

Manual de usuario

Índice

1. Instalación	2
1.1. Requerimientos de software	2
1.2. Requerimientos de hardware	3
1.2.1. Requerimientos mínimos aproximados	3
1.3. Proceso de instalación: Linux	3
1.3.1. Desde un explorador web	3
1.3.2. Desde la terminal	4
1.3.3. Resumen comandos de instalación	6
2. Configuración	6
3. Forma de uso	8
3.1. Server	8
3.2. Cliente	8
3.2.1. Unirse a una sala	9
3.2.2. Crear una sala	10
3.2.3. Sala de espera	10
3.2.4. Dentro del juego	10
3.2.5. Configuración del teclado	11
3.2.6. Estado de los demas jugadores	11
3.2.7. Usar un arma/herramienta	12
3.2.8. Potencia variable	13
3.2.9. Temporizador	14
3.2.10. Cámara libre	15
3.3. Editor de mapas	16
3.3.1. Cantidad de jugadores	16
3.3.2. Creación de objetos	17
3.3.3. Configurando un gusano	18
3.3.4. Configurando una viga	19
3.3.5. Elegiendo las armas/herramientas	19
3.3.6. Guardado de mapas	21
3.3.7. Cargado de mapas	21

1. Instalación

A continuación, se mostrarán los requerimientos de software y hardware para instalar el proyecto junto al proceso de instalación del mismo.

1.1. Requerimientos de software

Para instalar el proyecto, es necesario utilizar como sistema operativo Linux, y las herramientas necesarias para el mismo son las siguientes:

- Box2D versión 2.1.2
- Qt versión 5:
 - Instalación Linux:

Ejemplo para sistema operativo de 64 bits. Para descargar otra versión, visitar la página oficial de qt: "<https://www.qt.io/download#section2>"

```
wget http://download.qt.io/official_releases/qt/5.7/5.7.0/qt-  
opensource-linux-x64-5.7.0.run  
chmod +x qt-opensource-linux-x64-5.7.0.run  
./qt-opensource-linux-x64-5.7.0.run  
sudo apt-get install libfontconfig1  
sudo apt-get install mesa-common-dev  
sudo apt-get install libglu1-mesa-dev -y
```

Proseguir con los pasos de instalación de Qt.

- QtMultimedia versión 5:
 - Instalación Linux:

```
sudo apt install libqt5multimedia5-plugins  
gstreamer1.0-plugins-base-apps libqt5gstreamer-  
dev qtmultimedia5-dev
```
- Yaml-cpp versión 0.6:
 - Instalación Linux:

```
git clone https://github.com/jbeder/yaml-cpp.git  
cd yaml-cpp  
mkdir build  
cd build  
cmake ..  
make  
make install
```
- g++ versión 5.4 o superior:
 - Instalación Linux:

```
sudo apt-get install build-essential
```
- Cmake versión 3.1 o superior
 - Instalación Linux:

```
sudo apt-get install cmake
```

1.2. Requerimientos de hardware

El proyecto fue probado en una máquina virtual con las siguientes características:

1.2.1. Requerimientos mínimos aproximados

- Ubuntu 14.04
- Memoria Ram: 2048 MB
- Memoria de Video: 16 MB
- Disco Rigido: 30 GB
- 1 procesador
 - Disquete
 - Optica
 - Disco Duro

1.3. Proceso de instalación: Linux

Para instalar el proyecto, lo primero que debe hacerse es descargar el proyecto de alguna de las dos maneras que se muestran a continuación:

1.3.1. Desde un explorador web

Se debe entrar al repositorio donde se encuentra el proyecto y bajar la versión más reciente tal y como se indica en la siguiente imagen:

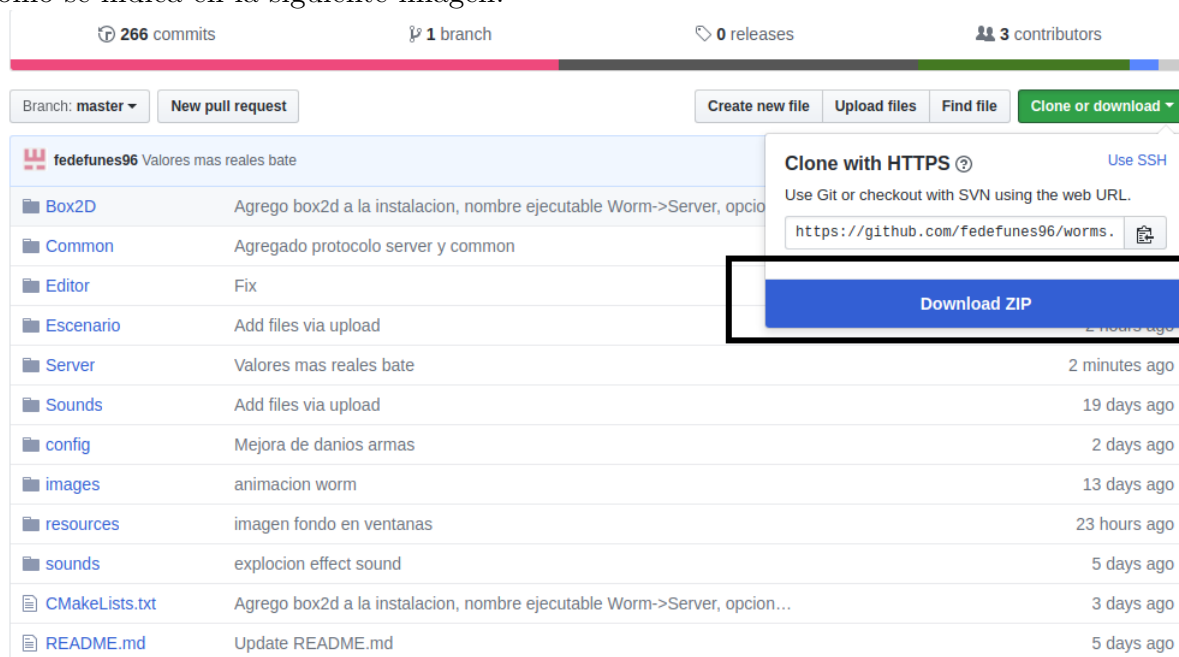


Figura 1: Enlace: <https://github.com/fedefunes96/worms>

Una vez descargado el proyecto, se extrae el mismo en cualquier carpeta (Ésta se utilizara de instalador, más adelante puede elegir donde desea instalar específicamente el proyecto).

1.3.2. Desde la terminal

Para instalar el proyecto desde la terminal, se deben de ejecutar los siguientes comandos ubicados en el directorio donde se desea bajar el instalador:

```
mkdir <DIR_INSTALACION>
cd <DIR_INSTALACION>
git clone https://github.com/fedefunes96/worms.git
cd worms
```

A modo de ejemplo, se muestra como deberían de quedar las carpetas:

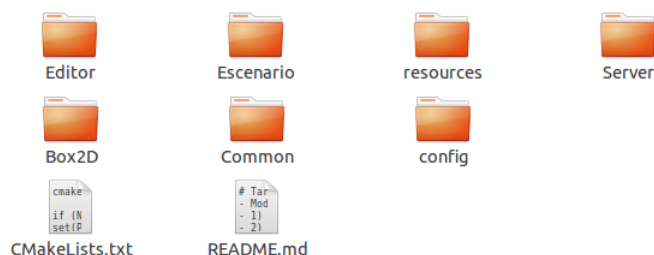


Figura 2: Estado de carpertas parte I

Lo siguiente es crear una nueva carpeta llamada "build" en donde se ha descomprimido el proyecto, con el siguiente comando:

```
mkdir build
cd build
```

Se muestra una imagen a modo de ejemplo:

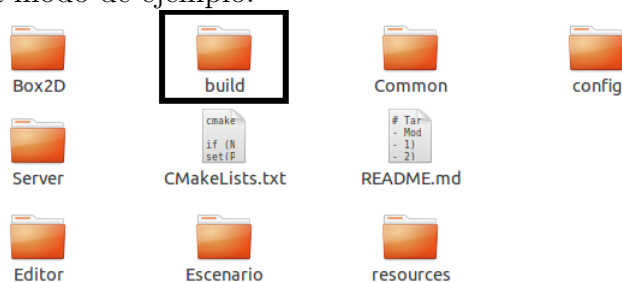


Figura 3: Estado de carpertas parte II

Una vez creada la carpeta "build", ingresamos a la misma, abrimos la terminal desde esa dirección y ejecutamos el siguiente comando:

```
cmake -DCMAKE_INSTALL_PREFIX=<DIR_INSTALACION> ..
```

Donde <DIR_INSTALACION> es la ruta donde se desea instalar el proyecto. En este caso y a modo de ejemplo, usaremos la ruta /home/fede/worms. **Importante:** Es necesario crear la carpeta donde se instalará el juego previamente a instalar el mismo en dicha dirección.

```
fede@fedesktop:~/Escritorio/Ejemplo/worms-master/build$ cmake -DCMAKE_INSTALL_PREFIX=/home/fede/worms ..
-- Configuring done
-- Generating done
-- Build files have been written to: /home/fede/Escritorio/Ejemplo/worms-master/build
fede@fedesktop:~/Escritorio/Ejemplo/worms-master/build$
```

Figura 4: Ejecución del cmake

En caso de error:

- Revisar la ruta dada al cmake.
- Verificar que existe la carpeta donde se desea instalar el proyecto y si se tienen los permisos necesarios.
- Revisar la versión de cmake y corroborar que sea igual o superior a la requerida (Ver sección **Requerimientos de software**). Para ver la versión de cmake: `cmake --version`

Una vez haya terminado el cmake, prosigue ejecutar el siguiente comando:

make

```
fede@fedesktop:~/Escritorio/Ejemplo/worms-master/build$ make
[ 0%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2DDistanceJoint.o
[ 1%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2DFrictionJoint.o
[ 2%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2GGearJoint.o
[ 2%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2Joint.o
[ 3%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2LineJoint.o
[ 4%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2MouseJoint.o
[ 4%] Building CXX object Box2D/Box2D/CMakeFiles/Box2D.dir/Dynamics/Box2PrismaticJoint.o
```

Figura 5: Ejecución del make

En caso de error:

- Revisar que estén instalados todos los requerimientos de software y corroborar que sean igual o superior a la requerida (Ver sección **Requerimientos de software**)

Una vez haya terminado el make, solo queda ejecutar un último comando:

make install

```
fede@fedesktop:~/Escritorio/Ejemplo/worms-master/build$ make install
[ 25%] Built target Box2D
[ 66%] Built target Server
[ 66%] Automatic MOC and UI for target Editor
[ 66%] Built target Editor_autogen
[ 77%] Built target Editor
[ 77%] Automatic MOC and UI for target GameWindow
[ 77%] Built target GameWindow_autogen
[100%] Built target GameWindow
Install the project...
-- Install configuration: ""
-- Installing: /home/fede/worms/bin/Server
-- Set runtime path of "/home/fede/worms/bin/Server" to ""
-- Installing: /home/fede/worms/bin/Editor
-- Set runtime path of "/home/fede/worms/bin/Editor" to ""
-- Installing: /home/fede/worms/bin/GameWindow
-- Set runtime path of "/home/fede/worms/bin/GameWindow" to ""
-- Installing: /home/fede/worms/lib/libBox2D.a
fede@fedesktop:~/Escritorio/Ejemplo/worms-master/build$
```

Figura 6: Ejecución del make install

Una vez realizados todos estos pasos en el orden indicado, el proyecto se habrá instalado en la ruta elegida. Para ejecutar el juego: Ver sección **Forma de uso**

1.3.3. Resumen comandos de instalación

Se mostrará un resumen de los comandos necesarios para instalar el proyecto:

```
mkdir <DIR_INSTALACION>
cd <DIR_INSTALACION>
git clone https://github.com/fedefunes96/worms.git
cd worms
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=<DIR_INSTALACION> ..
make
make install
```

2. Configuración

Una vez instalado el proyecto, nos ubicamos en la ruta donde hayamos instalado el mismo, de manera que veremos las siguientes carpetas:



Figura 7: Carpetas: Ya instalado el proyecto

Dentro de la carpeta "config" se encuentra el archivo de configuración del juego: "config.yaml". El mismo puede ser editado a mano utilizando cualquier editor de texto. Para reconocer que representa cada valor del archivo, se muestra a continuación una imagen con los parámetros configurables.

Water: 0	→	Nivel del agua
Wind:		
- 0.2	→	Mín. velocidad viento
- 1.0	→	Max. velocidad viento
Worm:		
- 0	→	Ángulo inicial
- 0.5	→	Longitud
- 0.5	→	Altura
- 0	→	Restitución
- 2.0	→	Vel. movimiento
- 2.0	→	Vel. salto adelante X
- 6.0	→	Vel. salto adelante Y
- 0.5	→	Vel. salto atrás X
- 4.0	→	Vel. salto atrás Y
- 10	→	Altura mínima de caída daño
- 1	→	Daño por metro de caída
- 25	→	Daño máximo por caída
Small Girder:		
- 1.5	→	Longitud
- 0.4	→	Altura
Large Girder:		
- 3	→	Longitud
- 0.4	→	Altura
Bazooka:		
- 10	→	Velocidad bala
- 0.25	→	Radio bala
- 35	→	Daño max
- 6.0	→	Max. empuje
- 2.0	→	Radio de explosión
Mortar:		
- 10	→	Velocidad bala
- 0.25	→	Radio bala
- 25.0	→	Daño max
- 4.0	→	Max. empuje
- 2	→	Radio de explosión
- 6	→	Cantidad de fragmentos
- 0.5	→	Velocidad fragmento
- 0.10	→	Radio fragmento
- 5.0	→	Daño fragmento
GreenGrenade:		
- 10	→	Velocidad bala
- 0.25	→	Radio bala
- 0.5	→	Restitución
- 35.0	→	Daño max
- 3.0	→	Max. empuje
- 2	→	Radio de explosión
RedGrenade:		
- 10	→	Velocidad bala
- 0.25	→	Radio bala
- 0.5	→	Restitución
- 30.0	→	Daño max
- 2.0	→	Max. empuje
- 2.0	→	Radio de explosión
- 6	→	Cantidad de fragmentos
- 0.5	→	Velocidad fragmento
- 0.10	→	Radio fragmento
- 5.0	→	Daño fragmento
Banana:		
- 10	→	Velocidad bala
- 0.25	→	Radio bala
- 1	→	Restitución
- 60.0	→	Daño max
- 6.0	→	Max. empuje
- 2.0	→	Radio de explosión
HolyGrenade:		
- 10	→	Velocidad bala
- 0.25	→	Radio bala
- 0.5	→	Restitución
- 130.0	→	Daño max
- 2	→	Max. empuje
- 6.0	→	Radio de explosión
Dynimite:		
- 0.25	→	Radio
- 0	→	Restitución
- 50	→	Max daño
- 2	→	Max empuje
- 2.0	→	Radio explosión
Bate:		
- 10	→	Empuje
- 10	→	Daño
AirAttack:		
- 1.0	→	Velocidad
- 0.25	→	Radio
- 40.0	→	Max daño
- 250.0	→	Altura de caída (Inútil) - Deprecated
- 4.0	→	Max empuje
- 2.0	→	Radio explosión

Figura 8: Configurar "config.yaml"

Importante: No se asegura un buen funcionamiento del juego si se modifican aquellos parámetros con valores inapropiados (Negativos) o se modifiquen el radio de los misiles y/o el tamaño de los gusanos.

3. Forma de uso

Como el proyecto consta de tres ejecutables diferentes:

- Server
- Client
- Editor

Se explicará como ejecutar cada uno en una sección separada.

3.1. Server

Antes de poder jugar al worms, es necesario levantar un servidor para que los jugadores puedan conectarse al mismo. Para ello, es necesario ejecutar el archivo "Server" ubicado dentro de la carpeta "bin", utilizando el siguiente comando:

`./Server port`

Donde port es el puerto donde se desea levantar el servidor (Ej: 7777), a modo de ejemplo, se muestra la ejecución de un servidor:

```
fede@fede-desktop:~/Escritorio/worms/bin$ ./Server 7777
```

Figura 9: Ejecución de un server en el puerto 7777

Ahora ya tenemos listo el servidor para aceptar jugadores y crear partidas, si se desea interrumpir el mismo, debe de escribirse el caracter "Q" por la entrada estándar como se muestra a continuación:

```
fede@fede-desktop:~/Escritorio/worms/bin$ ./Server 7777
Q
fede@fede-desktop:~/Escritorio/worms/bin$ █
```

Figura 10: Interrumpiendo el servidor

Importante: El server no finalizará de inmediato, sino que bloqueará nuevas conexiones de jugadores hasta que terminen todas las partidas que están en curso. Una vez finalizadas las partidas, el servidor terminará su ejecución.

3.2. Cliente

Para poder jugar efectivamente al worms, hay que conectarse a un servidor. Para ello, hay que ejecutar el archivo "Client" ubicado dentro de la carpeta "bin".

Una vez ejecutado el mismo, se mostrará la siguiente ventana:



Figura 11: Conexión de un jugador a un servidor

Donde:

- Port: Puerto del servidor donde se desea conectar.
- Ip: Dirección ip del servidor donde se desea conectar.

Ya conectado al servidor, puede optarse por crear o unirse a una sala:

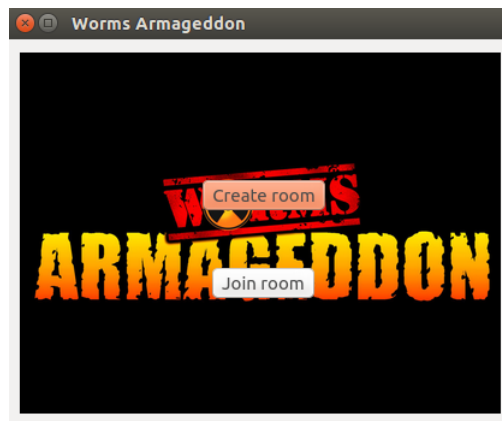


Figura 12: Crear o unirse a una sala

3.2.1. Unirse a una sala

Se mostrarán a continuación todas las salas disponibles para unirse, basta con elegir una y seleccionar el botón "Join" para unirse a la misma:

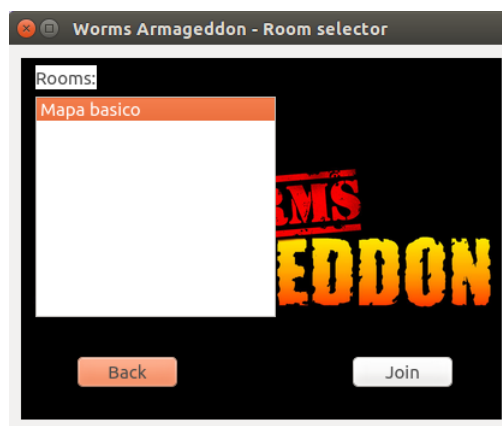


Figura 13: Unirse a una sala

3.2.2. Crear una sala

Se mostrarán a continuación todos los mapas disponibles para crear una sala, así como también un nombre que debe de elegirse para la sala:

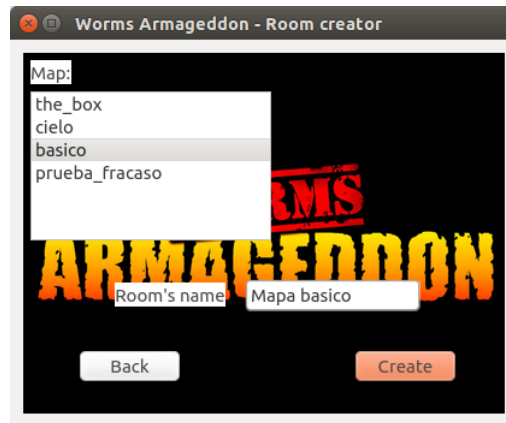


Figura 14: Crear a una sala

3.2.3. Sala de espera

Ya creada una sala o unido a una, solo queda esperar a que se conecten los demás jugadores para empezar la partida. Si se desea salir de la sala, basta con apretar el botón "Exit room" para salir de la misma.



Figura 15: Sala de espera

3.2.4. Dentro del juego

Una vez iniciada la partida, se notificará el turno del jugador actual en la siguiente porción de la pantalla:

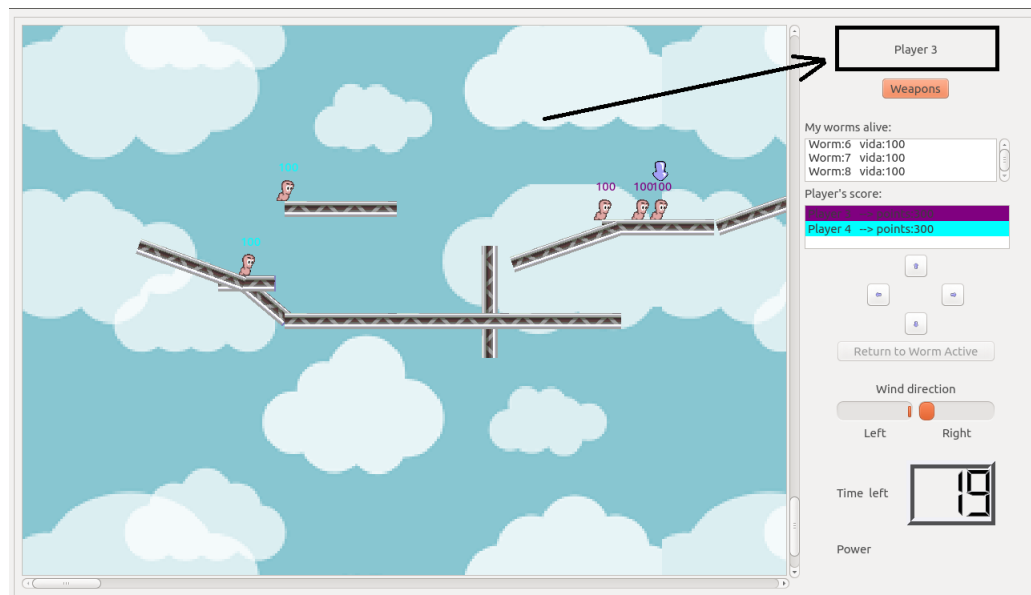


Figura 16: Dentro del juego: Turnos

3.2.5. Configuración del teclado

En la siguiente imagen se muestra como ejecutar las diversas acciones dentro del juego:

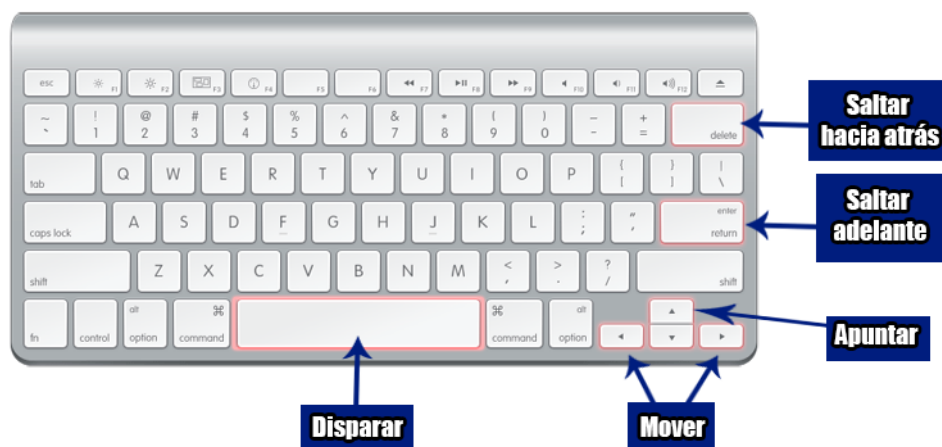


Figura 17: Configuración del teclado

3.2.6. Estado de los demas jugadores

Para saber en todo momento el estado de los demás jugadores, basta con mirar donde se indica en la siguiente imagen para ver la vida en conjunto de todos los worms de todos los jugadores:

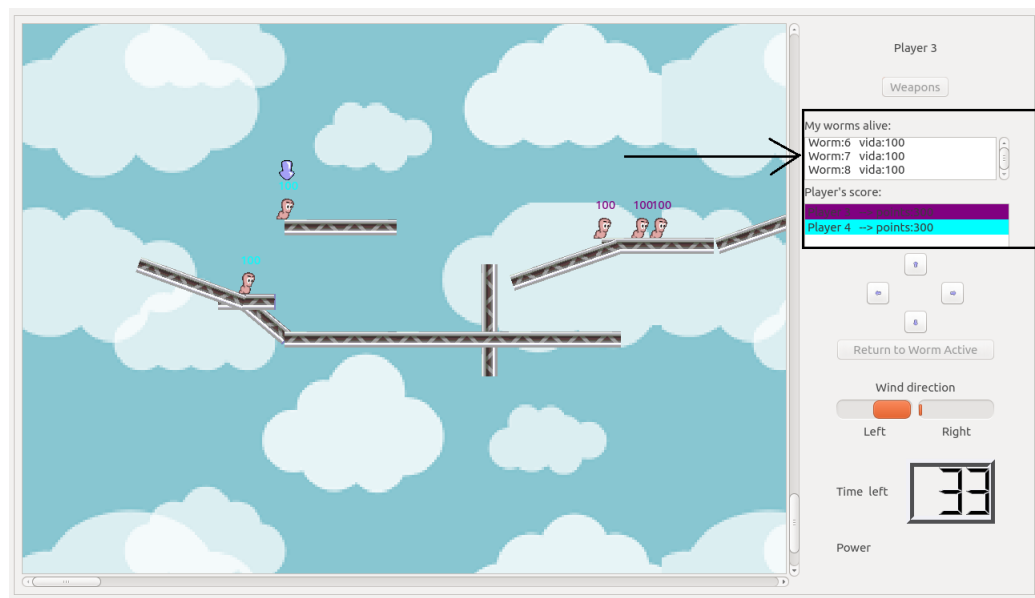


Figura 18: Estado de los jugadores

3.2.7. Usar un arma/herramienta

Para elegir un arma/herramienta para utilizar, basta seleccionar donde indica la siguiente imagen:

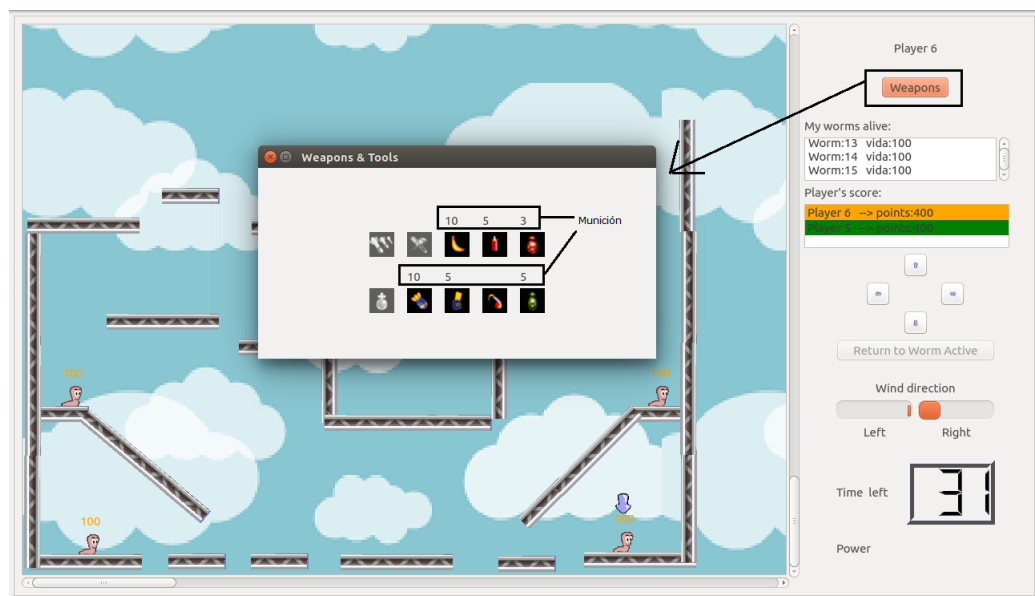






Figura 19: Seleccionando un arma

Aquellos números que se ven en las armas, indican la munición que actualmente se posee en dicha arma.

Diversas armas en el juego requieren más que solo apretar un botón para ejecutarse de la manera que uno desea, las siguientes armas las que poseen características extras:

▪ Mortar: 

- Potencia variable ✓

- Granada verde: 
 - Potencia variable ✓
 - Temporizador ✓
- Granada roja: 
 - Potencia variable ✓
 - Temporizador ✓
- Banana: 
 - Potencia variable ✓
 - Temporizador ✓
- Dynamita: 
 - Temporizador ✓
- Holy grenade: 
 - Potencia variable ✓
 - Temporizador ✓

A continuación, se explicará como utilizar estos parámetros extras de las armas/herramientas.

3.2.8. Potencia variable

La potencia variable se muestra en la siguiente sección de la pantalla:

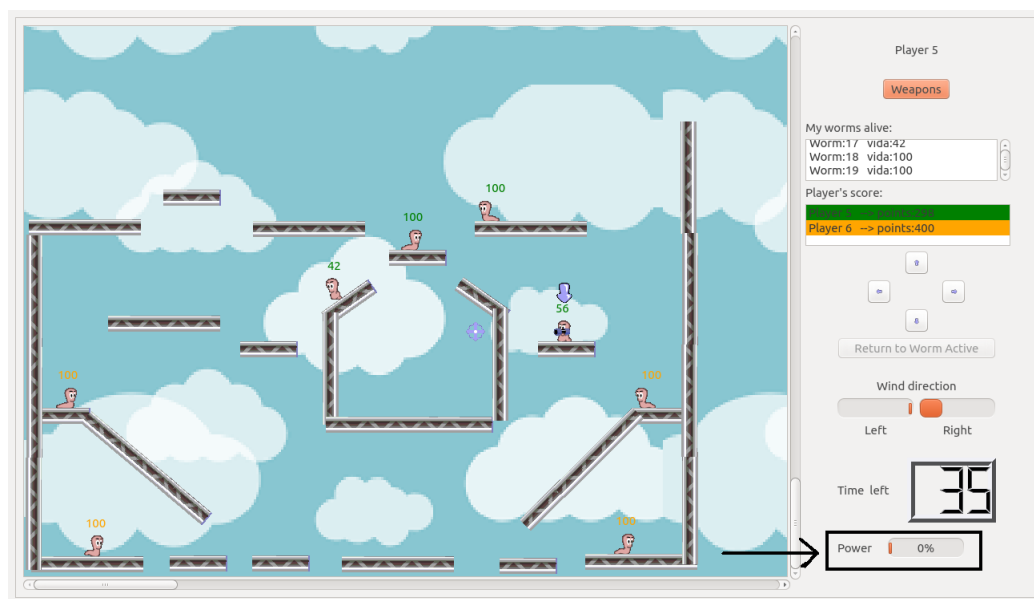


Figura 20: Estado potencia variable I

Para que un misil salga disparado con una potencia mayor, es necesario mantener apretado el botón "espacio" (Ver subsección **Configuración del teclado**) hasta que se obtenga la potencia necesaria, tal y como se ve en la siguiente imagen:

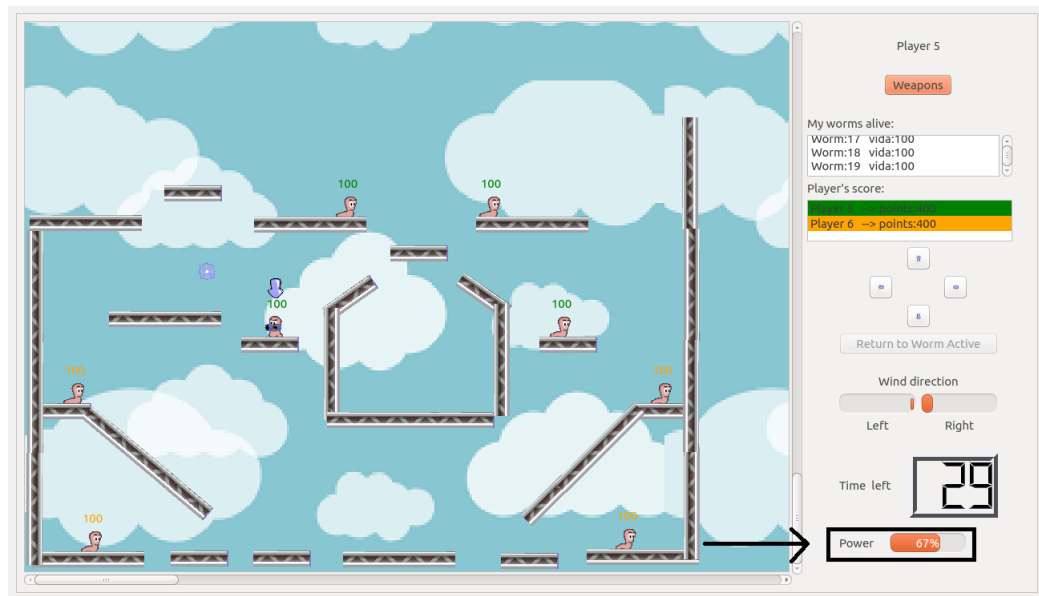


Figura 21: Estado potencia variable II

3.2.9. Temporizador

El temporizador se muestra en la siguiente sección de la pantalla:

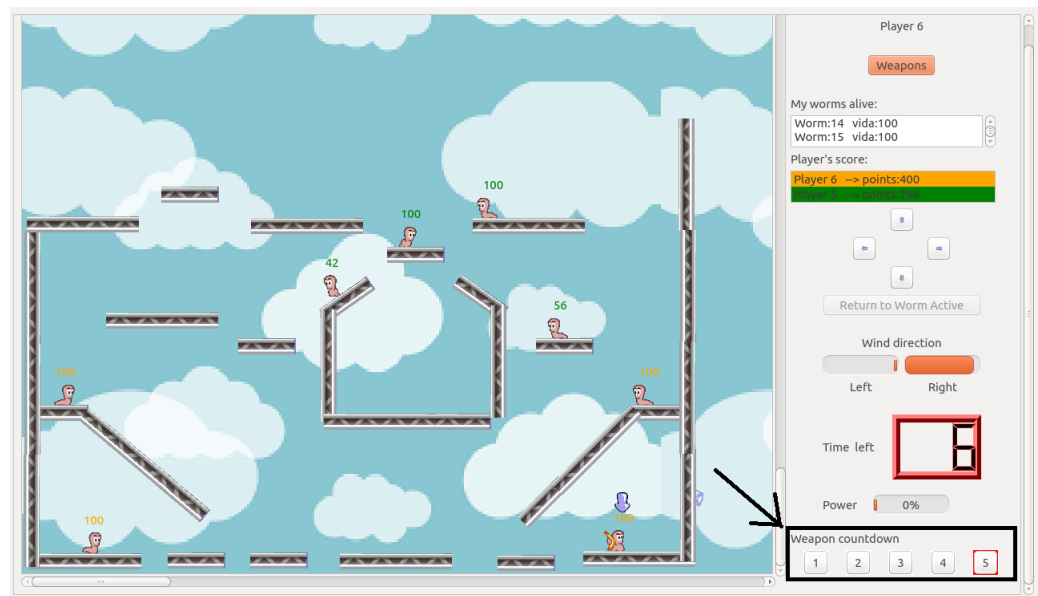


Figura 22: Estado temporizador I

Para establecer un tiempo de explosión, debe de elegirse cualquiera de los números que aparecen en la pantalla, lo que establecera la cantidad de segundos que habrá que esperar para que el arma explote, tal y como se indica en la siguiente imagen:

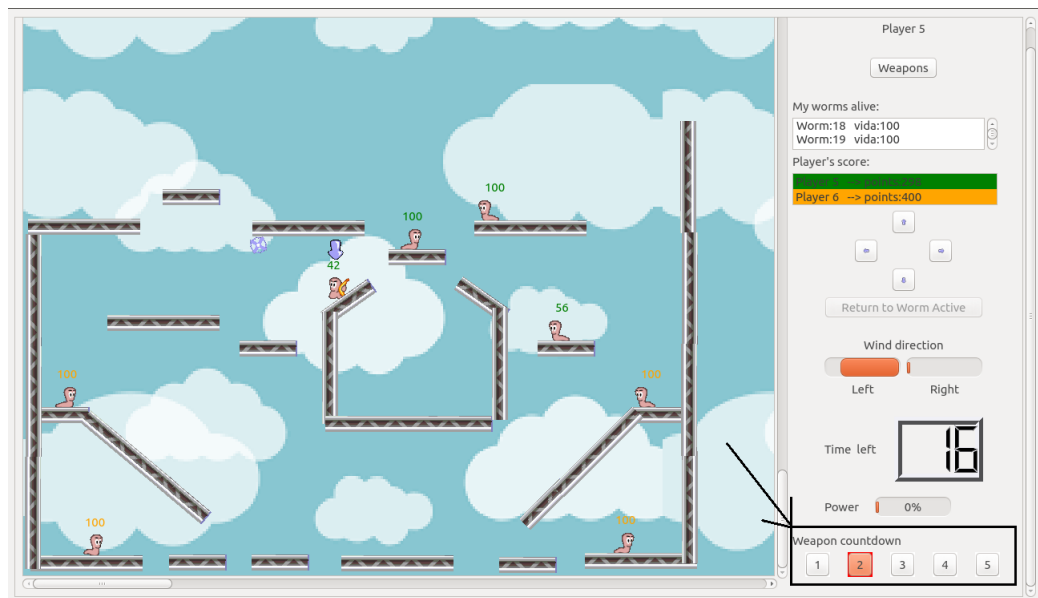


Figura 23: Estado temporizador II: Con tiempo 2 segundos

3.2.10. Cámara libre

Si en algún momento se desea mover la cámara libre por el mapa, basta con utilizar los siguientes botones para desplazar la pantalla:

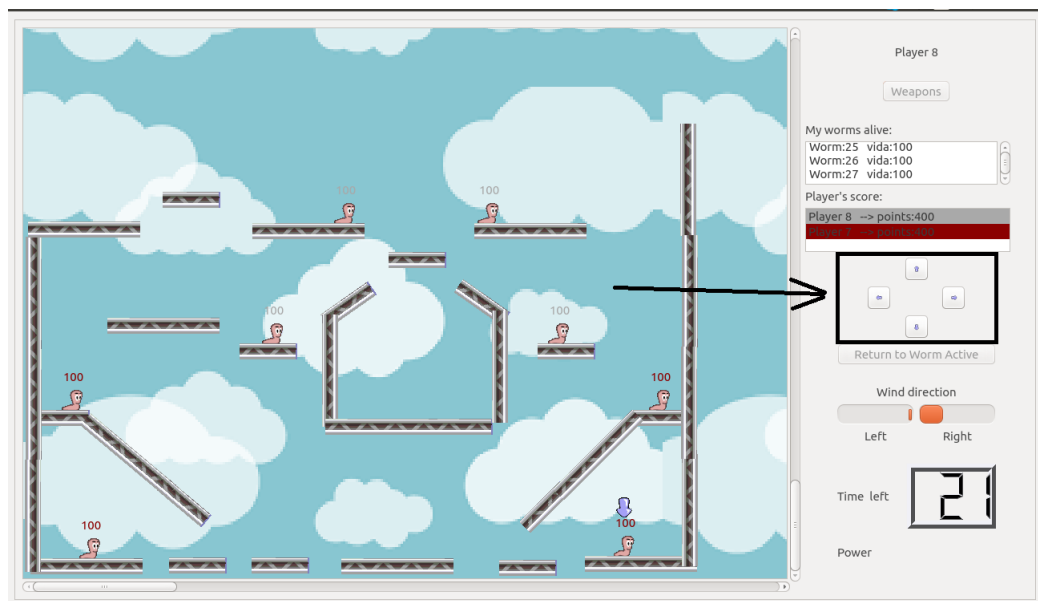


Figura 24: Desplazamiento de pantalla

Si en algún momento desea volver a centrar la cámara en el gusano actual, basta con apretar el siguiente botón:

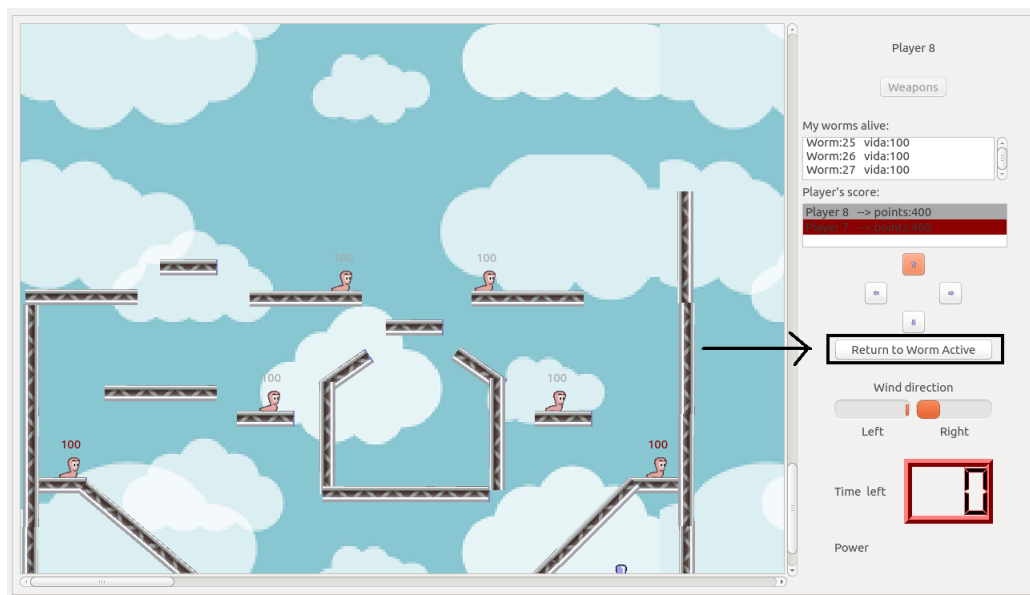


Figura 25: Centrar cámara en worm actual

3.3. Editor de mapas

Para poder crear mapas para jugar al worms, es necesario ejecutar el archivo "Editor" ubicado dentro de la carpeta "bin".

Una vez ejecutado el mismo, se mostrará la siguiente ventana donde, se puede crear un nuevo mapa o cargar uno ya diseñado para modificarlo:



Figura 26: Pantalla principal del editor

3.3.1. Cantidad de jugadores

Para elegir la cantidad de jugadores, basta con modificar el cuadro que se muestra a continuación:

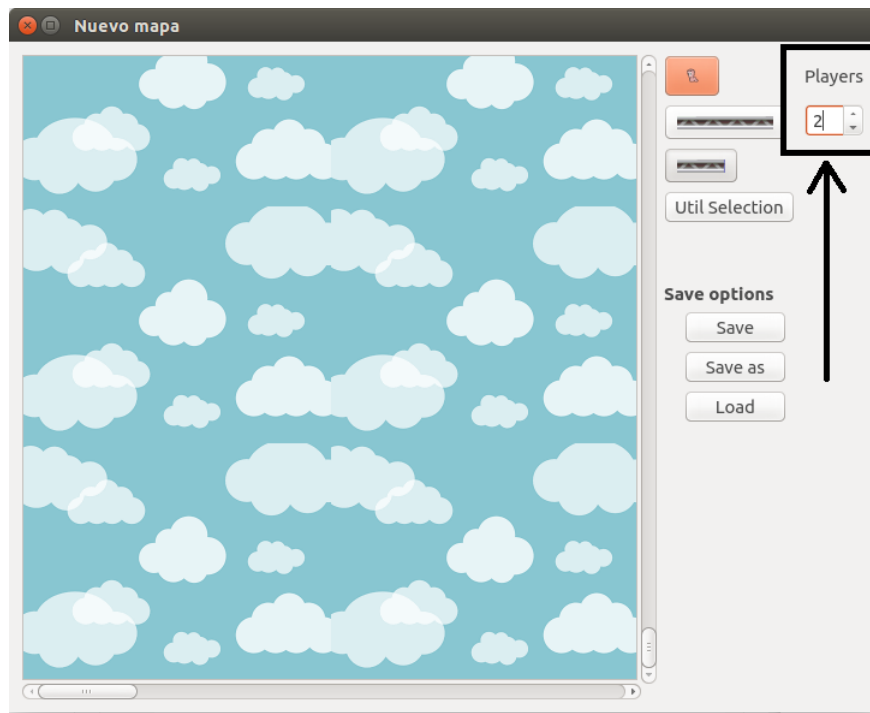


Figura 27: Eligiendo la cantidad de jugadores

Importante: Es recomendable elegir más de dos jugadores para un mapa.

3.3.2. Creación de objetos

Para crear un objeto en el editor de mapas, debe de apretarse cualquiera de los siguientes botones:

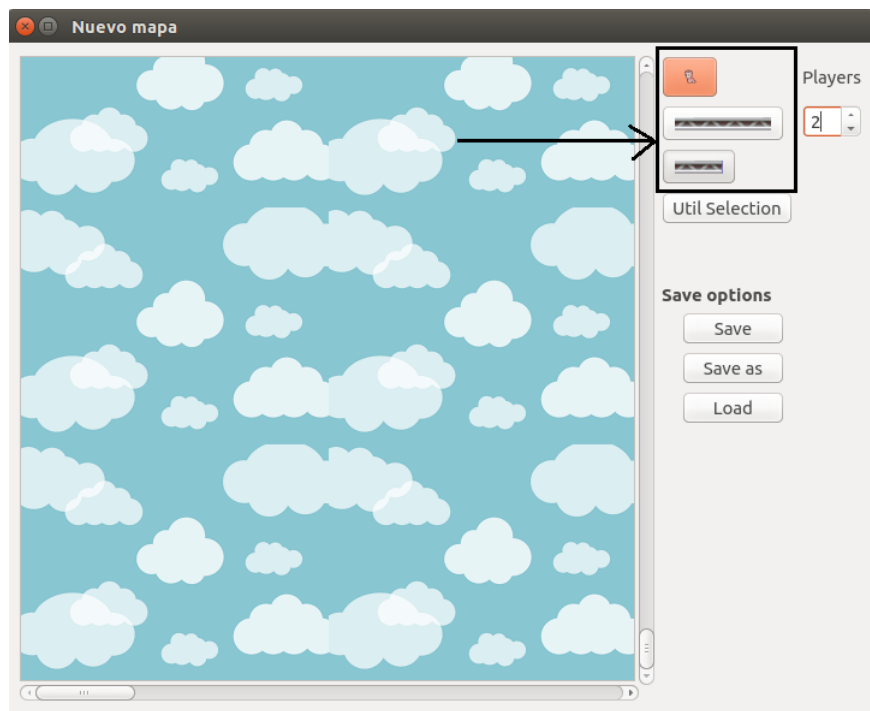


Figura 28: Creando un objeto I

Una vez seleccionado el botón, basta con tocar en cualquier parte de la pantalla para crear al mismo:

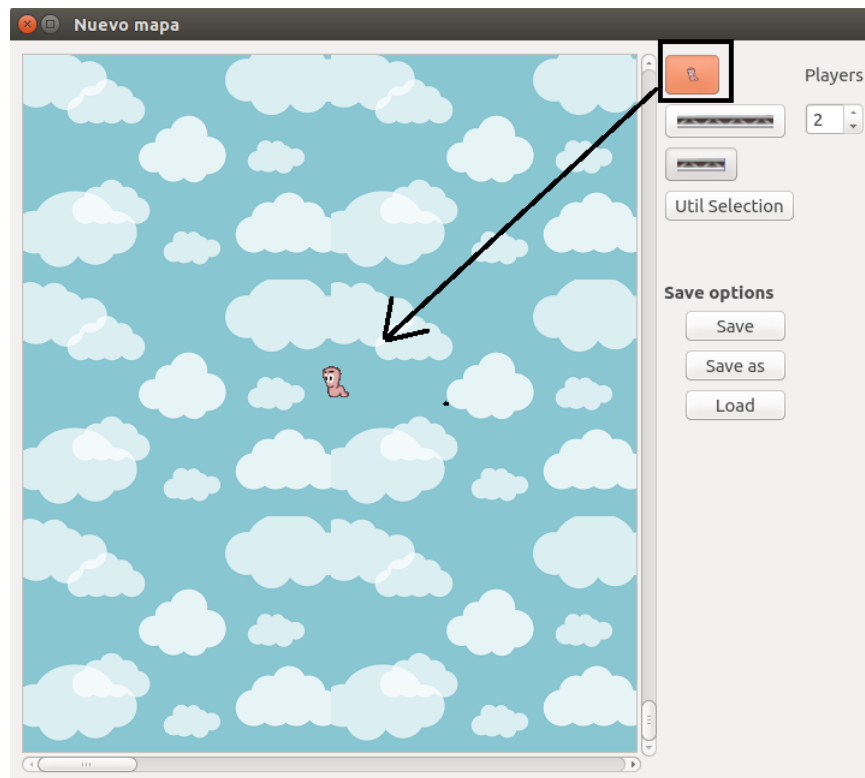


Figura 29: Creando un objeto II: En este caso, un gusano

3.3.3. Configurando un gusano

Si se desea cambiar los atributos de un gusano creado, basta con tocar click en el mismo para que se ponga transparente y se pueda editar el mismo en la siguiente porción de la pantalla:

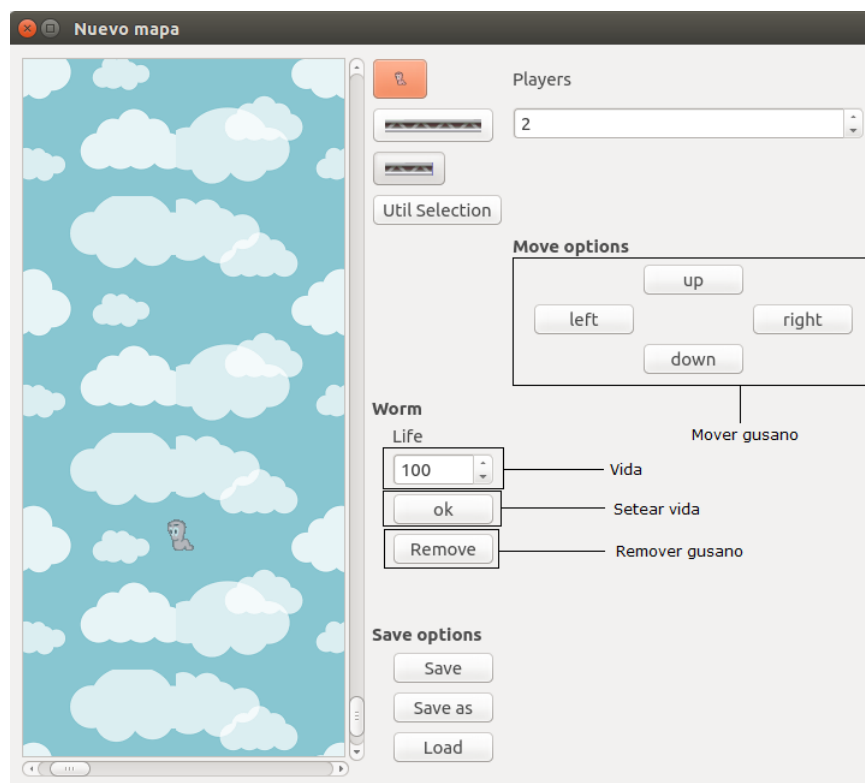


Figura 30: Configurando un gusano

3.3.4. Configurando una viga

Si se desea cambiar los atributos de una viga creada, basta con tocar click en el mismo para que se ponga transparente y se pueda editar el mismo en la siguiente porción de la pantalla:

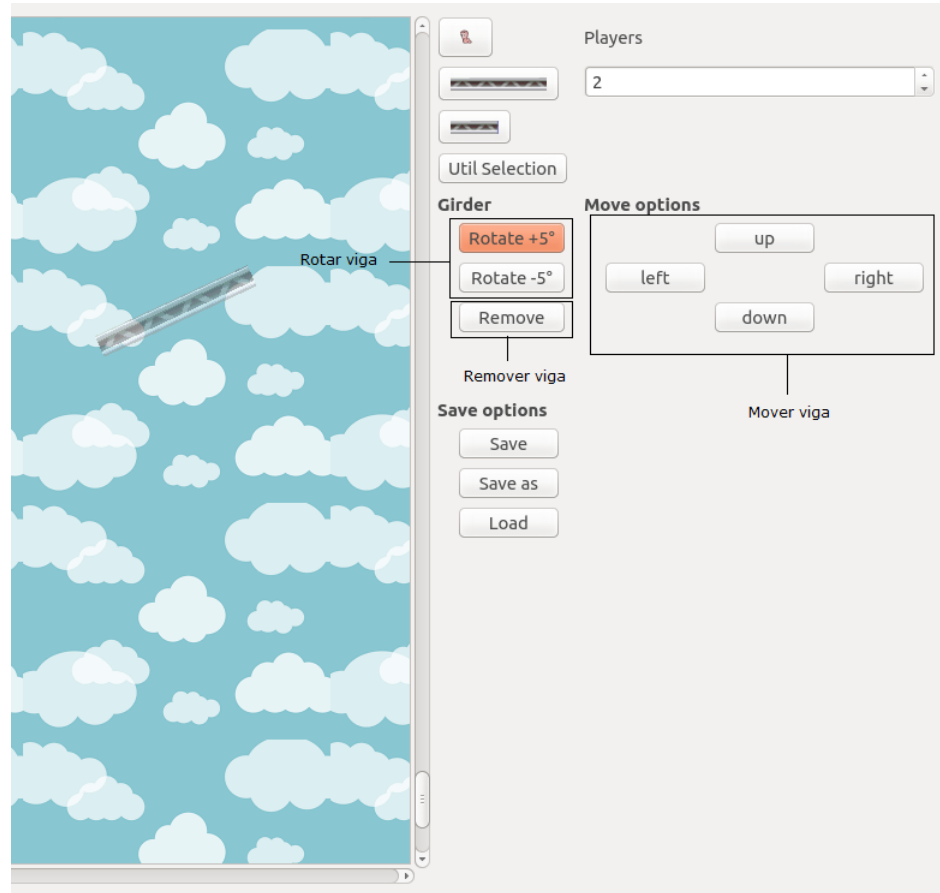


Figura 31: Configurando una viga

3.3.5. Elegiendo las armas/herramientas

Para elegir que armas se utilizaran efectivamente en el mapa seleccionado, basta con apretar el botón "Util selection" indicado en la siguiente pantalla:

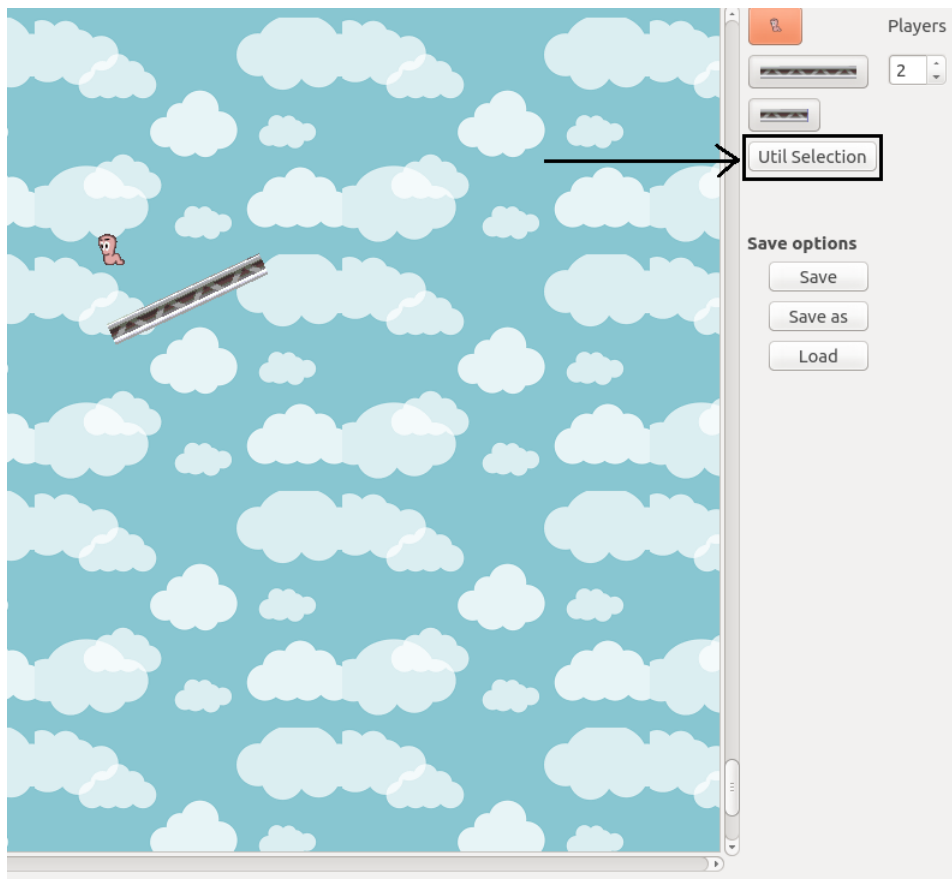


Figura 32: Eligiendo las armas/herramientas del mapa

Una vez seleccionado el botón, se abrirá la siguiente ventana donde se podrán seleccionar que armas/herramientas se utilizarán y la munición que van a tener las mismas, también, está la opción de elegir "Infinite ammo" para que tenga municiones infinitas:

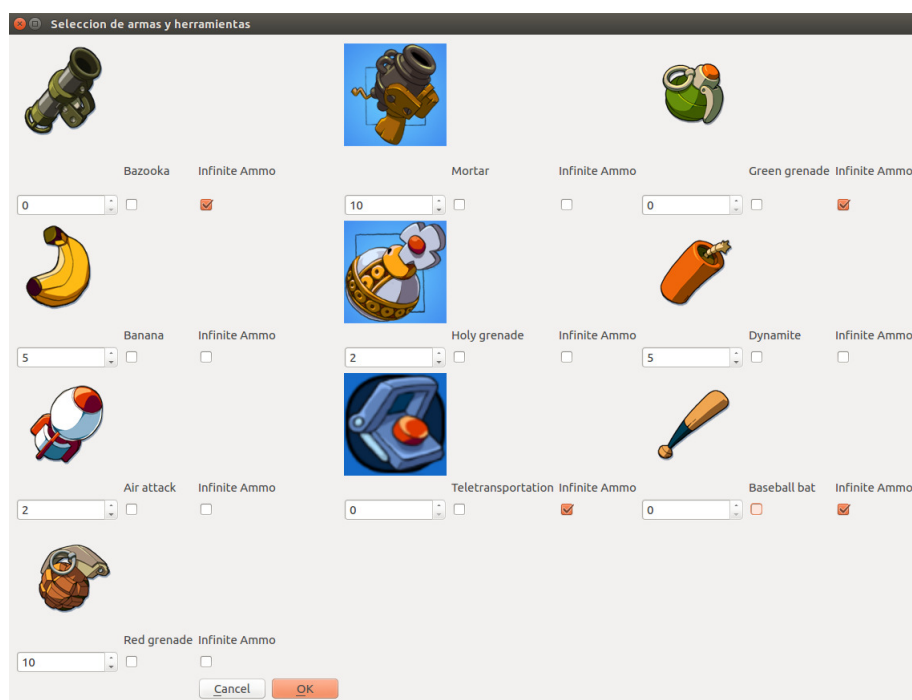


Figura 33: Configurando las armas/herramientas

3.3.6. Guardado de mapas

Una vez finalizado el mapa, puede seleccionarse cualquiera de las siguientes dos opciones para guardar el mismo:

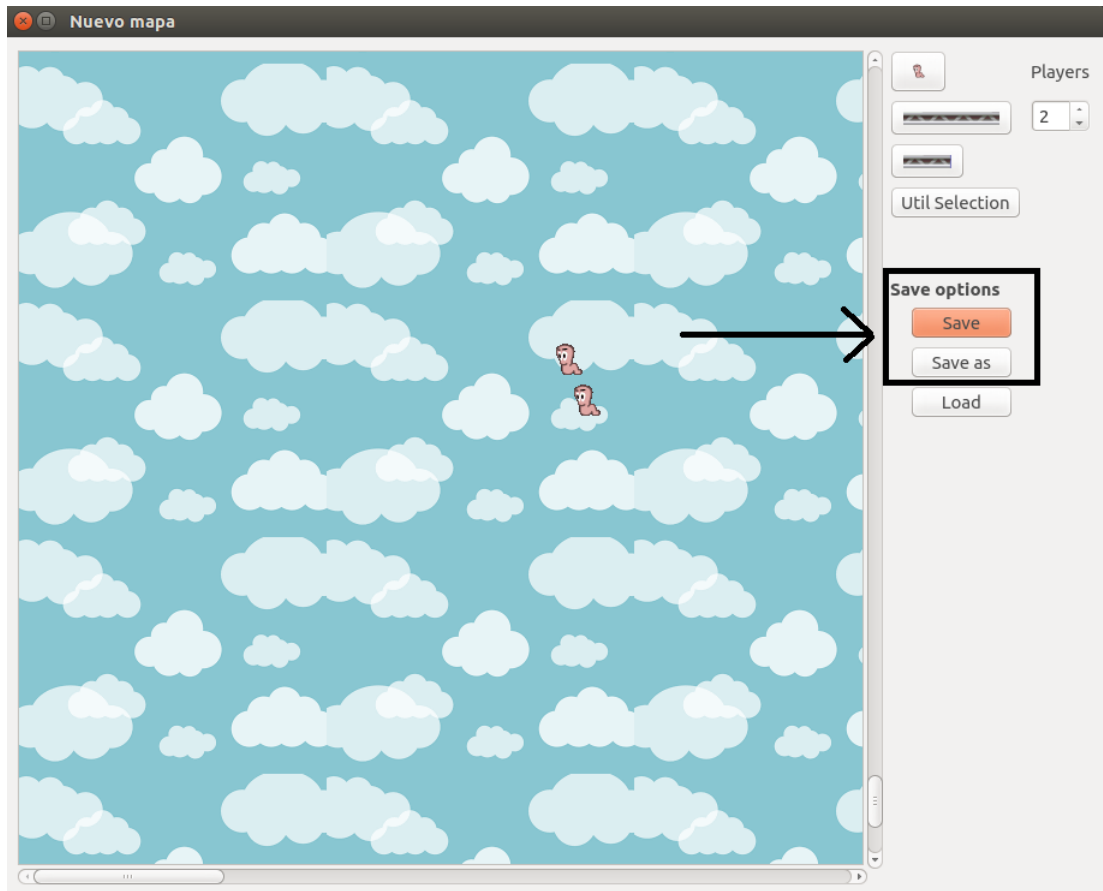


Figura 34: Guarda el mapa creado

Importante: Si se desea que el mapa se cargue en el servidor, debe de ubicarse al mismo dentro de la carpeta "resources/maps" en la ruta de instalación del juego.

3.3.7. Cargado de mapas

Si en algún momento se desea cargar un mapa ya creado, basta pulsar el siguiente botón y buscar el mapa a modificar:

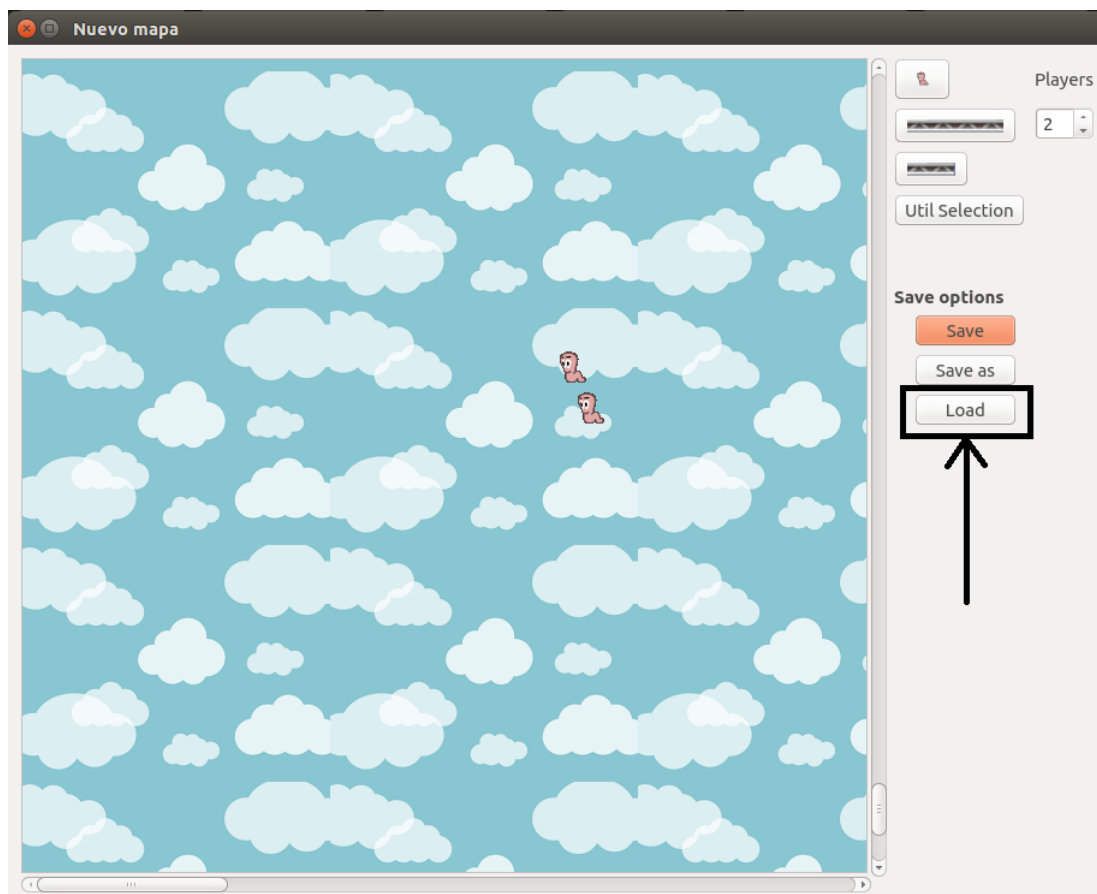


Figura 35: Cargando un mapa ya creado