

## ----- PORTFOLIO 2 -----

Student numbers:

- s341873
- s341879
- s341857
- s341844

For the Portfolio 2 the group have chosen **task 2** from the assignment. We have created a snake game, based on the classic game, but with an extra feature - lasers/gun. Therefore, our game is called *Snakes with Guns*.

## Development

### Installing dependencies

Running

```
npm install
```

will install dependencies listed in `dependencies` and `devDependencies` in package.json

### Format code

We use [Prettier](#) to format JavaScript, CSS and HTML files. Running

```
npm run format
```

will run prettier and format the code.

### Features and languages in the program

The languages and styling that are used in this project are the following:

- Javascript/Typescript
- Node
- HTML
- CSS

### Running the program

#### From the terminal

If you want to run the program from the terminal, open a new terminal window with the path of the project and make sure you are in the root of the project (*Portfolio2\_DATA2410* in this case).

To install dependencies you need to have node installed, and run the following in your terminal:

```
$ npm install
```

This will install all dependencies listed in `package.json`.

After installing dependencies you need to build the server and frontend applications.

We use [Webpack](#) for building and bundling our frontend app. We chose to use Webpack because it can process import/export statements in our code. Webpack creates one single javascript file as the entry point for our frontend-app.

We also use [TypeScript](#) to add typings to our JavaScript. We found it easier to discover potential bugs and work together if we had a type system to help us.

After these dependencies have been installed, continue in your current terminal window and run the following commands to build the apps:

```
$ npm run build
```

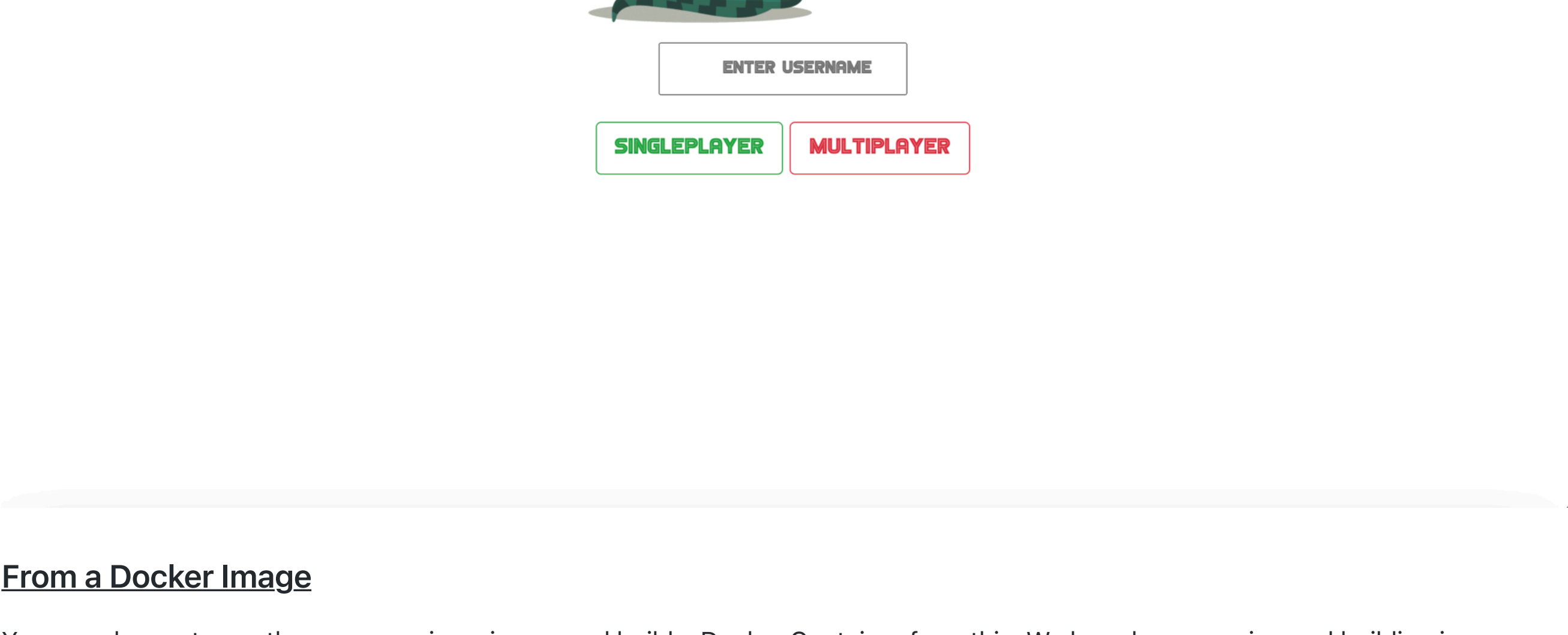
Then you can start the server locally using:

```
$ npm run start
```

This will start the server and make the app (server) available to serve HTTP requests. The terminal should show this message:

```
Server started on port 3000!
```

You can open a browser window with this link: 127.0.0.1:3000 and the program should be up and running and looking like this:



#### From a Docker Image

You can choose to run the program via an image and build a Docker Container from this. We have been running and building images and docker container with *Docker Desktop*, which is compatible with both Windows, Mac and Linux.

1. Open a terminal window and make sure you are in the root of the project folder (Portfolio2\_DATA2410). Proceed to run the following commands:

```
$ docker build -t image_name .
```

*image\_name* is a name you can choose. Give it some time, and when the build-process is finished your terminal window should look something like this:

```
[+] Building 88.8s (5/10)
=> [internal] load build definition from Dockerfile
=> => transfering dockerfile: 597B
=> [internal] load .dockerignore
=> => transferring context: 148B
[+] Building 89.0s (5/10)
=> [1/5] FROM docker.io/library/node:16-alpine@sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263811
=> => resolve docker.io/library/node:16-alpine@sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263811
=> => sha256:2a658af977f6d31f7487c6f54d65875770274428b604c949bd54e4f439a083f5 1.16kB / 1.16kB
=> => sha256:72ca8e2f26fa0f3384989bc175ae6eb322fb33afdae8a7b6129bda752d95ca411 6.73kB / 6.73kB
=> => sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263817f4a4 1.43kB / 1.43kB
[+] Building 89.3s (5/10)
=> [1/5] FROM docker.io/library/node:16-alpine@sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263811
[+] Building 89.4s (5/10)
=> [1/5] FROM docker.io/library/node:16-alpine@sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263811
=> => sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263817f4a4 1.43kB / 1.43kB
[+] Building 199.1s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 597B
=> [internal] load .dockerignore
=> => transferring context: 148B
[auth] library/node:pull token for registry-1.docker.io
=> [1/5] FROM docker.io/library/node:16-alpine@sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263811
=> => resolve docker.io/library/node:16-alpine@sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263811
=> => sha256:2a658af977f6d31f7487c6f54d65875770274428b604c949bd54e4f439a083f5 1.16kB / 1.16kB
=> => sha256:72ca8e2f26fa0f3384989bc175ae6eb322fb33afdae8a7b6129bda752d95ca411 6.73kB / 6.73kB
=> => sha256:572689dd24a48fb0058c7fe92229108cc47f428be5fec36ec367e8263817f4a4 1.43kB / 1.43kB
=> => sha256:4f9832ab4c84eeeebe0f731d77832bf516f6eadbba3e95982549488e61c1898f 35.83MB / 35.83MB
=> => sha256:b660f9df4970674d8cfd489741fa49d6d0f9179d1b56a0f524212240598d6e1ce 2.35MB / 2.35MB
=> => sha256:feaa9ebc3631f4a358d1f4fd58ea4d622db86d0adb633cde0cc4f183407fc9 281B / 281B
=> => extracting sha256:4f9832ab4c84eeeebe0f731d77832bf516f6eadbba3e95982549488e61c1898f
=> => extracting sha256:b660f9df4970674d8cfd489741fa49d6d0f9179d1b56a0f524212240598d6e1ce
=> => extracting sha256:feaa9ebc3631f4a358d1f4fd58ea4d622db86d0adb633cde0cc4f183407fc9
[internal] load build context
=> => transferring context: 4.17MB
[2/5] WORKDIR /home/node/app
=> [3/5] COPY package*.json ./
=> [4/5] RUN npm install
=> [5/5] COPY .
=> => exporting to image
=> => exporting layers
=> => writing image sha256:8acbf15a091605c67c7e19bf72990b561914af6b755af0e55103b51ce95577d8
=> => naming to docker.io/library/portfolio2_data2410
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

2. To start a docker container from the image you have built, run the following command:

```
$ docker run -p 3000:3000 image_name
```

The terminal should show the following:

```
> portfolio2_data2410_snakeswithguns@1.0.0 start
> node server/server.ts

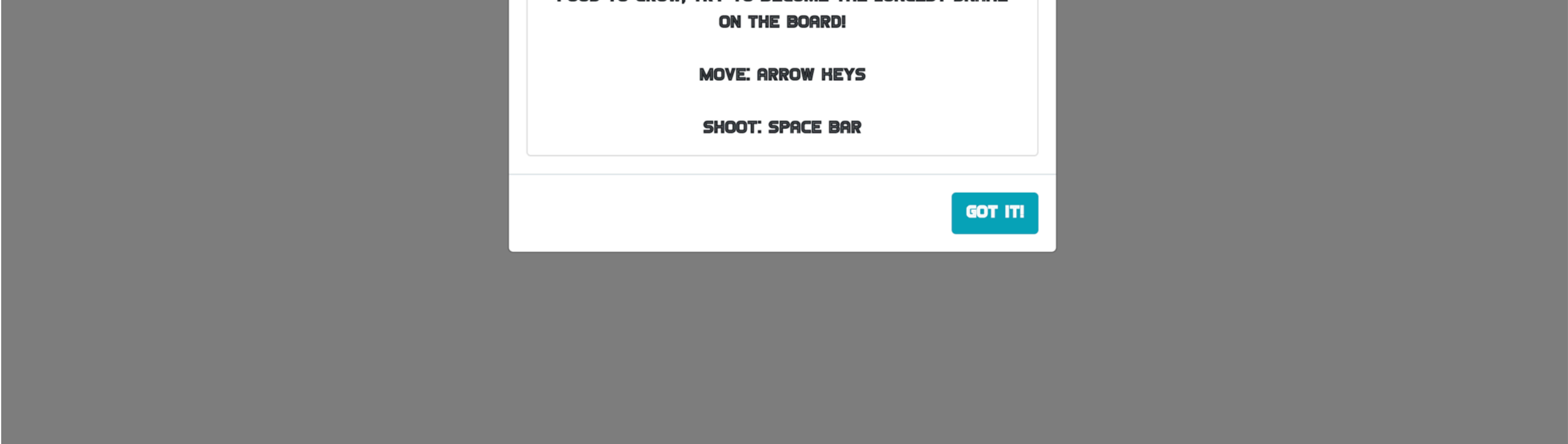
Server started on port 3000!
```

You can proceed into a browser window and type 127.0.0.1:3000, and you should see the same page as the picture of the browser above.

### The program: Expected outcome

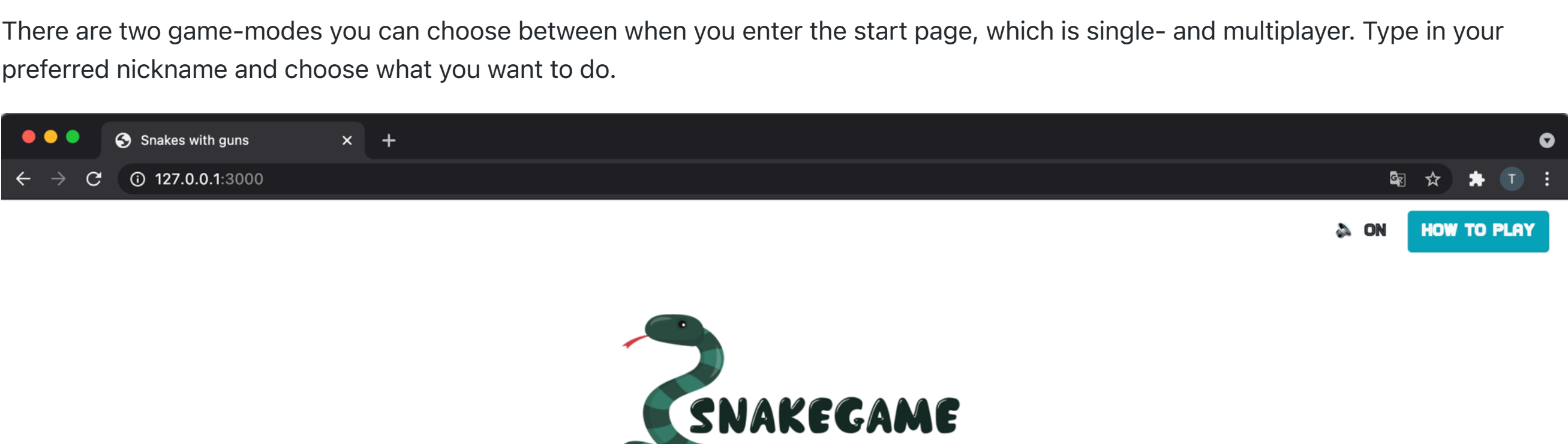
#### General information

In the top right corner of the start page there is a button that says *How to play*. Clicking this gives you some help and information about how the game works and which controllers you have to use.



To the left of the button, there is a speaker with the text *On* besides it. Clicking the speaker turns the music and sound effects in the game *off*. Both this and the *How to play* button can be clicked anytime.

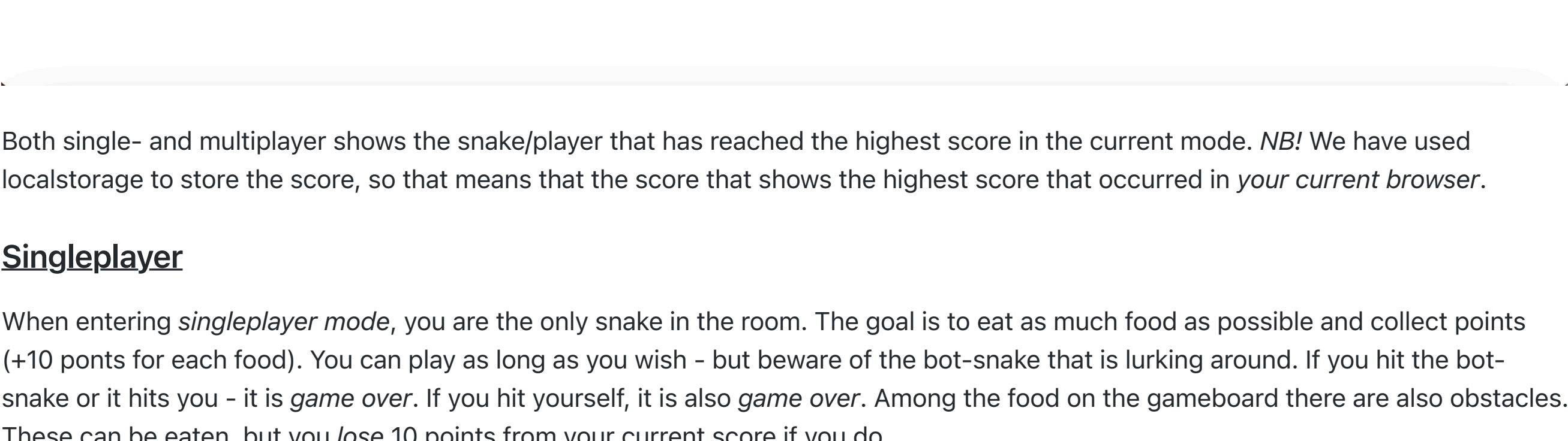
There are two game-modes you can choose between when you enter the start page, which is single- and multiplayer. Type in your preferred nickname and choose what you want to do.



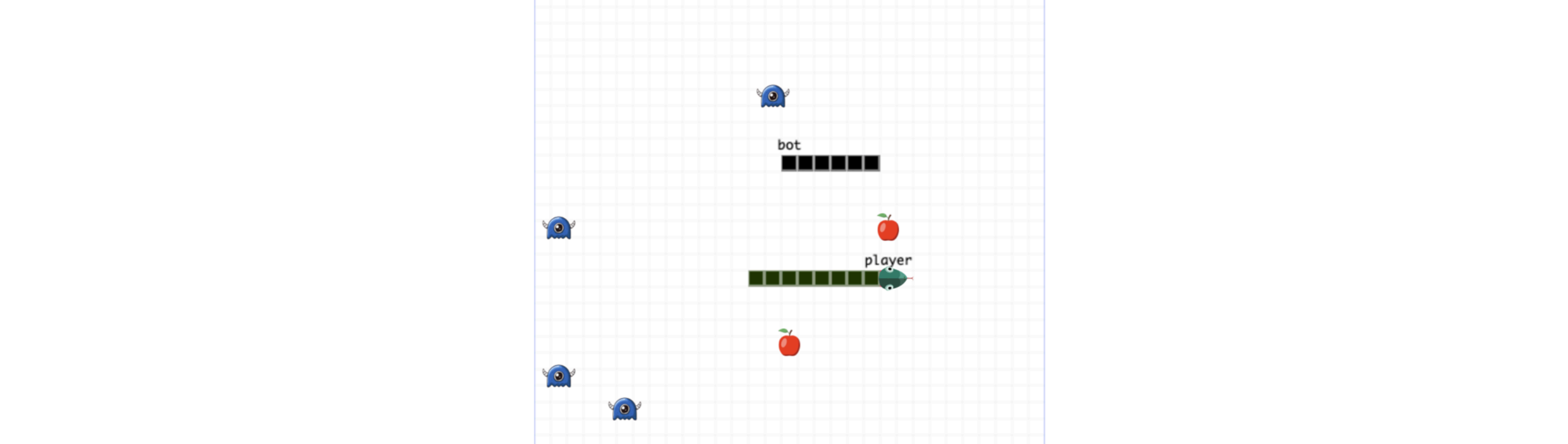
Both single- and multiplayer shows the snake/player that has reached the highest score in the current mode. *NB!* We have used localstorage to store the score, so that means that the score that shows the highest score that occurred in *your current browser*.

#### Singleplayer

When entering *singleplayer mode*, you are the only snake in the room. The goal is to eat as much food as possible and collect points (+10 points for each food). You can play as long as you wish - but beware of the bot-snake that is lurking around. If you hit the bot-snake or it hits you - it is *game over*. If you hit yourself, it is also *game over*. Among the food on the gameboard there are also obstacles. These can be eaten, but you lose 10 points from your current score if you do.



Here is a screenshot that shows the highscore in singlemode, and the current score of the one playing:



The game clearly tells you when the game is over for you:



If it is game over for you, you can simply refresh the page and play again. (Or hit play again if the button works :-))

#### Multiplayer

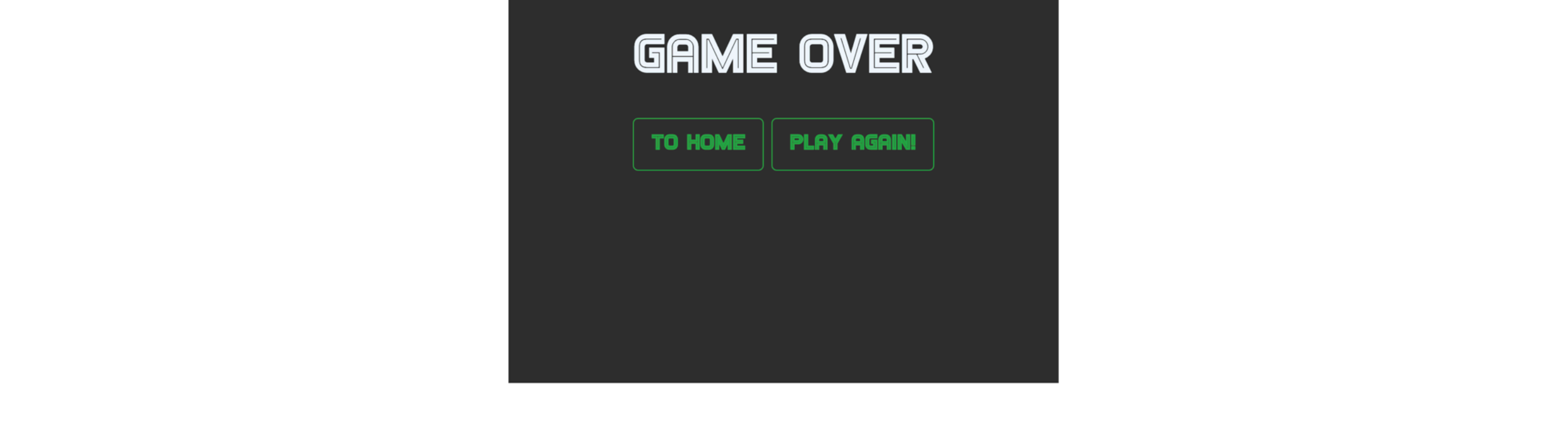
When entering *multiplayer mode*, you are presented with two options: create a room and start playing or join a room with a lobby-id. If a friend has already created a room, they can share this ID with you and you can join the same room and start playing together.

The gameplay is the same as in singleplayer-mode, except there are not any bots or obstacles other than the other players/snakes. If you run into yourself or any other snake/player - it's *game over*.

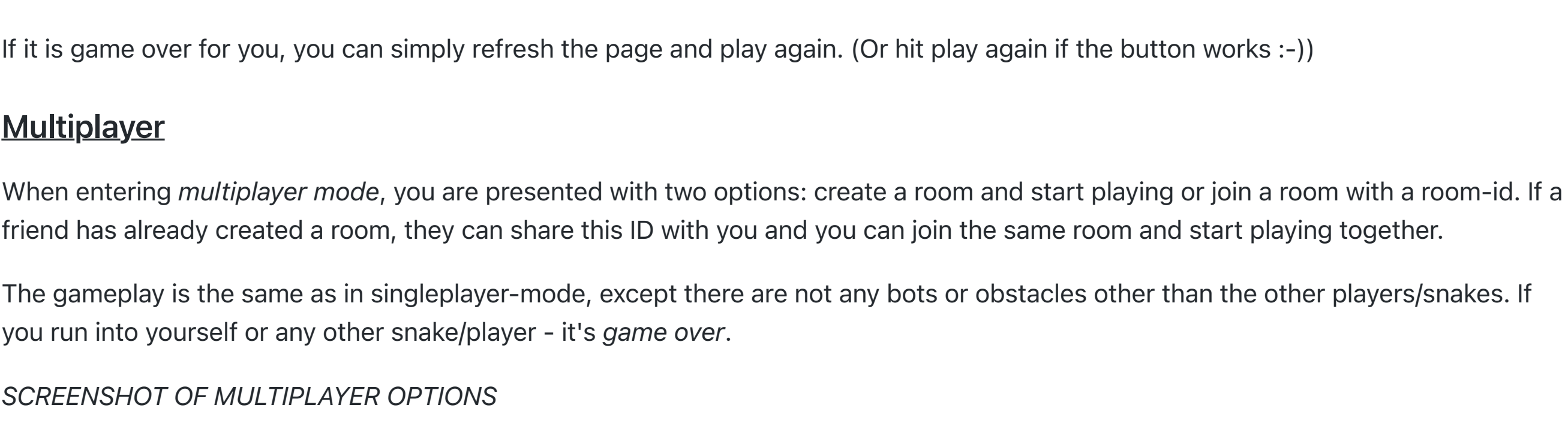
#### SCREENSHOT OF MULTIPLAYER OPTIONS

There must me at least two snakes in the room for the game to start. Once there are two snakes together in the room, the game starts. Maximum players in one room are 5. If a room is full, you are not able to enter.

Here is a screenshot of one player waiting, and it clearly says below the gameboard that we need at least two snakes to start - so we are waiting. The *lobby ID* can be seen to the right of the gameboard, and this can be shared with friends you want to join and play.



Once a second snake has entered, the game starts and the board is filled with food.



The overview of the connected players can be seen to the left.