

# Documentation for the final project

## Code Clicker

## Content

User documentation .....	3
Game information .....	3
Interacting with the program .....	3
Development documentation .....	7
Specification summary .....	7
Program structure .....	7
Possible extensions .....	8
Conclusion .....	8

## User documentation

### Game information

Code Clicker is a game inspired by Cookie Clicker, and hence falls into the idle gaming category.

The game has no definite goal. You essentially gain LOC to buy buildings and upgrades that grant you more LOC. However, due to the code limitations, you can “win” if your LOC count is high enough. The game then restarts.

There are nine buildings in total. You start with one and more appear as you gain LOC. Each of them adds a different amount of LOC each second and they all have their own unique upgrades. Upgrades appear as you gain LOC and buy buildings.

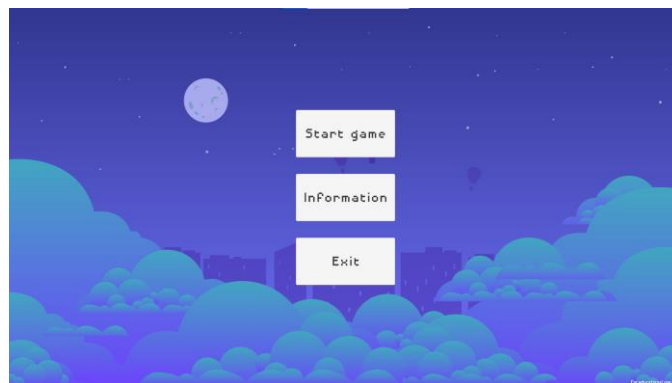
You can gain special boosts as well. Each time an energy drink appears on the screen, if you manage to click on it in time, it will grant you one of four available boosts.

Additionally, you can also collect achievements. As of right now, the achievements are only granted for clicking a certain number of times and buying a certain number of buildings.

The game supports multiplayer. You can choose whether you want to host a lobby or join one. Lobbies can be either public or private, in which case you’ll get a code through which players can connect. You can’t choose which public lobby you want to join; you will be added into one based on the capacity. The maximum number of players is four. Upon starting the game, all previous progress apart from achievements will be lost. Only the host can start the game. You receive notifications to let you know whether certain actions have been successful or not, for example whether you joined a lobby, when the game is started, etc.

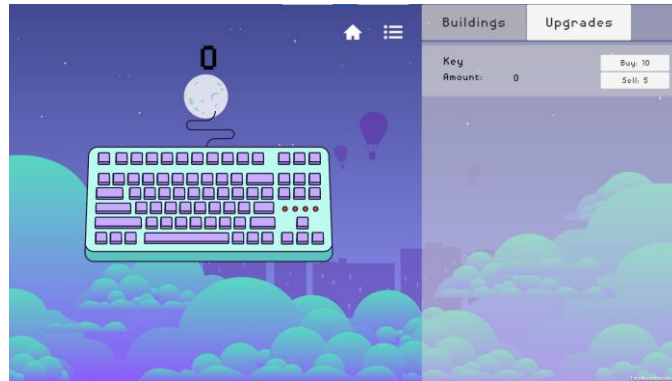
### Interacting with the program

Upon starting the game, you are met with the home screen, which offers you three options. If you’re not familiar with the game or would like to refresh your knowledge, you can read through the basic information after clicking on the information button. Otherwise, you have an option to start a new game or exit the application. The data will not be saved upon exiting and all progress will be lost, including achievements.

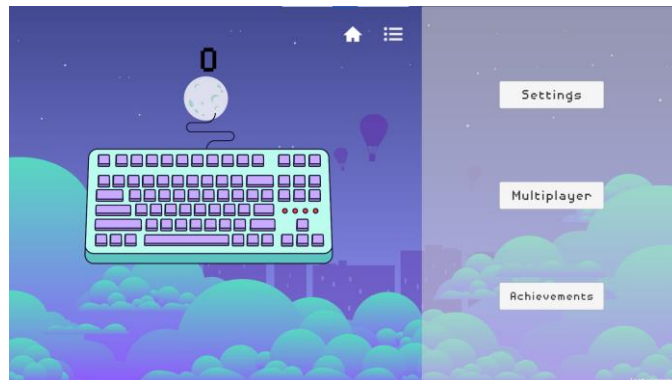


Even after starting the game, you can return to the home screen through the home button. However, your progress will be lost and only your achievements will be kept.

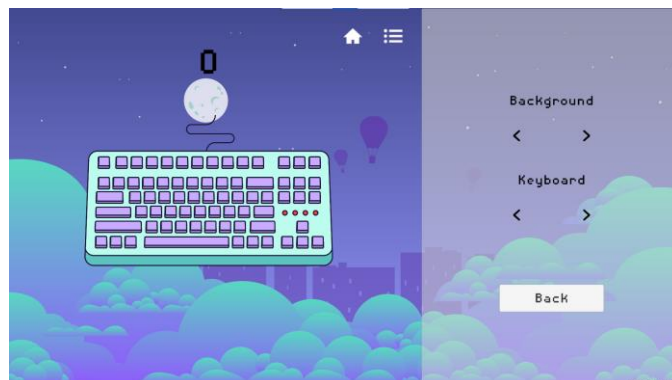
You gain LOC by clicking on the big keyboard on the left side of the screen. Buildings will appear in the panel on the right side of the screen. You can purchase them by clicking on the buy button or sell them by clicking on the sell button. Those actions will change the price. You can switch to the upgrades panel by clicking on the upgrades button in the upper right corner. Once an upgrade appears, you can buy it by clicking on the buy button.



You can view the menu by clicking on the menu button. The shop will be exchanged for the menu panel, but you can still see your progress on the left side of the screen. The menu offers three options. You can go into settings, multiplayer or view your achievements.

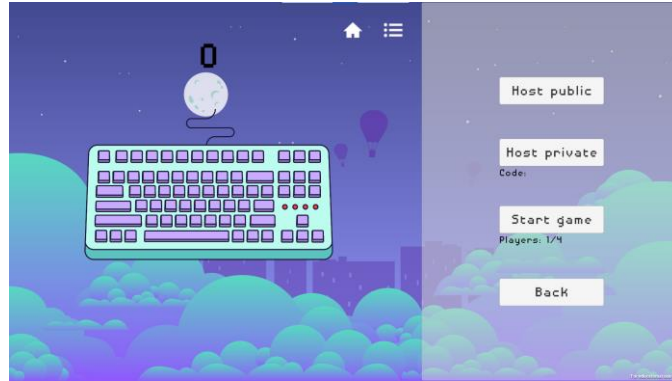


If you are not satisfied with the game visuals, you can go into settings and change them. There are six versions of the background and six versions of the keyboard that you can look through and pick which ones you like the best.

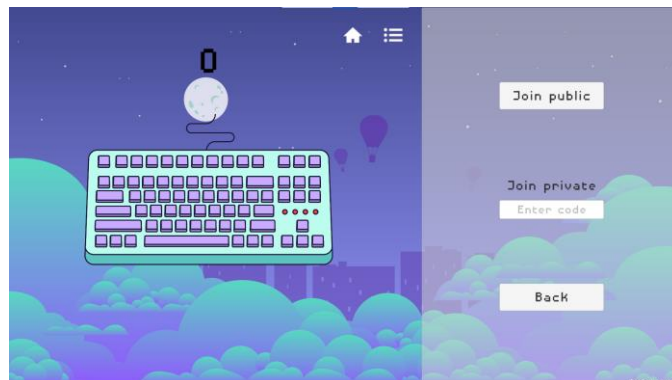


If you click on the multiplayer button, you will be given two options. One is to host a new game and one is to join a new game.

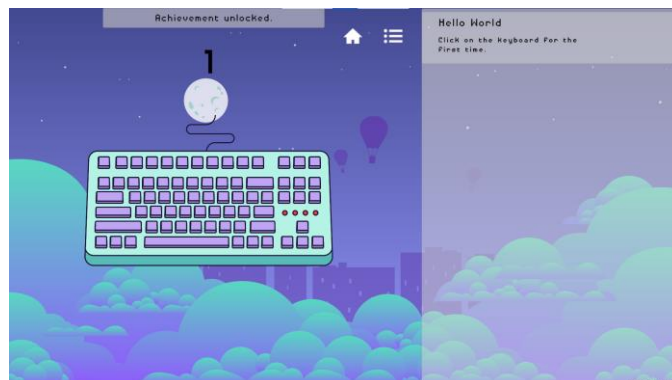
If you choose to host, you will see this screen. Hosting a public lobby will enable anyone to join. Hosting a private lobby will generate a code in the space below it. Others can join the lobby with that code. How many players are currently in the lobby is displayed below the button to start the game. Once you have the number of players you want, you can start the game. Doing so will restart everyone's progress.



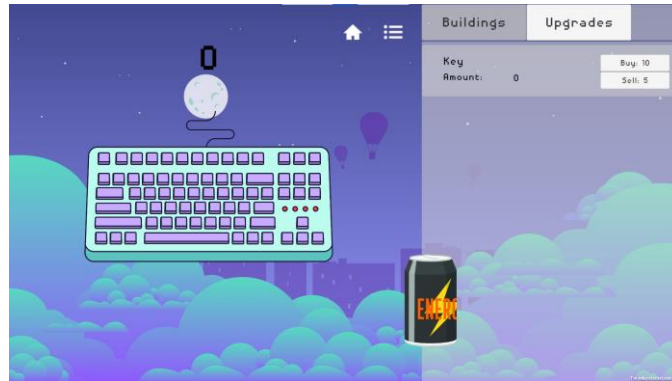
If you choose to join, you will see this screen. Joining a public lobby will automatically move you to a lobby that is not full. If no such lobby exists, you will be notified. Joining a private lobby requires you to type in the join code the host gets.



Whenever you unlock an achievement, you will receive a notification. You can then look at your achieved achievements by clicking on the achievements button.



The position of energy drinks is randomized. They appear between five to fifteen minutes after the last one. By clicking on them, you receive a boost. Which boost it is will be shown in a notification.



## Development documentation

### Specification summary

The application is inspired by Cookie Clicker. It contains a lot of the same mechanisms – gaining currency by clicking, buying and selling buildings, buying upgrades, unlocking achievements, getting boosts. However, they are implemented differently. The biggest difference is the fact that Code Clicker supports multiplayer.

### Program structure

The program is split into several scripts based on a logical structure.

#### **Homescreen.cs**

The home screen has its own script that handles what happens when buttons related to it are clicked.

#### **Notification.cs**

Similarly, notifications have a script that handles the appearing and disappearing animation as well as changing the message.

#### **NumberSuffixes.cs**

The number suffixes script handles number formatting and which suffix to append to it.

#### **Achievements.cs**

Achievements logic is also in one file. It contains two classes. One of the classes defines a single achievement, along with the requirement to unlock it. The other class then handles setting up all the achievements correctly and updating the UI when one of them is achieved.

#### **EnergyDrink.cs**

The script for energy drinks contains functions to position the game component off screen and on screen at the correct times. It also handles all the logic related to randomizing the boost, stopping the boosts and the boosts themselves.

#### **Menu.cs**

Menu has functions related to showing and hiding all the panels from the menu and the shop. Additionally, it also takes care of the settings options (keyboard and background).

#### **Clicker.cs**

The clicker script has all the logic regarding LOC. It contains functions for adding LOC from clicking and from buildings. It takes care of showing new buildings at the right time and has all the logic regarding restarting the game. It ensures that the values are displayed correctly in multiplayer through server and client RPCs.

#### **Buildings.cs**

Buildings have one parent script, which contains an abstract building class. It has all the required functions for buildings, such as buying, selling, and refreshing the UI. Other than that, it has functions that reset the building into its default state, which is needed for the restart. It

also has the basic functions for upgrades, such as showing them in the panel and adding them to the building once bought. However, they are set up independently.

### **Key.cs, URandom.cs, ...**

Each building has its own script that takes care of setting up everything related to itself, such as the default values and upgrades related to it. It also sets up the correct conditions for each upgrade.

### **Condition.cs**

The condition script consists of a class that contains information about what must be achieved before an upgrade can be shown, as well as a function that returns whether those conditions have been met or not.

Multiplayer is handled mostly in the scripts for lobbies and relays.

### **LobbyManager.cs**

The lobby manager script sets up the multiplayer buttons and has all the logic related to lobby creation, joining a lobby, and starting the game.

### **Relay.cs**

The relay script contains functions related to relay creation, cancellation and joining a relay.

## **Possible extensions**

A save and load function could be added. As it stands right now, once you exit the application, all your progress is lost. Implementing a save and load function could make the game experience more enjoyable long term, as well as changing the along limitations for how many LOC can be gathered to something bigger.

Of course, more buildings, upgrades and achievements can be added. As it stands, upgrades aren't very balanced, as some buildings have several, while others only have one. Achievements could be more varied.

## **Conclusion**

The game itself wasn't extremely hard to implement in singleplayer, instead I spent a lot of time learning how to operate Unity. The most difficult part was adding a functioning multiplayer and all the functionalities related to it, but I believe I've managed to conquer that as well. Overall, it was a fun project, thanks to which I learned a lot.