

Jazz JackRabbit 2

Repositorio del Grupo 7 para la materia Taller de Programación I - Curso Veiga

Corrector

Leonardo Giampieri

Integrantes

- **Avalos, Victoria** - 108434
- **Castro Martinez, José Ignacio** - 106957
- **Diem, Walter Gabriel** - 105618

Manual del proyecto

Se presenta el manual del proyecto: [Manual del proyecto](#).

Documentación técnica

Se presenta la documentación técnica: [Documentación Técnica](#).

Manual para ejecución de pruebas

Se presenta el manual para la ejecucion de pruebas: [Tests](#).

Manual de usuario

Se procederá a explicar cómo instalar las dependencias y el juego, cómo correrlo y cómo usarlo.

Tabla de contenidos:

1. [Cómo buildear y correr en local](#)
2. [Cómo setupear y correr con Vagrant](#)
3. [Cómo funciona la interfaz del juego](#)
4. [Acerca del juego](#)
5. [Gameplay](#)

Cómo buildear y correr en local

Para instalar los paquetes necesarios correr el script:

```
./scripts/setup_libs.sh
```

Para crear los ejecutables (cliente y servidor) correr el script:

```
./scripts/build.sh -release
```

Ejecución sin Valgrind

Los ejecutables se encontrarán en la carpeta recién creada **build**. Se pueden ejecutar desde cualquier lugar utilizando los scripts:

```
/folder_path_to_source/scripts/run_client.sh
```

```
/folder_path_to_source/scripts/run_server.sh <puerto>
```

O también se puede correr el siguiente comando estando con la terminal situada en la carpeta del root del proyecto:

```
./build/client
```

```
./build/server <puerto>
```

Ejecución con Valgrind

Los ejecutables se encontrarán en la carpeta recién creada **build** por lo que para ejecutarlos con Valgrind y los archivos de supresión hay que correr, estando parado con la terminal en el root del proyecto:

```
valgrind --tool=memcheck --leak-check=full --show-reachable=yes --trace-children=yes --track-fds=yes --track-origins=yes --suppressions=./valgrind/all.sup ./build/client
```

```
valgrind --tool=memcheck --leak-check=full --show-reachable=yes --trace-children=yes --track-fds=yes --track-origins=yes --suppressions=./valgrind/all.sup ./build/server <puerto>
```

Cómo setupear y correr con Vagrant

Vagrant setup

Para instalar los paquetes necesarios para levantar la virtual machine (vm) con Vagrant de acuerdo a las configuraciones declaradas al Vagrantfile hay que correr los siguientes comandos.

Utilizaremos VirtualBox como VM Provider, para instalarlo hay que correr:

```
sudo apt-get --yes install virtualbox
```

```
sudo apt-get --yes install virtualbox-ext-pac
```

Probado con VirtualBox version 7.0.14

Para instalar Vagrant hay que correr:

```
sudo apt-get --yes install vagrant
```

Probado con Vagrant version 2.3.6

Utilizaremos un plugin de Vagrant para reiniciar la VM en provision time y así realizar un setup correcto. Para instalarlo hay que correr:

```
vagrant plugin install vagrant-reload
```

Cómo correr la VM

Para ejecutar y provisionar la VM hay que correr:

```
vagrant up
```

De esta manera se realizarán todos los pasos para realizar un correcto provisioning de una VM corriendo Ubuntu 22.04 con el entorno gráfico xubuntu-desktop.

Una vez que en la consola donde Vagrant corre se puede observar el mensaje "Provisioning complete", es decir, la VM tiene el juego listo para correr (en esta etapa la VM se reinició dos veces), hay que logearse en la VM con la contraseña **vagrant** y correr el cliente o servidor de la siguiente manera. El proceso de provisioning puede tardar 10 minutos con una PC y conexión a internet buena buena o más dependiendo del estado de la conexión y los recursos de la computadora host que instalará las dependencias.

Primero abrir una terminal, con **CTRL + ALT + T** y se pueden ejecutar el cliente y servidor de la siguiente manera:

```
/vagrant/scripts/run_client.sh
```

```
/vagrant/scripts/run_server.sh <puerto>
```

O también situándose en la carpeta root del proyecto con el comando:

```
cd /vagrant/
```

Luego, para ejecutar el servidor correr:

```
./build/server <puerto>
```

Y para ejecutar un cliente correr:

```
./build/client
```

Comandos útiles de Vagrant

1. Si se actualizaron los scripts de provisioning, se puede correr el siguiente comando para aplicar los scripts sobre la misma VM actual:

```
vagrant reload --provision
```

2. Si sólo se quiere reiniciar la VM:

```
vagrant reload
```

3. Si se cambió la estructura de la VM, es decir, se editó el Vagrantfile, hay que correr (preferentemente antes de guardar los cambios del Vagrantfile) el siguiente comando para destruir la VM (la para y luego la elimina):

```
vagrant destroy
```

Luego levantar una nueva VM con la nueva configuración con el comando:

```
vagrant up
```

4. Si se quiere apagar la VM:

```
vagrant halt
```

5. Para acceder a una terminal dentro del sistema operativo de la VM:

```
vagrant ssh
```

Cómo funciona la interfaz del juego

Nota: la música de fondo se puede activar o desactivar mediante un parámetro en el archivo de configuración.

Interfaz del servidor

El servidor se lanza por terminal corriendo el binario con el puerto como parámetro, y correrá en el hostname **localhost**, sinónimo para la IP local **127.0.0.1**. Para detener el servidor, ingresa **q** en la terminal y presiona Enter.

Interfaz del cliente

El cliente primero lanza una UI que llamaremos **Lobby**, una vez finalizadas las acciones del lobby e iniciada la conexión para comenzar a jugar, el lobby se cerrará y se abrirá el **Juego**.

Lobby

La primera acción a realizar es realizar la conexión del cliente con el servidor. Esto se hace rellenando el hostname y el puerto, posteriormente apretando el botón de conexión.



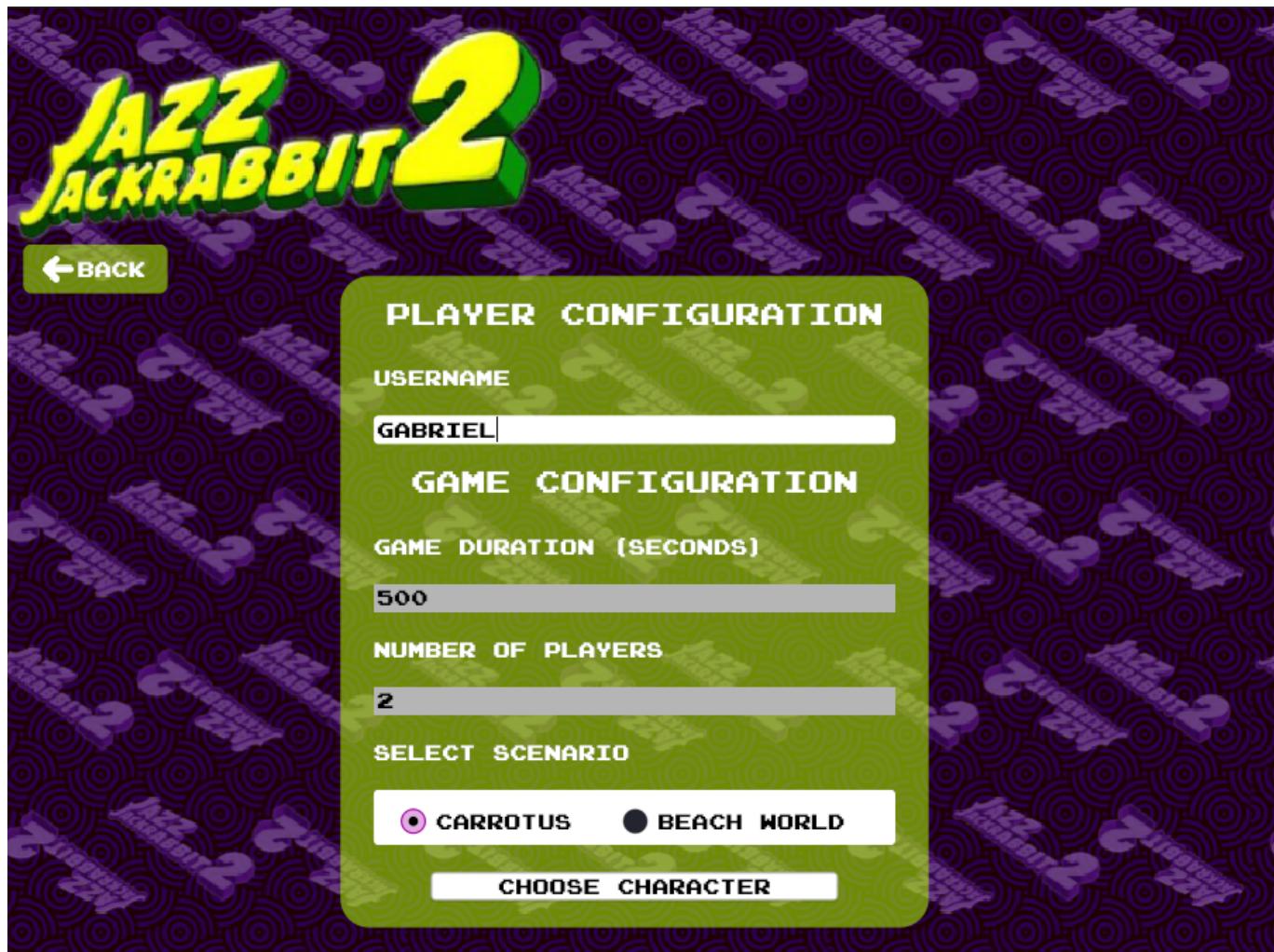
Si se habilitan los botones de la ventana inicial la conexión fue exitosa. Por defecto el servidor espera una conexión en localhost, así que en el campo de hostname poner `localhost`, y luego el puerto correspondiente.



Todos los campos de texto tienen validación de input así que si un dato se puso de manera incorrecta o la conexión falla, se mostrará un mensaje con lo que está ocurriendo.

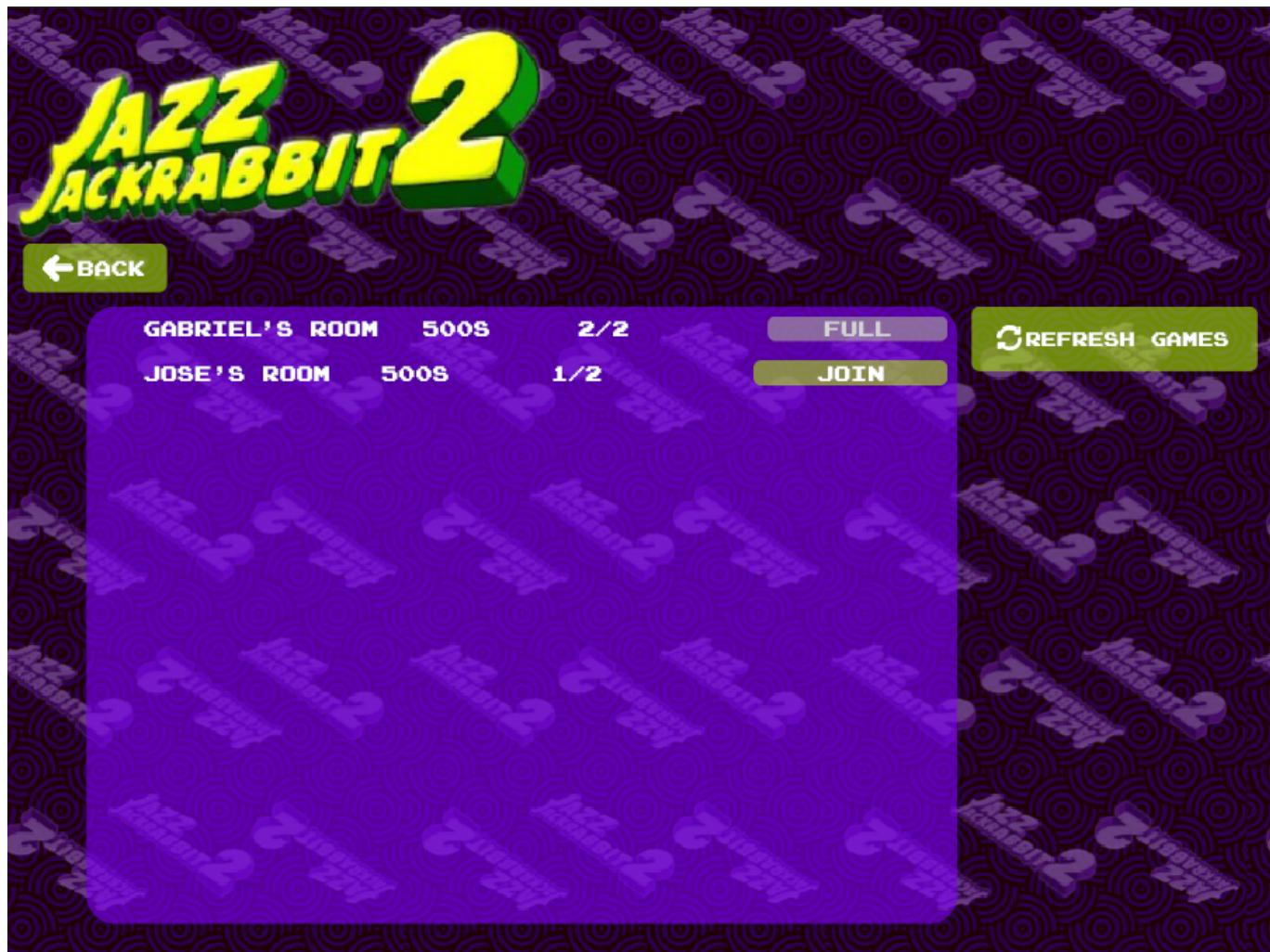
En la ventana inicial podemos elegir entre crear una partida o unirse a una de las partidas ya creadas.

Si creamos una nueva partida debemos ingresar nuestro nombre de usuario, luego las configuraciones de una partida constan de la duración de la misma y la cantidad de jugadores. Las mismas no se modifican desde la interfaz gráfica y están definidas en el archivo de configuración del juego, ubicado en el root del proyecto y usado en todos los componentes del juego llamado `games_config.yaml`. Además se puede seleccionar el escenario en que se quiere jugar, ya sea Carrotus o Beach World. Luego se selecciona el personaje que uno quiera utilizar y se quedará en la ventana de espera hasta que se conecten todos los jugadores y se llene la partida.





Si se seleccionó ingresar a un juego ya existente, la partida se visualizará con el nombre del usuario que la creó. Hay un botón para refrescar las partidas que están en vivo en el momento. Se ingresa a la partida deseada con el botón **Join** correspondiente, a menos que esté llena en cuyo caso se verá como **Full**.



Luego se verán las opciones del juego y se debe ingresar el nombre de usuario. Una vez completado ese paso se procede a la pantalla de espera.

Juego

Cuando todos los jugadores entraron a la partida, la ventana del Lobby se cierra y se abre el Juego.



Los controles son los siguientes:

- **FLECHA DERECHA**: mueve el jugador a la derecha
- **FLECHA IZQUIERDA**: mueve el jugador a la izquierda
- **FLECHA ARRIBA**: ejecuta la acción de salto
- **BARRA ESPACIADORA**: ejecuta la acción de disparo
- **SHIFT IZQUIERDO**: ejecuta la acción de correr mientras se mantenga apretado, el jugador ya debe estar en movimiento para comenzar a correr
- **CONTROL IZQUIERDO**: ejecuta el ataque especial del personaje seleccionado
- **NÚMERO 1**: selecciona el arma inicial
- **NÚMERO 2**: selecciona el arma secundaria
- **ESCAPE**: cierra el juego
- **F1**: muestra el panel de información debug, como los FPS
- **F2**: ejecuta el primer cheat, que resetea los coleccionables
- **F3**: ejecuta el segundo cheat, que finaliza la partida

La cámara del juego sigue al personaje que el usuario está controlando. Un jugador puede salir de la partida en todo momento presionando **ESCAPE** o la cruz de cierre de la ventana. Las partidas con múltiples jugadores finalizan automáticamente al momento de que un jugador miembro de la partida salga, señalando que un jugador se desconectó.



Una vez que el tiempo llega a cero, el juego termina y se muestra la leaderboard final con los nombres de los jugadores.



HUD

En el centro sobre el borde izquierdo de la pantalla se visualiza el juego, mientras que en el borde derecho se encuentra el HUD (heads-up display), donde se puede visualizar información sobre la partida y sobre el estado del jugador.



En orden desde arriba hacia abajo se observa:

- El tiempo restante de la partida. Cuando llegue a cero la partida finalizará y se verá el leaderboard final.
- El username del jugador.
- Los puntos de jugador al lado del ícono del personaje que eligió.
- El nombre del personaje que eligió para jugar.
- Los puntos de vida restantes sobre los puntos de vida totales.
- El arma principal (Blaster), que muestra una munición de 999 porque tiene munición ilimitada. Tendrá un recuadro naranja sobre el ícono del arma si el arma está seleccionada.
- El arma secundaria (Orb), junto con la munición restante. Tendrá un recuadro naranja sobre el ícono del arma si el arma está seleccionada.
- Un resumen del top 3 de cómo va el leaderboard.

Acerca del juego

Se trata de un remake del juego JazzJackRabbit 2 multijugador con modalidad "Arcade" que tiene como objetivo obtener la mayor cantidad posible de puntos antes de que se acabe el tiempo, enfrentándose a enemigos y recolectando ítems. Se puede matar a otros jugadores, quienes tienen vidas ilimitadas pero un tiempo personalizable de respawn.

Items

El mapa posee diferentes ítems que el jugador recolecta al pararse sobre ellos. Cuenta con monedas y gemas que otorgan puntos, zanahorias que curan, zanahorias envenenadas que intoxican al jugador por unos segundos y no curan (las cuales se ven igual a las zanahorias no envenenadas), y munición para el arma secundaria, la cual solamente se obtiene al eliminar enemigos.

Al recoger una zanahoria envenenadas el jugador no puede disparar durante el tiempo que dure el efecto, el jugador se puede dar cuenta porque el personaje tiene una animación de caminata "tonta" y le salen burbujas.

Enemigos

Existen 3 tipos diferentes de enemigos que inflingen daño al estar cerca de ellos. Al morir, sueltan una zanahoria o municiones del arma secundaria en base a una probabilidad. Su daño, vida, probabilidades de dropeo, cuántos puntos dan y tiempo de reapación son configurables. Los enemigos disponibles son: Bubba, Turtle Goon y Schwarzenguard.

Armas

Todos los jugadores cuentan con 2 armas. La principal, el Blaster, tiene munición infinita mientras que para la secundaria, el Orb, se debe recolectar munición. Ambas poseen diferente daño, tiempo de recarga y velocidad de proyectil, que son personalizables.

Personajes

Los personajes disponibles para jugar son Jazz, Spaz y Lori. Todos comparten las mismas habilidades (saltar, disparar, correr) pero cada personaje tiene un ataque especial. Jazz ejecuta un salto vertical que no le permite moverse lateralmente, Spaz puede realizar una patada que la desplaza de forma lateral, sin poder saltar hacia arriba, y Lori puede ejecutar un patada voladora de corto alcance mientras da un salto. Todos los ataques especiales realizan daño en ráfaga con todo lo que toquen.

Gameplay

Se dejan dos videos de gameplay, uno con un jugador y otro con dos jugadores, embebidos desde Youtube:

