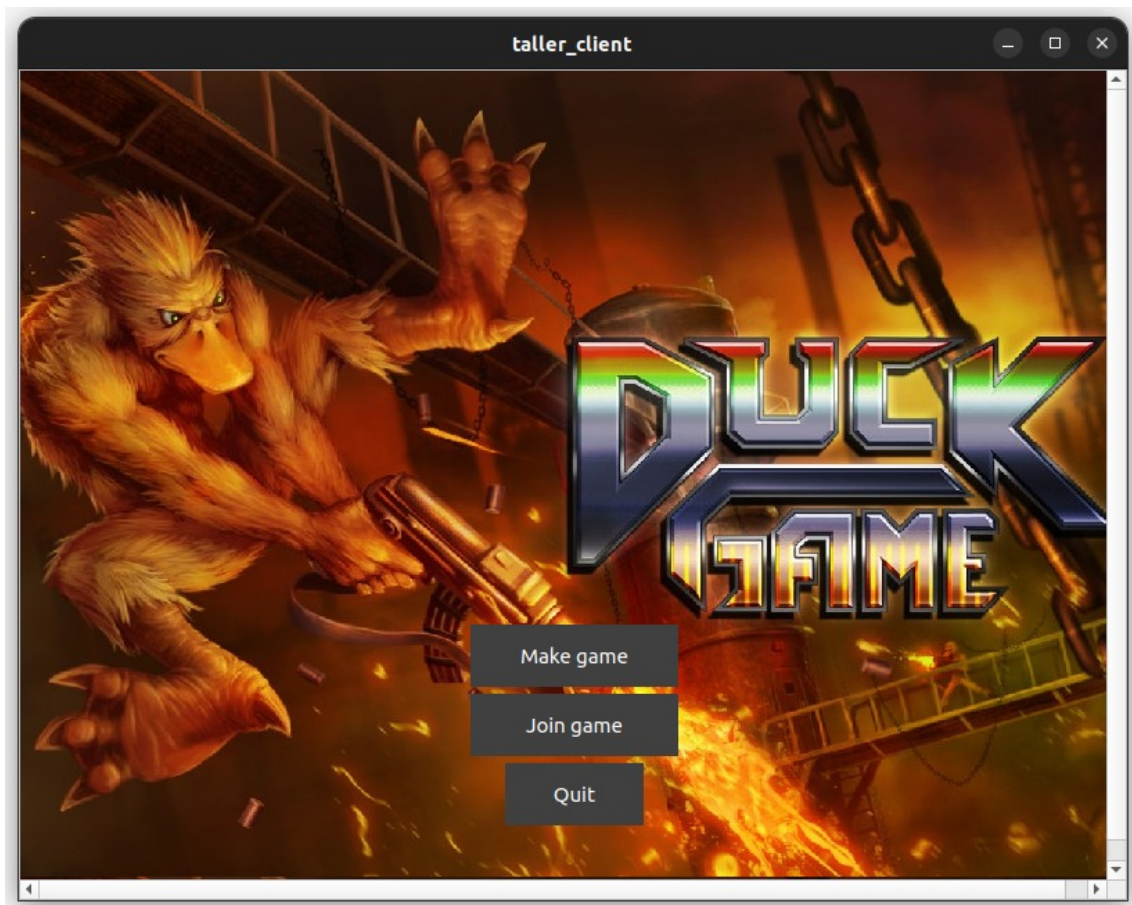


Figura 1: La portada del juego.

## 4.3. Crear partida

En el caso de crear una partida, tendrás la posibilidad de elegir si jugar con uno o dos jugadores. Esta selección se realiza presionando el botón correspondiente: *Single Player* o *Two Players*. En

ambos casos, se te pedirá un nombre para la partida y el nombre de cada uno de los jugadores. El nombre de la partida elegida será el que vean todos los usuarios. .

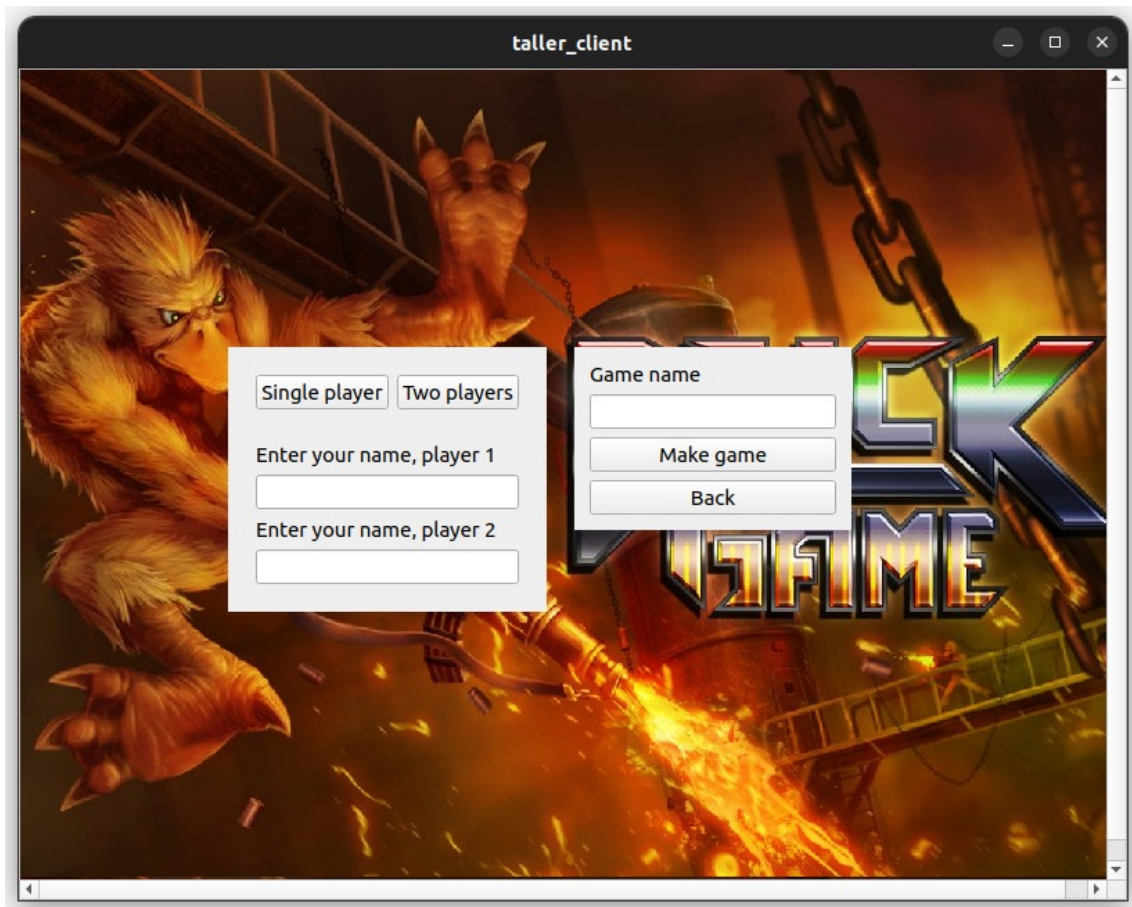


Figura 2: Creación de un juego

Después de definir estos datos, se deberá crear la partida. Los jugadores serán enviados a un lobby hasta que se desee empezar la partida (presionando el botón de *Start*). Mientras permanezcas en este lobby, otros jugadores podrán unirse a la partida.

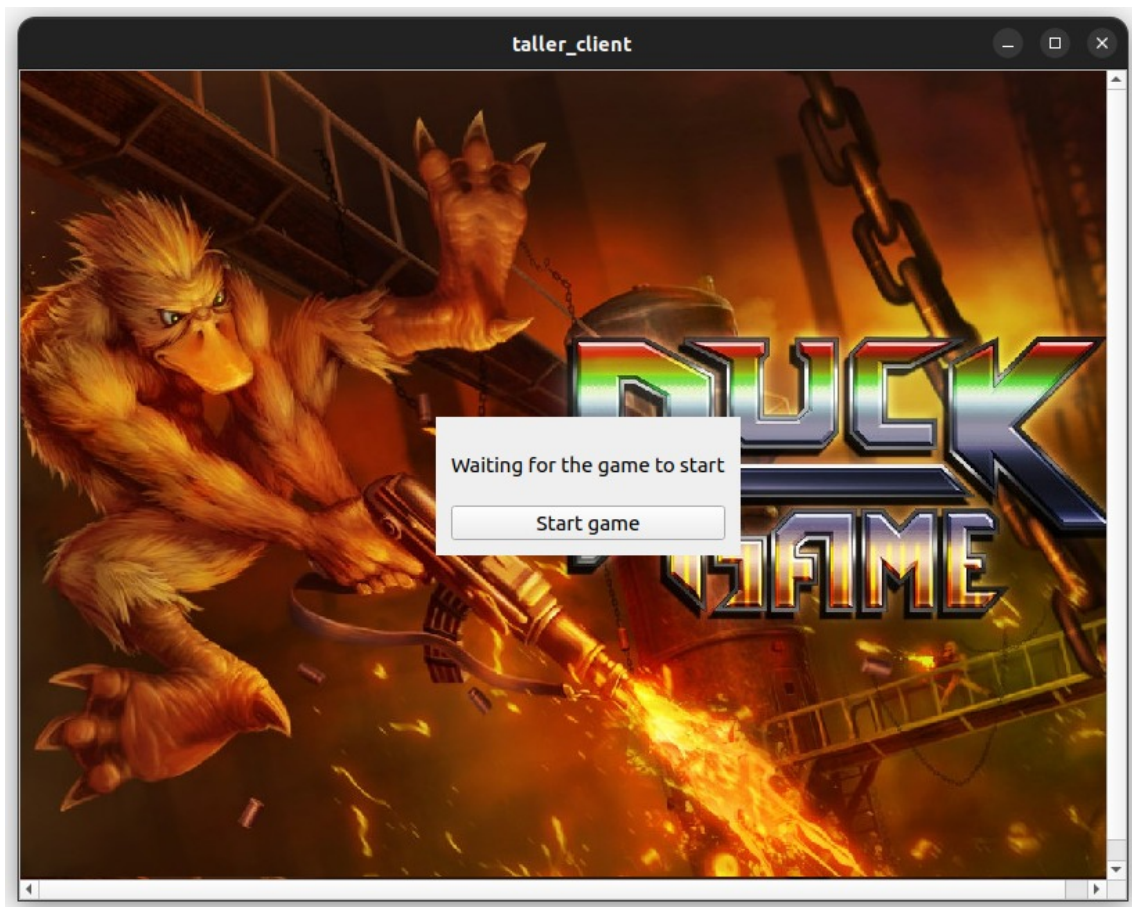


Figura 3: La espera hasta el comienzo de la partida

#### 4.4. Unirse a partida

Al igual que en la creación de una partida, se deberán ingresar los jugadores. Además, habrá un botón "Listar partidas" que mostrará, en un drop-down button, las partidas disponibles para unirse en ese momento. Se deberá presionar el botón "Listar partidas" la primera vez que se acceda a esta ventana. Luego, se debe seleccionar la partida deseada presionando el botón *Join game*.

En caso de que la partida se encuentre activa, el jugador se mantendrá en espera hasta el comienzo de esta.



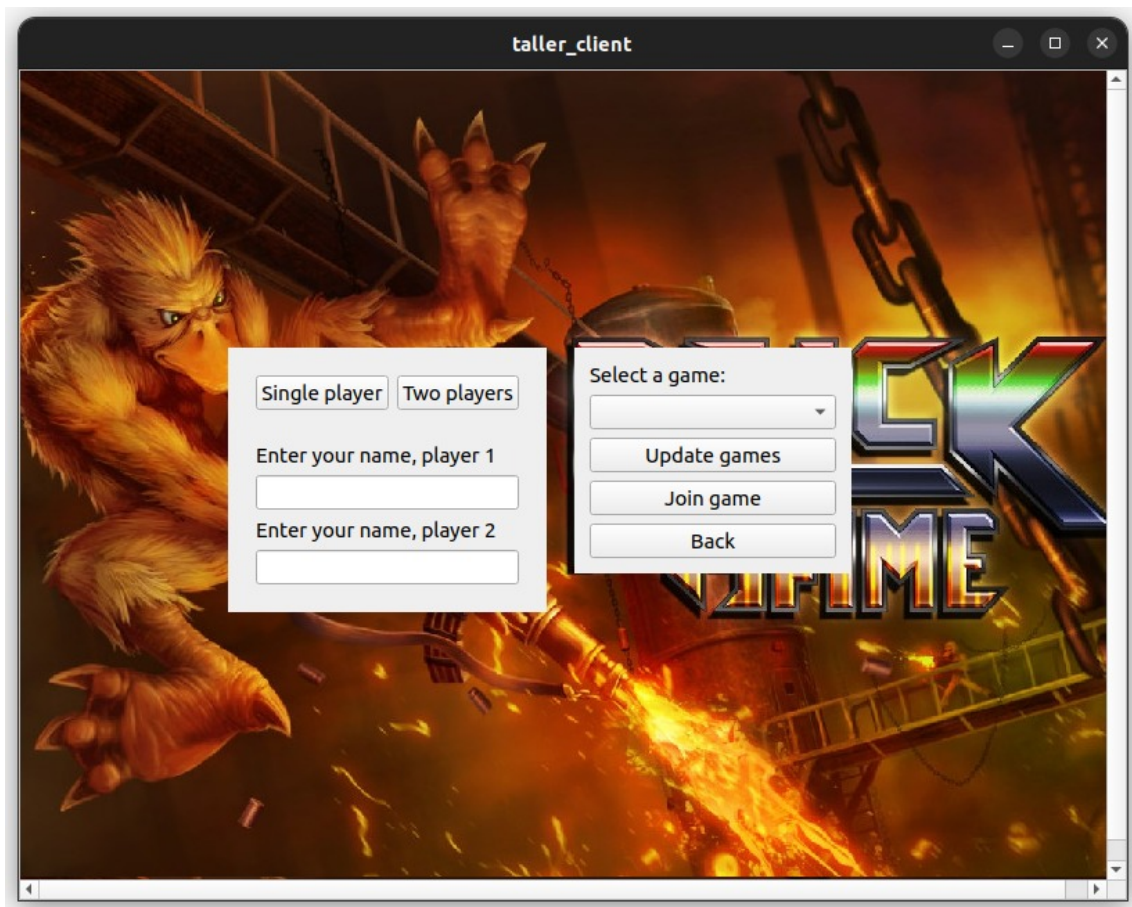


Figura 4: Join game lobby

#### 4.5. Colores

Previo al comienzo de la partida se sera notificado el color de cada uno de los jugadores.



Figura 5: Notificación de colores

#### 4.6. Controles

En el caso de jugar con un jugador o siendo el primer jugador ingresado en una partida de dos jugadores:

- **W**: Mover hacia arriba (arriba en el eje Y).
- **A**: Mover hacia la izquierda (izquierda en el eje X).
- **S**: Mover hacia abajo (abajo en el eje Y).
- **D**: Mover hacia la derecha (derecha en el eje X).
- **SHIFT IZQ**: Disparar
- **E**: Pick up (te permitirá agarrar un arma, escudo o casco)
- **Q**: Leave (te permitirá soltar un arma, escudo o casco)
- **Espacio**: Apuntar hacia arriba con el arma.

En el caso de contar con dos jugadores, al segundo se le otorgarán los siguientes controles(En caso de ser un solo jugador tambien podras acceder a este set de controles):

- **I**: Mover hacia arriba (arriba en el eje Y).



- **J:** Mover hacia la izquierda (izquierda en el eje X).
- **K:** Mover hacia abajo (abajo en el eje Y).
- **L:** Mover hacia la derecha (derecha en el eje X).
- **SHIFT DER:** Disparar
- **U:** Pick up (te permitirá agarrar un arma, escudo o casco)
- **H:** Leave (te permitirá soltar un arma, escudo o casco)
- **M:** Apuntar hacia arriba con el arma.

Al apretar continuamente el salto se podrá aleetar, ralentizando la caída del pato.

#### 4.7. Respawn points

Los *respawn points* son zonas donde las armas, protecciones, cajas y personajes aparecerán. En el caso de las cajas y personajes, será una única aparición por ronda. En cambio, las armas y protecciones reaparecerán de manera aleatoria una vez que hayan sido recogidas. Estos *respawn points* son específicos para cada arma; es decir, siempre aparecerá una del tipo definido.

#### 4.8. End of round/Victory

Una ronda se gana siendo el último jugador vivo. Cada cinco rondas se analizará si algún jugador ha ganado diez rondas. En caso de que esto ocurra, ese jugador será declarado ganador.

Si dos jugadores terminan victoriosos con la misma cantidad de rondas ganadas, se jugará una serie adicional de cinco rondas para determinar el ganador.

La pantalla de *end of round* mostrará el nombre del jugador y la cantidad de rondas ganadas que lleva. Después de algunos segundos, retornará a la partida.



Figura 6: Pantalla end of round

Por otro lado, si un jugador cumple con la condición de victoria, se le mostrará a todos los jugadores el ganador de esa partida, finalizando el juego.



Figura 7: Pantalla victoria

## 5. Configuración

El juego está preestablecido con una configuración inicial que puede ser modificada accediendo al archivo en el siguiente path:

```
server/configuracion/confi.yaml
```

En este archivo YAML, se pueden modificar varios atributos, algunos de estos son:

- **Cantidad de municiones:** Configura la cantidad de municiones iniciales para cada arma.
- **Rango:** Ajusta el alcance de las armas en el juego.
- **Daño:** Modifica el daño provocado por cada arma.
- **Recoil (retroceso):** Controla la cantidad de retroceso que experimenta el arma al disparar.
- **Velocidad del personaje:** Modifica la velocidad de movimiento del jugador.
- **Vida del personaje:** Ajusta la cantidad de vida inicial de cada uno de los patos.

Esto permite personalizar la experiencia facilmente segun sea necesario, evitando depender de una unica forma de juego.

## 6. Editor

El juego comienza con mapas preestablecidos, pero existe la posibilidad de personalizar al máximo los mapas que desees usar.

### 6.1. Ejecución

Si ya te encuentras en el directorio build, puedes pasar directamente al siguiente paso. De lo contrario, deberás ingresar:

```
$ cd build
```

Luego:

```
$ ./taller_editor
```

### 6.2. Funcionamiento

Luego tendrás la portada del editor, donde podrás decidir si crear un mapa o editar uno creado anteriormente. En el caso de querer editar un mapa, es necesario ingresar únicamente el **nombre** del archivo, sin incluir la extensión **.yaml**.

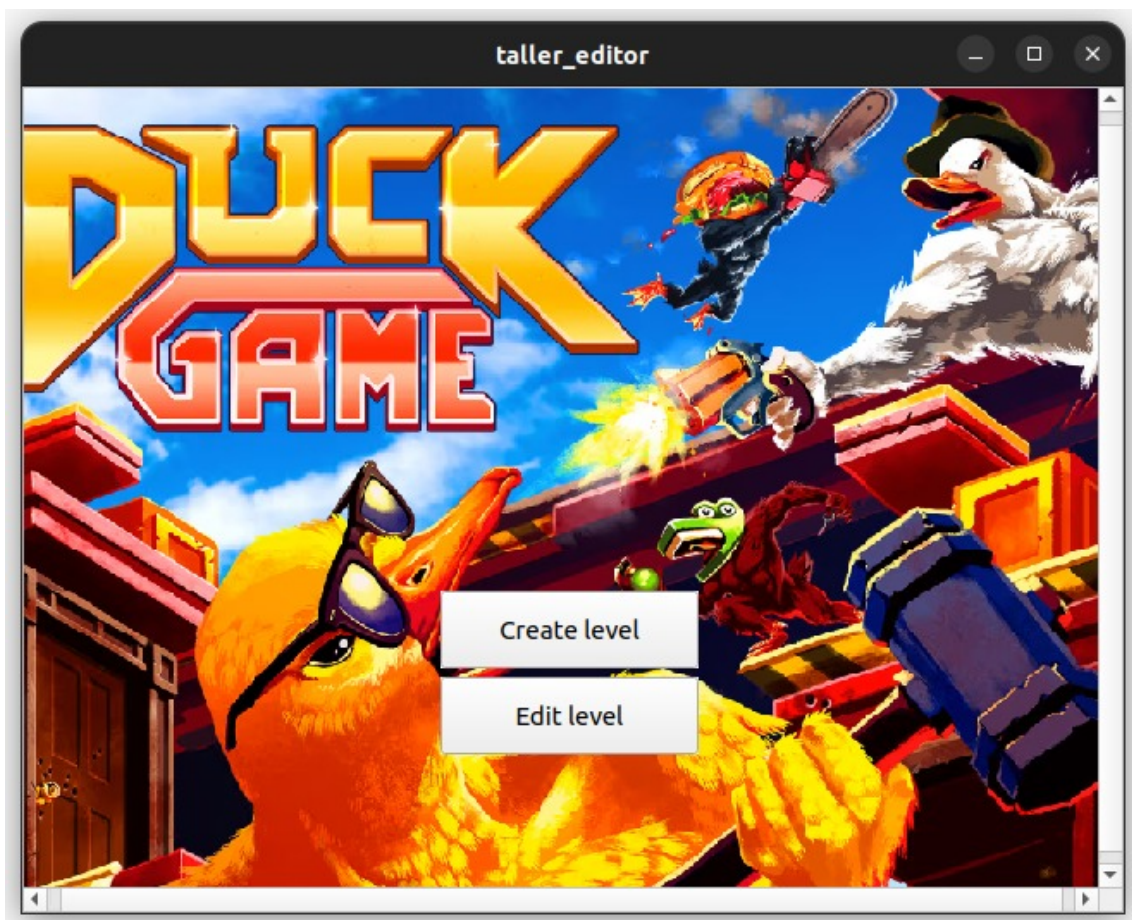


Figura 8: Pantalla principal del editor

Una vez que estés editando el mapa, haciendo clic derecho, aparecerá un repertorio de opciones donde podrás elegir:

- El respawn de los personajes, armas, cajas y protecciones.
- Las diferentes plataformas y paredes.
- El fondo del juego.



Figura 9: Repertorio del editor

Luego deberás presionar **Save** y nombrarlo como desees, es importante no olvidar setear una posición de respawn para el pato. Podrás hacer un diseño único para disfrutar del juego a tu manera.

## 7. Cheats

Si se desea probar el juego más fácilmente, habrá *cheats* disponibles que harán esto posible. Algunas de sus funcionalidades son:

- **Tecla 2:** Permitirá equipar diferentes armas al jugador 1, facilitando la prueba de sus características.
- **Tecla 8:** Permitirá equipar diferentes armas al jugador 2, facilitando la prueba de sus características.



- **Tecla T:** Declarará al jugador 1 como ganador de la ronda.
- **Tecla V:** Declarará al jugador 1 como ganador de forma inmediata.
- **Tecla R:** Equipará a todos los patos con armadura completa.
- **Tecla B:** Aparecerá una caja en la posición de cada pato.