

User Guide Mini Golf 2D

20 Mai 2018

User Guide

TABLE OF CONTENTS

1.0	INTRODUCTION	3
1.1	Mini Golf	4
1.2	Goal of Mini Golf	4
2.0	GETTING STARTED	5
2.1	System Requirements	6
2.1.1	Java	6
2.1.2	Windows	6
2.1.3	Linux	6
2.2	Minimum Hardware Requirements.....	6
2.2.1	CPU & GPU	6
2.2.2	Input	6
3.0	TUTORIAL	7
3.1	Starting the Game and Setting a Username	8
3.2	The Chat box features	12
3.2.2	Creating a lobby	12
3.2.3	Changing a lobby	12
3.2.4	Deleting a lobby	12
3.2.5	Changing the username	12
3.2.6	Whisper	12
3.2.7	Broadcast.....	12
3.2.8	Exiting	12
3.2.9	Ping status	12
3.5	The Game Window	13
3.4	The Score List	14
4.0	GAMEPLAY	15
4.1	Aim	16
4.2	Power	16
4.3	Environment Interaction	16
4.4	Camera.....	16
6.0	MULTIPLAYER INSTRUCTIONS	17
5.1	Multiplayer	18
7.0	OBSTACLES	19
7.1	Boost pad	20
7.2	Slow pad (Sand).....	20
7.3	Black hole.....	21
7.4	Portal	21
8.0	MAP EDITOR.....	22
8.1	Create a Map.....	23
8.3	Spawn Points.....	25
8.4	Scripts	25
8.5	Cosmetics	25
8.6	Cosmetic Instance.....	25

1.0 INTRODUCTION

1.1 Mini Golf

Mini Golf is a smaller version of Golf which is played on a smaller scale i.e. in a smaller area. It features obstacles like barriers and tunnels to hinder the Golf Ball and to make the game interesting.

1.2 Goal of Mini Golf

Complete each hole with as few strikes as possible.
The player with the lowest score, i.e. the number of strikes, wins the game.

2.0 GETTING STARTED

2.1 System Requirements

2.1.1 Java

Java Version 8.0 or higher

2.1.2 Windows

Windows Vista SP2 or higher

2.1.3 Linux

Most Linux Platforms

2.2 Minimum Hardware Requirements

2.2.1 CPU & GPU

- Processor: Dual Core or higher Core Processor
- Graphics card: Any NVidia or AMD Card with OpenGL Support
- RAM: 2GB or more

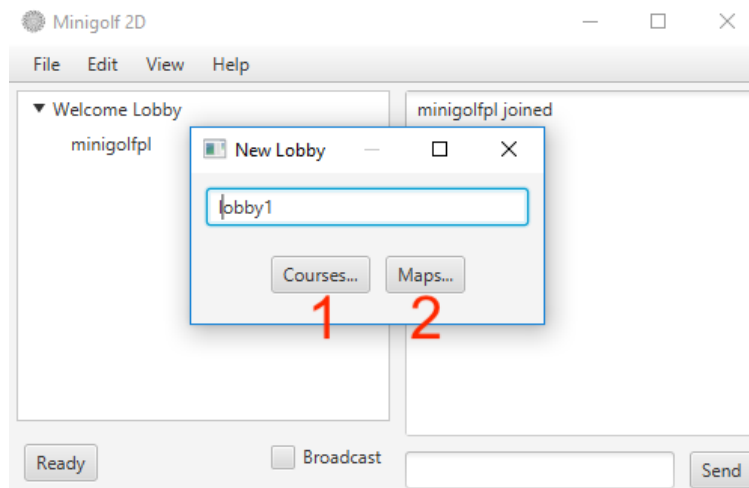
2.2.2 Input

- Mouse
- Keyboard

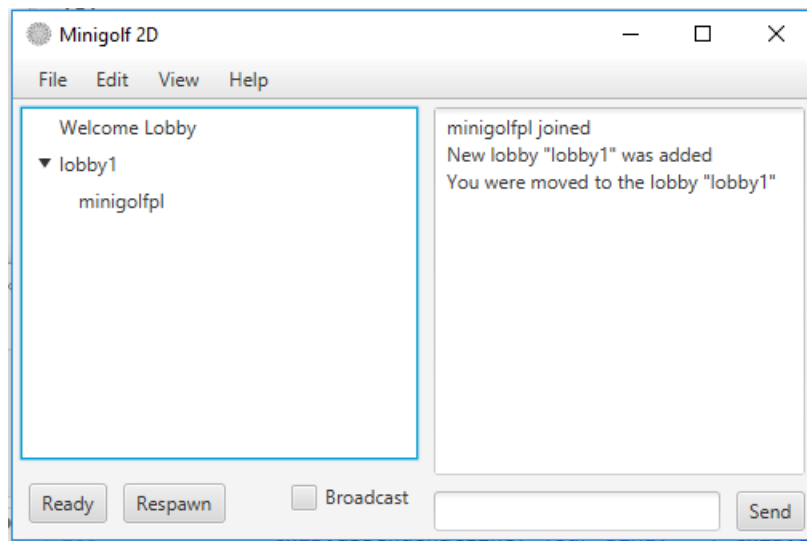
3.0 TUTORIAL

3.1 Starting the Game and Setting a Username

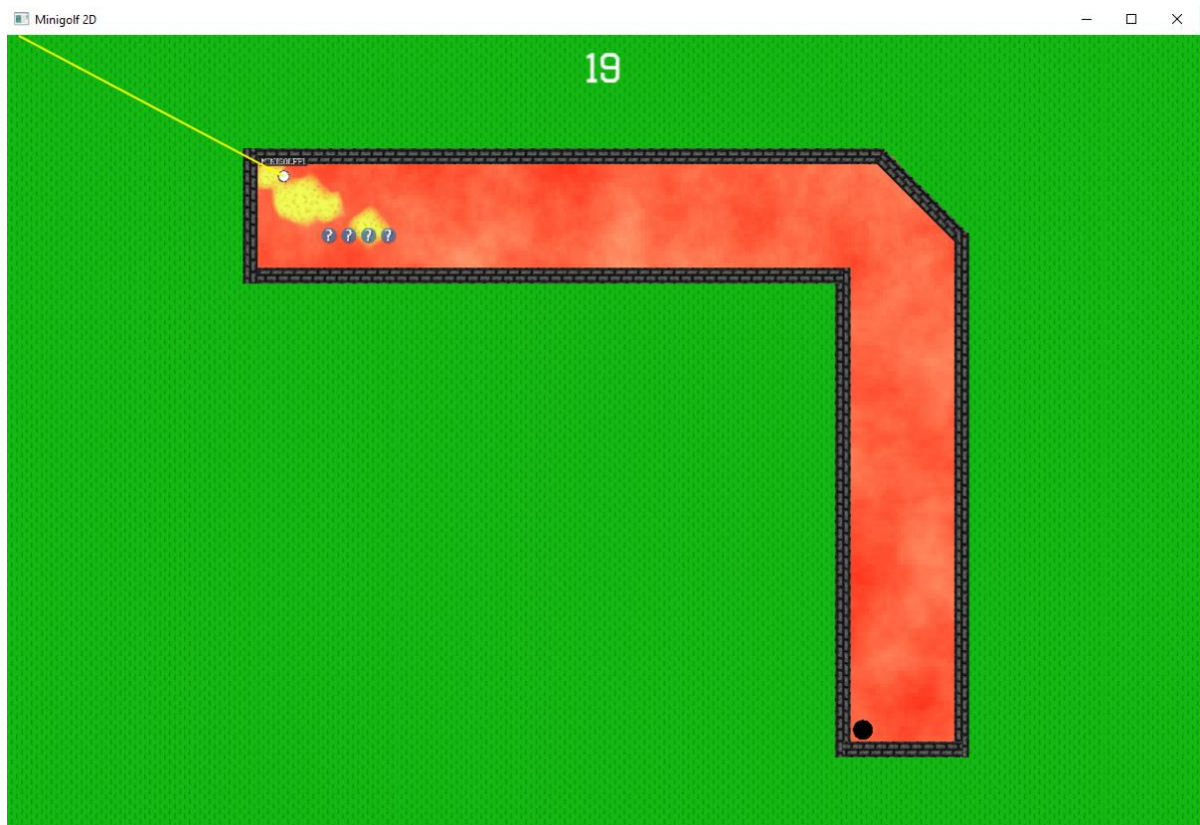
1. Navigate with CMD/Terminal to the directory where the game's JAR file is located
2. Run the following command in the CMD/Terminal: `java -jar MiniGolf2D.jar arg`, where MiniGolf2D.jar should be replaced by the game's JAR's file name.
3. The `arg` can either be `client <hostaddress>:<port> [username]` for the client (username is optional; if not specified the system's username will be used), or `server <port>` to launch the server.
4. Set up a server on the port 6000 by typing: `java -jar MiniGolf2D.jar server 6000`
5. Now to connect to the server as a client run the following: `java -jar MiniGolf2D.jar client localhost:6000`
6. Wait for the chat window to open
7. Create a new lobby by right click on lobby list – New Lobby...



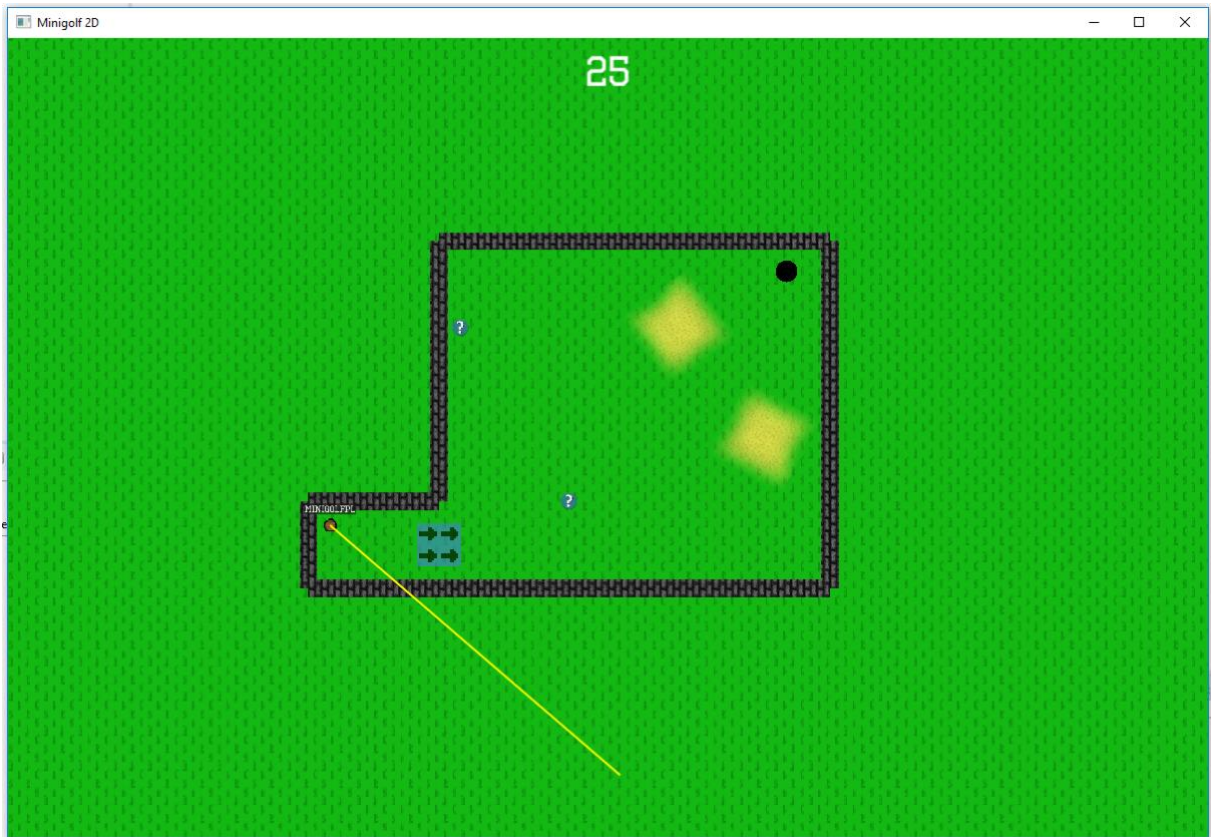
8. Type any name for the lobby that is not yet taken and select the predefined
1) Courses or select 2) Maps to play individual or more Maps



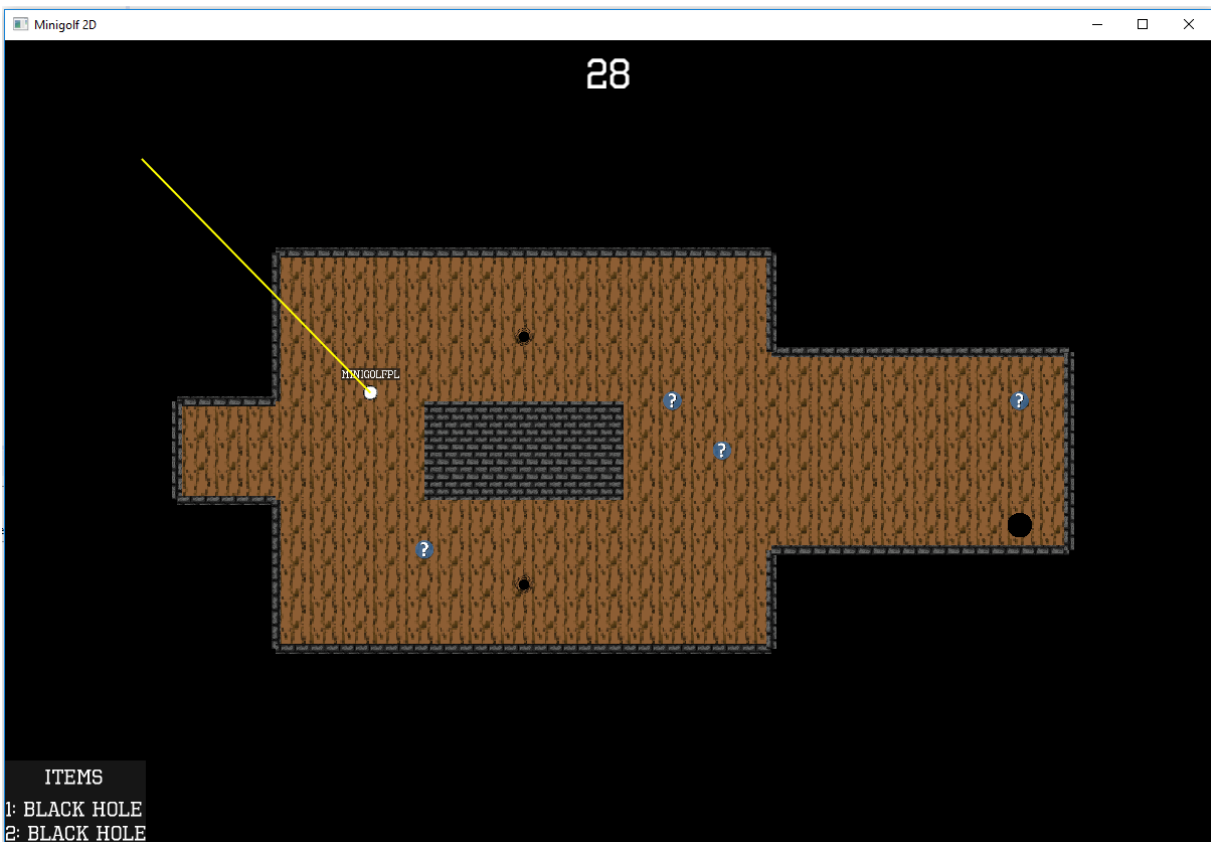
9. Click the Ready Button to start the game



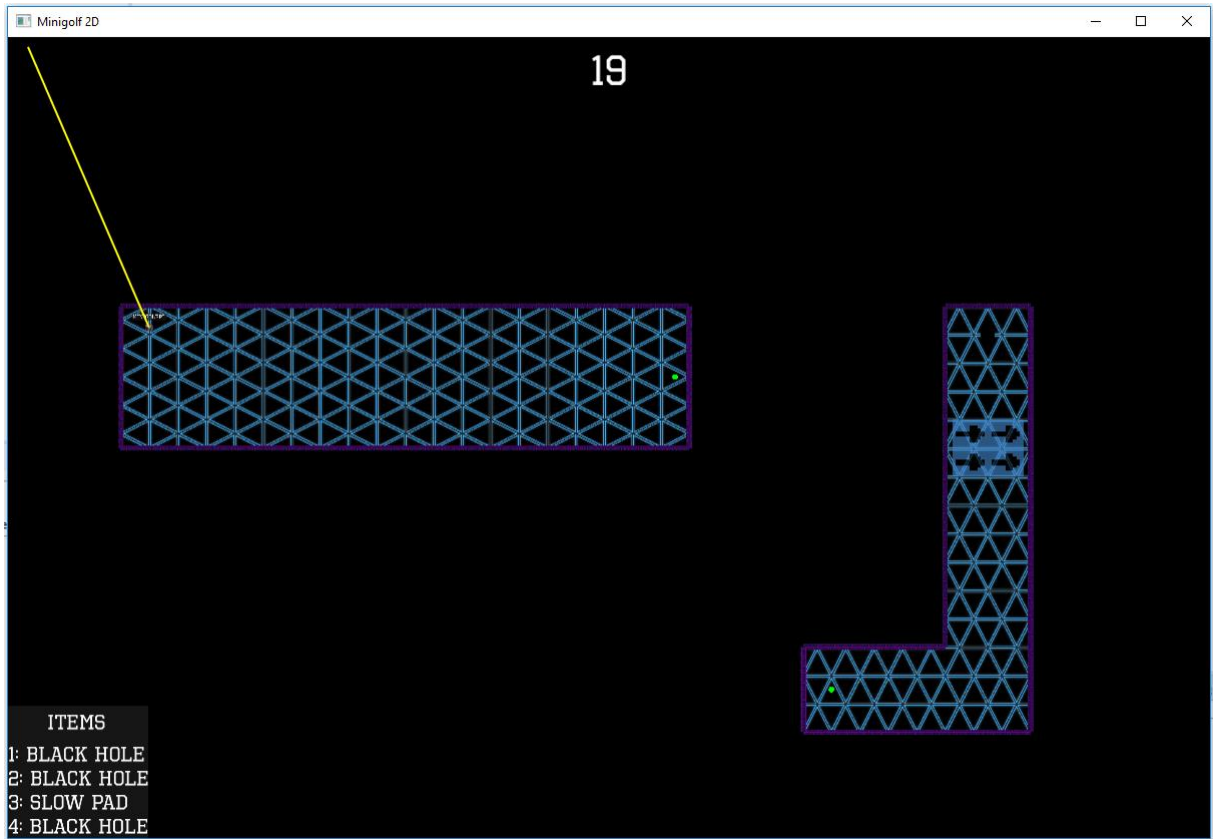
Map 1



Map 2



Map 3



Map 4

3.2 The Chat box features

3.2.2 Creating a lobby

To create a lobby

Right click on the Lobby list and click new Lobby

3.2.3 Changing a lobby

To change the lobby

Click on the preferred Lobby you want to change

3.2.4 Deleting a lobby

To delete a lobby

Right click on the lobby list and click delete

All players that were still in the lobby are moved to the "Welcome Lobby".

3.2.5 Changing the username

To change the username

Go to Edit – Change Username

3.2.6 Whisper

To send a message privately to a player so that it will
Right click on the lobby list and select whisper

3.2.7 Broadcast

To send a message across all lobbies to all players
Tick the Broadcast box

3.2.8 Exiting

To exit the chat and close the game

Type " :close" and you will be logged out from the chat

By exiting the chat, the player will also
automatically exit the Game

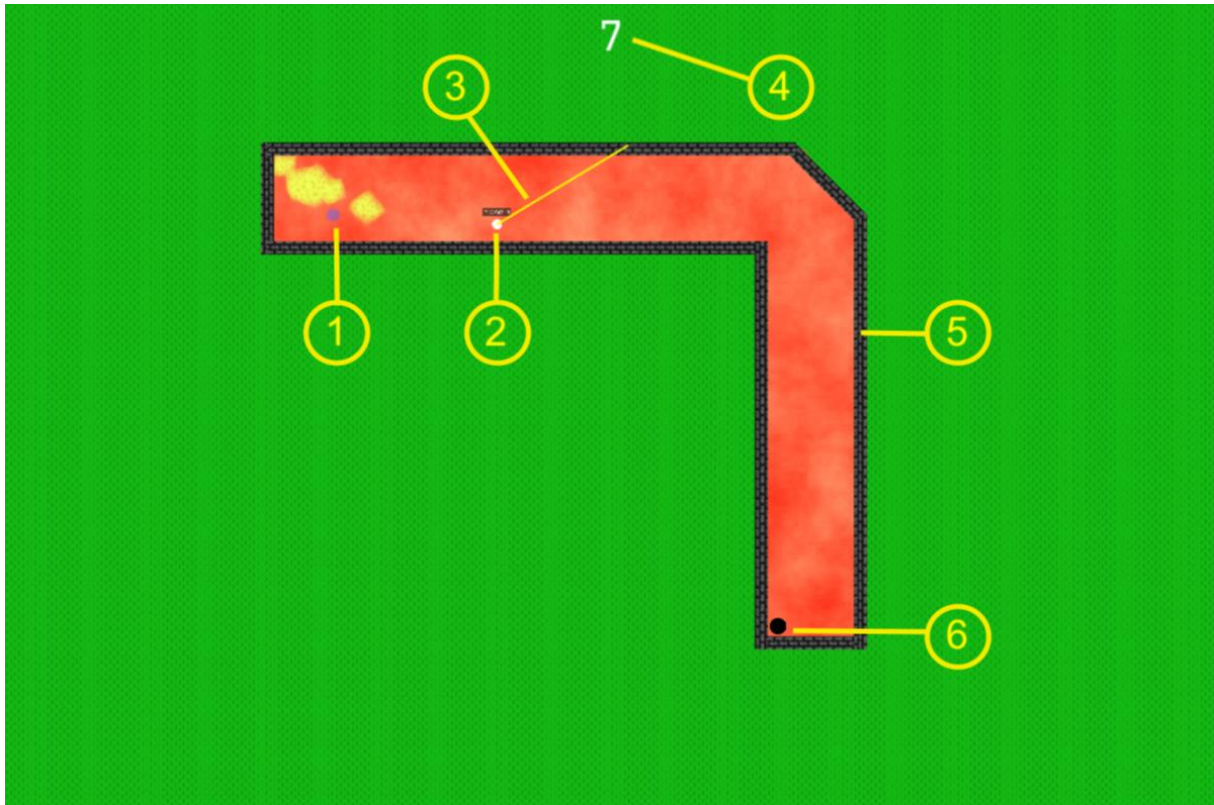
3.2.9 Ping status

To get the ping/the current roundtrip time to the server

Type " :ping"

3.5 The Game Window

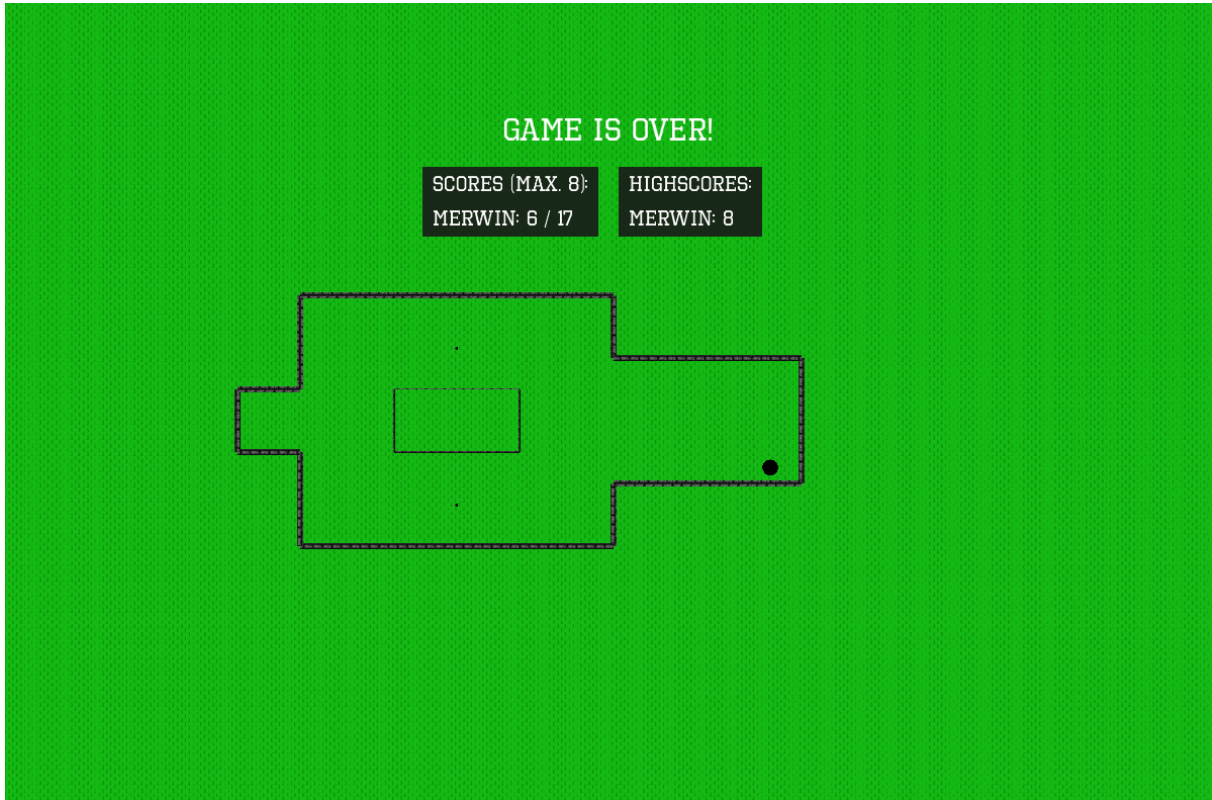
- 1) Item Pickup
- 2) Golf Ball
- 3) Assistive Line
- 4) Timer
- 5) Wall
- 6) Hole



3.4 The Score List

At the end of each game the players' scores are shown in the chat. Additionally, if a player has a new high score it is saved persistently. Each players' high score is also shown in the chat at the end of a game.

During a game the score list can be shown by pressing the TAB Key.



4.0 GAMEPLAY

4.1 Aim

The player can aim the ball with help of the assistive line pointing from the Golf ball to the mouse pointer. A left-click causes the player to strike the ball.

4.2 Power

The player can choose various strike strengths. The longer the line between the golf ball and the mouse pointer, the stronger the strike is and the further the ball will roll.

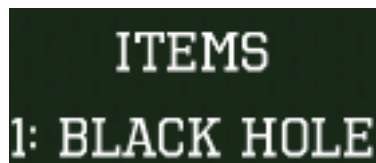
4.3 Environment Interaction

Players can place different types of items in the map to decrease the opponents chance of winning or to gain an advantage.

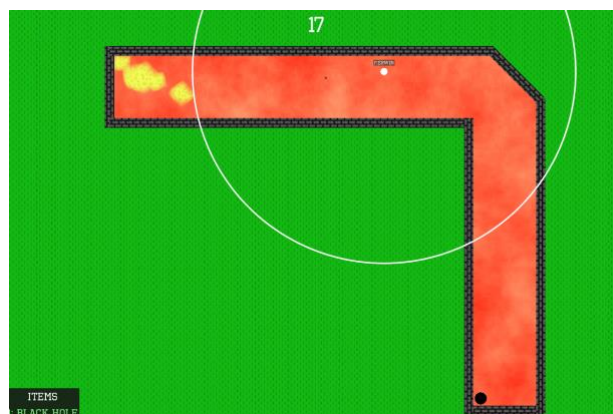
Before placing the obstacles, players must collect the items. Items can be collected by shooting the ball at Item Pickups.



Once the player has collected an item, he can press the number designated to the item as shown in the bottom left of the game window. Here for example it's 1 for a Black Hole.



The player can choose a position to place the object within a radius around the golf ball and the circle as shown below:



The objects can be rotated by scrolling with the scroll wheel.

4.4 Camera

The camera can be moved around by pressing and holding right-click and moving the mouse pointer. Additionally, by scrolling with the scroll wheel the zoom can be changed.

6.0 MULTIPLAYER INSTRUCTIONS

5.1 *Multiplayer*

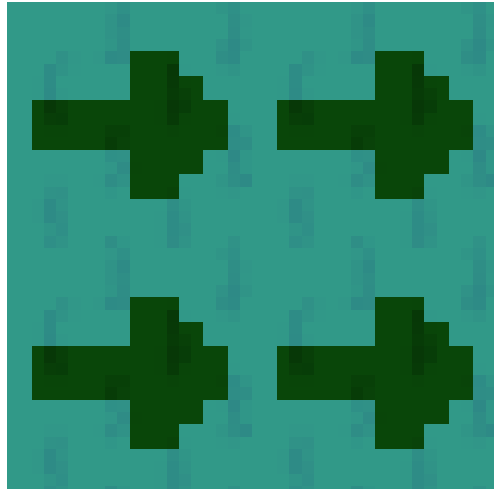
- 1) If lobbies are available in the lobby list, the player can select the desired lobby and press the “Connect” button to join the lobby.
- 2) By clicking the “Ready” button, the player can start a game with other players in the same lobby. All players must be ready to start the game.



Players must be connected to the same network and lobby to be able to play together

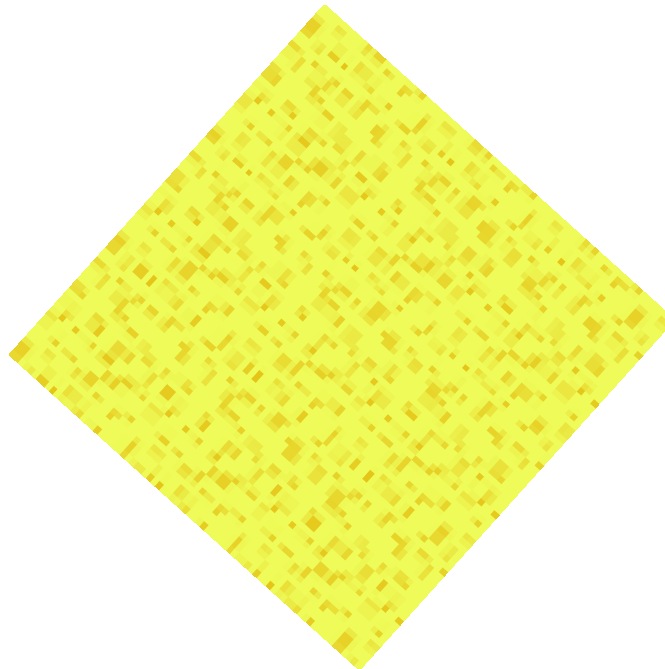
7.0 OBSTACLES

7.1 *Boost pad*



Boost the speed of the ball in the direction the arrows point when the ball enters the area.

7.2 *Slow pad (Sand)*



Quickly slows the speed of the ball when the ball enters the area.

7.3 Black hole



Pulls the ball towards the black hole due to its gravitational force and once the ball falls into the hole it respawns the ball at its starting position.

7.4 Portal



Teleports the ball from one portal to the other portal. Portals are always one way unless the map was specifically designed to allow entering them both ways.

8.0 MAP EDITOR

8.1 Create a Map

A map can contain unlimited type of Entities at any X, Y coordinates. The map format is explained in the next section.

To create a map follow these instructions:

- 1) Open your preferred text editor
- 2) On each line you can write an entity type and its X, Y coordinates. You can find each entity type and format in the section below. If you are not sure how to do it look the example code below
- 3) Make sure Wall and Hole entities as well as spawn points are at least present. If not the game will likely not work as intended.
- 4) Once you are done save the File as a *.map file in the "data/maps/" folder.

```
# Wall args: x, y, width, height, rotation
entity: wall(0, 400, 40, 800, 0)
entity: wall(600, 0, 1200, 40, 0)
entity: wall(1200, 400, 40, 800, 0)
entity: wall(600, 740, 1200, 40, 0.1)

# Wall next to hole
entity: wall(900, 600, 400, 40, -0.1)

# Slow- and Boostpad args: x, y, width, height, rotation
entity: slowpad(600, 450, 140, 140, 0.2)
entity: boostpad(250, 280, 180, 180, 2.5f)

# Hole args: x, y
entity: hole(1100, 700)

# BlackHole args: x, y
entity: blackhole(800, 50)

# PortalIn args: x, y, target portal name, [uses]
entity: portalin(1000, 100, test_portal, 2)
entity: portalin(850, 200, test_portal_2)

# PortalOut args: x, y, portal name
entity: portaloout(100, 200, test_portal)
entity: portaloout(200, 300, test_portal_2)

# Add spawn points. Args: x, y, [weight]
# The weight determines how likely it is for that spawn point
# to be chosen
spawn: 50, 50, 4
spawn: 85, 50, 3
spawn: 120, 50, 2
spawn: 155, 50, 1
```

8.2 Entity Format

- wall(<x, double>, <y, double>, <width, double>, <height, double>, <rotation, double>)
Creates a Wall with the midpoint (x, y) with a width, height and rotation in radians.

- wall_(<start.x, double>, <start.y, double>, <end.x, double>, <end.y, double>, <width, double>)
Creates a Wall with a start and endpoint (start.x, start.y)/(end.x, end,y) and a width.
- hole(<x, double>, <y, double>)
Creates a Hole with the midpoint (x,y)
- slowpad(<x, double>, <y, double>, <width, double>, <height, double>, <rotation, double>)
Creates a “Slow Pad” which reduces the velocity of the golf ball. Midpoint (x,y), width, height and rotation in radians.
- boostpad(<x, double>, <y, double>, <width, double>, <height, double>, <rotation, double>)
Creates a Boost pad which increases the velocity of the golf ball in the direction the arrows point.
- blackhole(<x, double>, <y, double>)
Creates a Black hole with the midpoint (x,y). The ball gets absorbed by the black hole and gets respawned at the origin position
- portalin(<x, double>, <y, double>, <target portal name, String>, [uses, int])
Creates an entrance portal with midpoint (x,y). The outgoing portal should be declared in target portal name. “usage” defines how often the portal can be used before it disappears.
- portalout(<x, double>, <y, double>, <name, String>)
Creates an outgoing portal with midpoint (x,y). The name should contain the portal’s name.
- pickupspawner((<x, double>, <y, double>, <entity_name, Entity>, <min_round, int>, <max rounds, int>, <uses, int>, <spawn initially, boolean>)
Creates a pickupspawner with midpoint (x,y). The collected item’s name should be in entity_name. The min and max rounds defines in how many rounds the pickupspawner is shown again. To spawn the pickup initially set “spawn initially” to true.

8.3 *Spawn Points*

Spawn points are used to set starting position for the golf ball when players are ready to start the game.

The map the must contain at least 4 spawn points.

- spawn: <x, double>, <y, double>, [weight, float]
Creates a spawn point with the midpoint (x,y) and with an optional priority weight. The spawn points are chosen randomly depending on the weight.

8.4 *Scripts*

JavaScripts can be added to mod and extend the game.

The scripts must be added to the “data/scripts/” folder.

- script: <script file name, String>
Adds the script to the map. The name of the script should be in script file name, example: my_script.js

8.5 *Cosmetics*

Cosmetics are cosmetic objects which can be added to the Map. They cannot interact with the entities.

Their main purpose is to beautify the game. Cosmetics define some reusable properties for Cosmetic instances and can be used by multiple instances at once.

- cosmetic: textured(<name, String>, <png texture file, String>, <nearest neighbor sampling, boolean>, <repeat texture, boolean>)
Creates a texture with a name which can be added to the map. The .png file can be set with “png texture file”. The boolean “nearest neighbor sampling” and repeat texture define if “nearest neighbor sampling” is used for the texture and if it should repeat itself.

8.6 *Cosmetic Instance*

A Cosmetic instance should be added to the map for presenting cosmetic instances. The instance defines where and how the cosmetic should be rendered.

- cosmeticinstance: textured(<cosmetic instance, String>, <x, double>, <y, double>, <width, double>, <height, double>, <rotation, double>, <pixels per tile, double>)
Renders a textured cosmetic with the name in “cosmetic instance”.
(x, y), width, height and rotations define the position, width, height and rotation of the texture.
Pixel per tile defines how large the texture should be in pixel.