Jazz Jackrabbit 2
User manual

Contents

Operating system and necessary dependencies

The required operating system is Ubuntu 22.04 LTS, using ISO C++17, libSDL2pp version 0.18.1, and yaml-cpp version 0.8.0.

To install the needed dependencies for libsdl2 run:
`sudo apt install libsdl2-dev libsdl2-image-dev libsdl2-ttf-dev`

Change game configuration settings

To set the attributes of the assignment task, you have a config.yaml file.

Using the 'player' attribute, you can set the number of players per game. For multiplayer games, adjust this attribute to specify the desired number of players for the session. Additionally, you have the flexibility to customize the health and damage parameters for both enemies and the player character. Furthermore, adjust the drop probability of specific items from enemies according to your preferences. Moreover, you can establish the duration of the match and specify the point value granted by collected coins.

Project building

To build the project run:
git submodule init && git submodule update
mkdir build
cd build
cmake ..
make

Once the project has been built, move the config.yaml file to the build folder.

Project installing and files deploy

The steps to install the game and deploy the configuration file are:

Compiling the Project: First, compile the project in the build folder to generate the necessary executables and binaries.

```
s@S:~/Desktop/tallerTPfinal/TP_TALLER/build$ make
[ 24%] Built target yaml-cpp
[ 42%] Built target SDL2pp
[ 72%] Built target Client
Consolidate compiler generated dependencies of target Server
[ 73%] Building CXX object CMakeFiles/Server.dir/src/server/config.cpp.o
[ 73%] Linking CXX executable Server
[ 86%] Built target Server
[ 88%] Built target gtest
[ 89%] Built target gtest_main
[ 92%] Built target serde_test
[ 94%] Built target yaml-cpp-sandbox
[ 96%] Built target yaml-cpp-parse
[ 97%] Built target yaml-cpp-read
[ 99%] Built target gmock
[100%] Built target gmock_main
```

Installing Binary Files: After compiling, proceed to install the binary files into their designated directory, /usr/local/bin.

```
s@S:~/Desktop/tallerTPfinal/TP_TALLER/build$ sudo cp Client Server /usr/local/bin
```

Setting Up Configuration Files: If the specified configuration folder (/etc/TP_TALLER/, for instance) does not exist, create it as part of the installation process. Then, install the configuration files into this newly created directory.

```
s@S:~/Desktop/tallerTPfinal/TP_TALLER/build$ sudo mkdir -p /etc/TP_TALLER/
s@S:~/Desktop/tallerTPfinal/TP_TALLER/build$ sudo cp config.yaml /etc/TP_TALLER/
```

Navigating Out of the Current Directory: After setting up the configuration files, exit the current directory to ensure the subsequent installation of additional resources takes place outside of it.

```
s@S:~/Desktop/tallerTPfinal/TP_TALLER/build$ cd ..
```

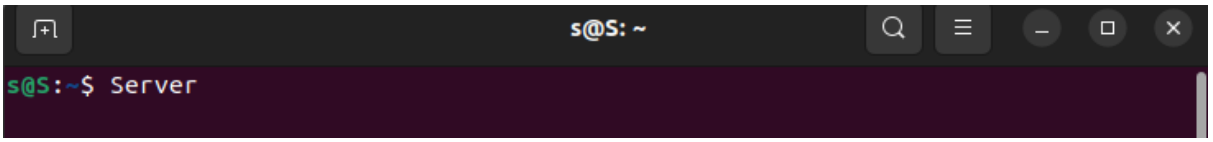
Installing Additional Resources: Finally, install any additional resources required by the application into their appropriate locations, often under /usr/share/TP_TALLER.

```
s@S:~/Desktop/tallerTPfinal/TP_TALLER$ sudo cp -r sources /usr/share/TP_TALLER/
```

Next, start the server, and then the client.

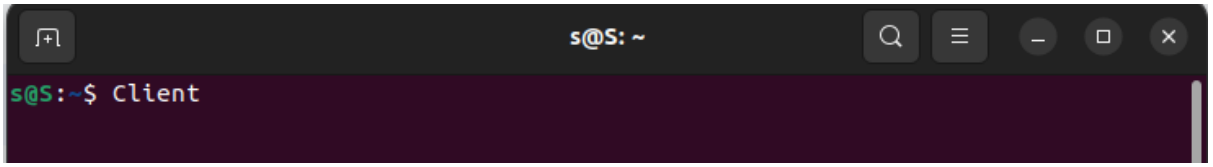
Starting the server

Run it from any location: Execute the binaries from any location in the terminal using their direct name, if they are in the PATH. So to start the server after installation run:



Starting the client

Run it from any location: Execute the binaries from any location in the terminal using their direct name, if they are in the PATH. So to start the client after installation run:



Stop server execution

To terminate the server execution, type the letter "q" and press Enter.

Game usage instructions

The following table provides instructions for using the keys to perform specific actions:

Key	Action
A	Move left
Shift-A	Move left faster
D	Move right
Shift-D	Move right faster

Space	Jump
P	Shoot
E	Switch guns
Space-S	Special Attack
Shift-F	Cheat refill ammo

Choosing characters and scenarios

Navigate using the right arrow key.

Selecting an option from the menu

Click on the menu’s word.

Level editor: form to create a level

By default, the first input box is selected. To change it, click on the other one.

To navigate within the input box, use the left or right arrow keys.

Delete one letter at a time with the backspace key.

Level editor

To move the camera horizontally or vertically, press the arrow keys.

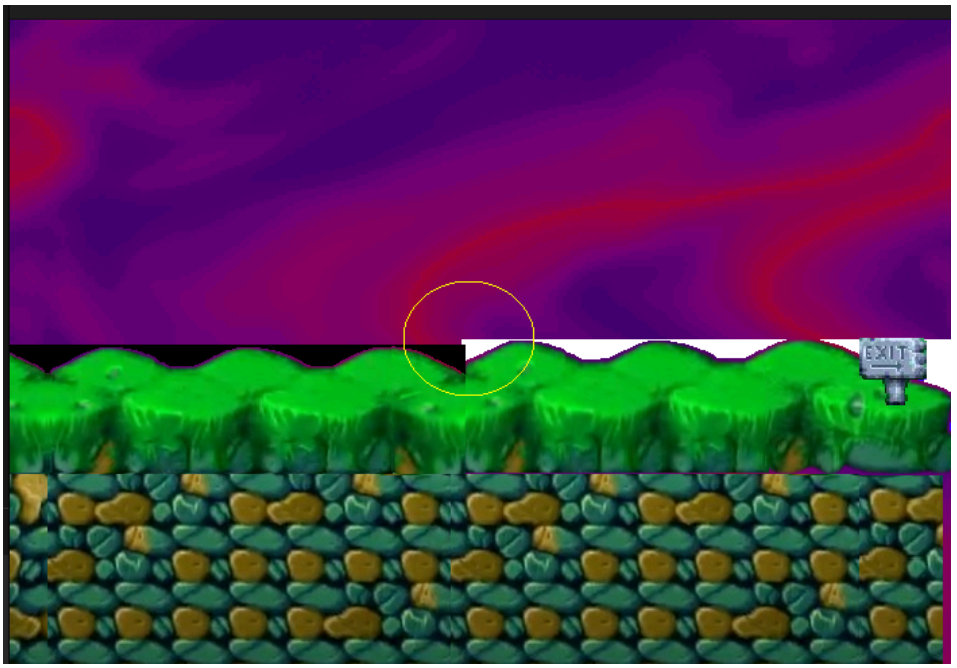
To delete an element, select it and click on the trash can icon.

Select an object by clicking on it, and click again to release the object at the cursor's position.

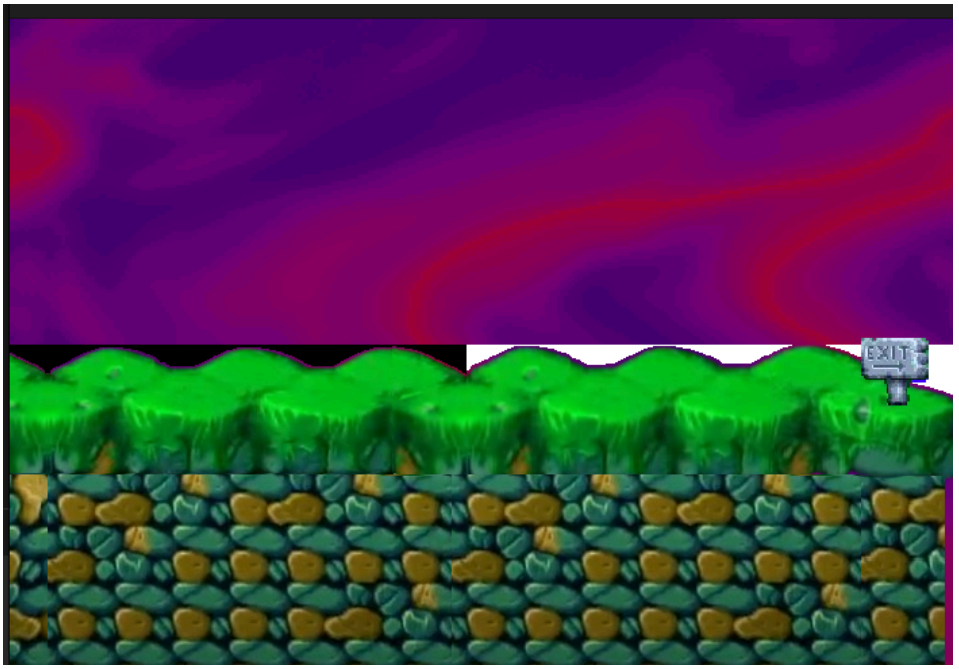
It is crucial to ensure that the various elements in the level editor palette are aligned. If not, due to the collision implementation, it could cause the character to get stuck.

The backgrounds of the elements have been painted white and black to illustrate the previous point. In the first image, we can observe how the element with a white background is positioned higher than the one with a black background. This serves

as an example of a scenario where the character might get stuck when attempting to move from left to right.



To address this issue, the second image demonstrates how the objects should be aligned to prevent collision problems.



Maps created for different scenarios

Two maps are presented for the Beach and Psychedelia scenarios.