

Taller de Programación I (75.42/95.08)

Trabajo Práctico Final

Manual de Usuario

Apellido y Nombre	Padrón
Inneo Veiga, Sebastian Bento	100998
Martinez Sastre, Gonzalo Gabriel	102321
Riedel, Nicolas Agustín	102130
Zbogar, Ezequiel	102216

Instalación	2
Requerimientos de Software	2
Para todas las aplicaciones	2
Sólo para el servidor	2
Sólo para el cliente	2
Sólo para el editor	2
Requerimientos de Hardware	3
Servidor	3
Cliente	3
Editor	3
Proceso de instalación	4
Instalación Automática	4
Instalación Manual	4
Configuración	5
config.yaml	5
map.yaml	10
Forma de uso	13
Servidor/server	13
Cliente/Wolfenstein3D	13
HUD	14
Editor	15
Apéndice de errores	16

Instalación

Requerimientos de Software

Para todas las aplicaciones

- Sistema operativo Linux Debian-based 64 bits.
- gcc/g++
 - gcc/g++ versión 9.3 o mejor.
- make
 - make versión 4.0 o mejor.
- cmake
 - cmake versión 3.15 o mejor.
- yaml-cpp
 - libyaml-cpp-dev versión 0.6.2 o mejor.

Sólo para el servidor

- lua
 - liblua5.3-0:amd64

Sólo para el cliente

- SDL2
 - libSDL2-dev, libSDL2-image, libSDL2-mixer, libSDL2-ttf
- qt5-default
 - qt5-default:amd64 versión 5.12 o mejor.

Sólo para el editor

- qt5-default
 - qt5-default:amd64 versión 5.12 o mejor.

Requerimientos de Hardware

Todas las aplicaciones deberían correr bien en cualquier PC/Notebook “moderno”. Son aproximados y puede que las aplicaciones corran en una computadora con menos requisitos.

Servidor

- Espacio en disco ~10MB
- RAM: ~10MB

Cliente

- Espacio en disco ~100MB
- RAM: ~10MB

Editor

- Espacio en disco ~10MB
- RAM: ~10MB

Proceso de instalación

Los siguientes pasos sirven para instalar el servidor, cliente y el editor.

Instalación Automática

1. Descargar los paquetes .deb de del repositorio (último release).
2. Ejecutar `sudo dpkg -i [file.deb]` para instalar los paquetes deseados.
 - a. file.deb puede ser W3DServer.deb, W3DClient.deb o W3DEditor.deb.

Para desinstalar un paquete ejecutar en terminal `sudo dpkg --remove [package-name]`.

Instalación Manual

1. Descargar el código del [repositorio](#)
 - a. Ya sea como zip y descomprimirlo o `git clone [link]` desde la terminal.
2. Dentro de la carpeta donde se encuentran los archivos del repositorio, crear una carpeta nueva (normalmente llamada build).
 - a. Esto para instalar tanto el **servidor** como el **cliente**. En caso de querer instalar el **editor** crear la carpeta dentro de editor_src ya que utiliza otro CMakeLists.
3. Moverse dentro de la carpeta build.
4. Correr por terminal `cmake ..`
 - a. Se verificarán las dependencias necesarias para compilar los programas.
5. Para instalar los 2 programas correr `make` por terminal.
 - a. Si se quiere instalar sólo el servidor correr `make server`.
 - b. Si se quiere instalar sólo el cliente correr `make Wolfenstein3D`.

Para desinstalarlo simplemente borrar los ejecutables

Configuración

Sólo el servidor hace uso de archivos de configuración. Los archivos de configuración están en formato *yaml*. El archivo puede tener tanto la extensión *.yaml* como *.yml*.

config.yaml

Se provee un archivo de configuración en formato *yaml* que contiene valores para balancear el juego, en la carpeta *server_src*, como ejemplo. **Todos los campos deben existir** dado que el archivo se utiliza para todas las partidas. La ruta del archivo se provee por línea de comando al momento de ejecutar el servidor. A continuación se explica cada bloque del archivo:

```
port: 8080
```

port es el puerto por el que escuchará el servidor.

```
tick_rate: 128
```

tick_rate representa la cantidad de veces que se actualiza el servidor por segundo.

```
weapons_general:
  machine_gun:
    time_between_shots: &machine_gun_shots 0.1
    time_between_bursts: &machine_gun_bursts 0.3
    bullets_per_burst: &machine_gun_bullets 5
  chain_gun:
    time_between_shots: &chain_gun_shots 1
  rocket_launcher:
    move_speed: &rpg_move_speed 2.5
    size: &rpg_size 0.2
    max_damage: &rpg_max_damage 100
    max_damage_distance: &rpg_max_damage_distance 3
```

Propiedades de las armas:

- `time_between_shots` es el tiempo que debe pasar entre un tiro y otro al mantener presionada la tecla de disparo.
- `time_between_bursts` es el tiempo que debe pasar entre una ráfaga y otra dado que el arma dispara de a varias balas a la vez.

- `bullets_per_burst` es la cantidad de balas por ráfaga.
- `move_speed` es la velocidad a la que viaja el rocket (cohete) al ser disparado.
- `size` es el tamaño del item `rocket_launcher`.
- `max_damage` es el daño máximo que puede causar el rocket (cohete).
- `max_damage_distance` es la distancia máxima en la que el rocket (cohete) hará daño medido en distancia euclidiana.

```
max_health: &max_health_global 100
```

Vida máxima de todos los jugadores.

```
#Player stats
player:
  #Speed
  move_speed: 0.02
  rot_speed: 0.02
```

Desde acá empiezan las propiedades del jugador. Es la misma para todos.

- `move_speed` es la velocidad de movimiento.
- `rot_speed` es la velocidad al girar hacia la izquierda o derecha.

```
#Collider
size: 0.2
```

- `size` es el tamaño del jugador, se usa para las colisiones con los objetos, paredes y puertas.

```
#Initial stats
#Cambiar por número en caso de ser necesario/para tests.
initial_health: *max_health_global
initial_score: 0
total_lives: 5
```

- `initial_health` es la vida inicial, por defecto es igual a la vida máxima, se puede cambiar por un valor menor.
- `initial_score` es la puntuación inicial, dejar en 0.
- `total_lives` son las vidas totales.

```
#Key
  key:
    radius: &key_radius 0.2
```

radius es el radio del item key.

```
#Initial weapons
weapons:
  initial_weapon: 1
```

- initial_weapon es el identificador del arma con el que se inicia, usar 0 para iniciar con knife (cuchillo) o 1 para pistol (pistola).

```
bullet:
  initial_bullets: 100
  max_bullets: &max_bullets 100
  radius: &bullet_radius 0.2
```

Propiedades de las balas.

- initial_bullets es la cantidad de balas iniciales, las balas son las mismas para todas las armas.
- texture es el id de la textura que se utiliza, no modificar.
- radius es el radio del ítem bala al pasar por arriba del mismo.

```
knife:
  time_between_stabs: 1
  range: 0.5
```

Propiedades del knife (cuchillo):

- time_between_stabs es el tiempo entre un ataque y otro.
- range es el alcance del cuchillo.

```
bot_path: "../server_src/Bots/Destripador.lua"
```

Ruta al archivo del bot. Se pueden agregar bots siempre que cumplan con las mismas firmas que Destripador.lua pero sólo 1 por servidor.


```
#Items
items:
```

Desde acá serán las propiedades de los ítems. `radius` representa el radio del ítem al pasar por encima del mismo.

```
#Treasure
cross:
  value: 10
  radius: 0.5
crown:
  value: 200
  radius: 0.5
cup:
  value: 50
  radius: 0.5
chest:
  value: 100
  radius: 0.5
```

Los treasures (tesoros) son los items que dan puntos, hay 3 tipos: `cross` son las cruces, `crown` son las coronas, `cup` son las copas y `chest` son las cajas de tesoro.

- `value` es la cantidad de puntos que dan cada tesoro.

```
#Health recover
food:
  value: 10
  radius: 0.4
  max_health: *max_health_global
medical_kit:
  value: 20
  radius: 0.2
  max_health: *max_health_global
blood:
  value: 1
  radius: 0.3
  max_health: 11
```

Los health recover son los objetos que recuperan vida del jugador, hay 3 tipos: `food` es la comida, `medical_kit` son los kit médicos, `blood` es la sangre.

- value es la cantidad de vida que recupera el jugador al pasar por encima del objeto.
- max_health es la vida desde la cual ya no se puede recuperar.

```
#Otros
key:
  radius: *key_radius
bullet:
  value: 5
  radius: *bullet_radius
  max_bullets: *max_bullets
```

key es la llave, su radio es el mismo que el de más arriba.

bullet representa las balas. Su value es la cantidad de balas que recupera.

- max_bullets es la cantidad máxima de balas que se puede tener.

```
#Weapons
machine_gun:
  radius: 0.3
  time_between_shots: *machine_gun_shots
  time_between_bursts: *machine_gun_bursts
  bullets_per_burst: *machine_gun_bullets
chain_gun:
  radius: 0.3
  time_between_shots: *chain_gun_shots
rocket_launcher:
  radius: 0.3
  move_speed: *rpg_move_speed
  size: *rpg_size
  max_damage: *rpg_max_damage
  max_damage_distance: *rpg_max_damage_distance
```

Los parámetros aquí son los mismos que explicados más arriba.

map.yaml

El servidor también se encarga de levantar los mapas. Estos deben estar en formato *yaml* dentro de la carpeta *server_src/Maps*. Se proveen algunos mapas de ejemplo.

Se recomienda utilizar el editor para generar estos archivos dado que automatiza el proceso de creación y se garantiza el correcto formato del mismo.

```
map_name: "Soy el mapa, soy el mapa..."
```

`map_name` es el nombre del mapa, sirve como identificador.

```
max_players: 3
```

`max_players` es la cantidad máxima de jugadores en el mapa.

```
#Map
length: 17 #Arriba a abajo
width: 14 #Izquierda a derecha
map:
  - [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
  - [1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1]
  - [1, 0, 0, 0, 0, 1, 1, 2, 1, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1]
  - [1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1]
  - [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

`length` y `width` son la cantidad de filas y columnas de la matriz que representa el mapa.

`map` es el mapa en sí. Los valores que puede tener son:

- 0 para espacio vacío.

- 1 para pared naranja.
- 2 para pared gris con musgo.
- 10 para puerta con llave (debe estar entre 2 paredes).
- 11 para pasadizo (estar entre 2 paredes).
- 12 para puertas automáticas (debe estar entre 2 paredes).

```
#Players
player_1:
  pos_x: 2.0
  pos_y: 2.0
  dir_x: 1
  dir_y: 0

player_2:
  pos_x: 3.0
  pos_y: 3.0
  dir_x: 1
  dir_y: 0

player_3:
  pos_x: 5.0
  pos_y: 5.0
  dir_x: 1
  dir_y: 0
```

Cada player (jugador) estará posicionado en una posición inicial con una dirección. Deben haber al menos max_players.

- pos_x y pos_y representan las posiciones iniciales de cada jugador en la matriz (aunque pueden ser valores con decimales).
- dir_x y dir_y representan la dirección en la que apunta el jugador al iniciar la partida.

```
#Items
items:
  #Treasures
  cup:
    cant_cup: 1
    cup_0:
      pos_x: 15
      pos_y: 1
  #Food
```

```

food:
  cant_food: 1
  food_0:
    pos_x: 5
    pos_y: 5
#Blood
blood:
  cant_blood: 1
  blood_0:
    pos_x: 1.0
    pos_y: 13.0
#Keys
key:
  cant_key: 1
  key_0:
    pos_x: 15
    pos_y: 2
#Bullets
bullet:
  cant_bullet: 1
  bullet_0:
    pos_x: 2
    pos_y: 12
#Weapons
chain_gun:
  cant_chain_gun: 2
  chain_gun_0:
    pos_x: 8
    pos_y: 8
  chain_gun_1:
    pos_x: 9
    pos_y: 9

```

En el caso de los ítems **no es necesario que estén todos**. En todos los casos se debe indicar la cantidad de ítems (`cant_nombre_del_item`). Los mismos deben estar numerados desde el 0. El nombre del ítem debe coincidir con el nombre en el apartado de `items` del archivo *config.yaml*.

Forma de uso

Servidor/server

Para ejecutar el servidor ejecutar por terminal:

- En caso de instalación manual
 - `./serverW3D` [ruta al archivo de configuración]
- En caso de instalación automática ubicarse en la carpeta donde está instalado `/usr/games/Wolfenstein3D/server/executable`
 - `./serverW3D` [ruta al archivo de configuración]

Una vez ejecutado ya se pueden conectar los clientes a su IP (o la IP del servidor al que haya subido la aplicación) y el puerto indicado en el archivo de configuración pasado por parámetro.

Cliente/Wolfenstein3D

Para ejecutar el cliente:

- En caso de instalación manual
 - `./Wolfenstein3D` en la terminal donde está instalado.
- En caso de instalación automática
 - Correr con el launcher en las aplicaciones o ejecutar desde la carpeta donde está instalado:
`/usr/games/Wolfenstein3D/client/executable`

Una vez ejecutado siga las instrucciones en pantalla.

HUD



- Score: Puntos.
- Lives: Vidas.
- HP: Puntos de vida.
- Imagen de llave: Indica si se tiene llave o no. Sólo puede tenerse 1 llave a la vez.
- Imagen de arma: Indica el arma que tiene, el número indica las balas. Las balas se comparten entre todas las armas pero no todas las armas consumen la misma cantidad.

Controles

Tecla	Ejecución
W / ↑	Mover hacia adelante
A / ←	Girar a la izquierda
S / ↓	Mover hacia atrás
D / →	Girar a la derecha
Left / Right Control / Espacio	Disparar
F	Interactuar con puerta/pasadizo
Q	Cambiar a arma anterior
E	Cambiar a arma siguiente

Editor

Para ejecutar el Editor ejecutar por terminal:

- En caso de instalación manual
 - `./Editor`
- En caso de instalación automática
 - Buscar entre las aplicaciones del sistema operativo con el nombre de "Editor".

Una vez Ejecutado seguir la clara interfaz gráfica, la dinámica consiste en agarrar ítems de la lista de la izquierda y arrastrarlos sobre el mapa de la derecha.

Es posible tanto abrir Mapas para editarlos, como guardar mapas generados desde 0 en el Editor.

Apéndice de errores

- Del lado del servidor si se imprime “Out of Range” es porque algún jugador está fuera de rango, verifique que los jugadores estén dentro de los parámetros del mapa.
- Al crear un mapa manualmente, procure que los ítems no estén al lado de una pared porque se verán la mitad del sprite. Lo mismo con los jugadores pero estos podrían no moverse.