

Documentation for Konkos (CPS1015 Assignment)

Student Dave Galea
Lecturer Benjamin Bugeja
Date April 2025

1. Introduction

Konkos is a web-based, idle and incremental game created as the CPS1015 Web Development Foundations assignment. The project aims to show proficiency in front-end development using HTML, CSS, and JavaScript, as well as back-end implementation with Node.js and Express.js.

This documentation gives an overview of the decisions made throughout development, such as user interface (UI) design, game mechanics, and technical server and client communication configuration. In addition, a briefing on how to test and run the game locally is provided at the beginning of the documentation.

Konkos is a game, that, as one can see upon playing (and by the language of the title), is heavily inspired by Malta and its rampant construction. The idea behind the design of the game is to give the user a fun experience while tapping into the absurdity of the idea, which is, unfortunately, a very realistic imitation.

2. Running and Testing

Below are the steps to play and test the game for yourself.

1. Clone the Repository from GitHub:

Open a terminal and run the following command to clone the repository from GitHub:

```
git clone https://github.com/davgal27/Konkos-Idle-Game.git
```

This will create a folder called `Konkos-Idle-Game` with the game's files.

2. Change Directory into the Game Folder:

Navigate to the folder that was just created:

```
cd Konkos-Idle-Game
```

3. Install Dependencies:

Run:

```
npm install
```

This command will install all necessary packages.

If you do not have Node.js Package Manager (npm) installed, you will get a prompt to install it. Answer y (yes) to proceed, and then re-run `npm install`.

4. Start the Server and Game:

After installing dependencies, you can run the server with the following command:

```
node Server.js
```

The terminal should display the following:

```
Server running on http://localhost:3000
```

This means that the server is running. The game should now be accessible through your web browser by pasting the link. For the best viewing experience, maximize the tab in landscape format.

3. UI Design

Overall Design Choice

The user interface for *Konkos* was designed to be compact, with all elements immediately visible to the user. Such elements include the Build button, counters, upgrades, and achievements. The choice for keeping all elements readily viewable (especially the achievements) rather than being hidden behind a menu is to keep a clear sense of progress and keep up with the theme of a 'race to make more'.

The overall interface is designed with heavy inspiration from pixelated graphics style, and was done so with intention of giving the player a sense of 'gamification' of the absurd reality in the world of the game.



Figure 1: Screenshot of the UI.

Calls to Action

In the first few seconds of the game, the player is given context on the world they are about to play in, and prompted to begin by clicking 'The Button'; this was highlighted with text fading in and out near it as a prompt for the player to start. Upon clicking, the context paragraph fades, and the player is met with the full interface of the game.

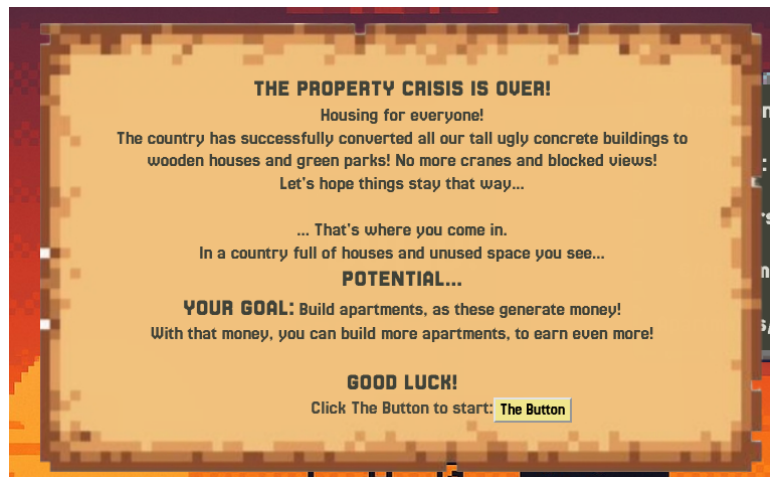


Figure 2: Screenshot of the starting screen giving context to the player.

Upon starting, the player can note that the whole layout of all parts of the UI center around the large 'Build' Button. This center focus of the game and the main way the player generates income, especially in the early stages of the game before builders are a true asset. Having just one button that the player was prompted to click throughout the game to generate income further enforced the theme of having a singular goal in mind: building.

Semantic HTML Usage

Semantic HTML elements were incorporated to improve structure and accessibility. The interface was organized with elements such as:

- `<button>` for interactive actions.
- `<article>` for the starting page which gives context to the player.
- `<section>` for the three main sections of the of the interface: Counters, Upgrades, and Achievements.
- `<h2>` and `` to emphasize upgrades and achievements section's headings.
- `<figure>` for images.
- `<main>` for images such as the sky which took up the entire background of the game.

Accessibility and Role Attributes

While semantic HTML was used, additional improvements were made regarding accessibility, such as using `aria-label` and `role` attributes on image-based buttons and images which were important to depict the scene of the game. These improvements provide descriptive text for users relying on assistive technology such as screen - readers. Additionally `alt=""` was used frequently for images so to aid the visually impaired as to what each button did.

Further improvements regarding accessibility can always be made, and such improvements are an important point going forward in future projects. During the development I was made more aware regarding the importance of accessibility integration as a design philosophy rather than an afterthought.

4. Game Mechanics

Core Game-play Idea

Being an Incremental game, *Konkos* aims to have a slow start, with a rampant increase in speed of game-play as the player progresses. The core idea revolves around building apartments; it is the **only** thing that the player can directly do in the game. From the apartments that the player builds, resources like money are generated, allowing the progression of the game through improvement. To improve, the player has a choice of upgrades, including a trigger for a timed event, which will assist them in increasing the speed of placement of apartments, and the amount of money earned for each apartment placed.

Upgrades

There are four upgrades, one of which is a timed event which will be later expanded on.

- **Increase Apartment Size:** This gives the player more money per apartment earned, and accordingly increments the counter displaying money earned per apartment built.
- **Increase Your Power:** This gives the player the ability to build more apartments for each click of the Build button they press. This increments the counter displaying the apartments built per click.
- **Hire a Builder:** This upgrade handles the 'idle' element of the game, granting the player a builder which will build an apartment every second even when the player does not interact with the game. The money earned from the apartments built by builders corresponds to the money earned per apartment built counter.
- **Deal With a Politician:** This is the Timed event which will be expanded on further in the following sections.

It should be noted that each upgrade can be purchased more than once, and the price for each upgrade increases slightly after each purchase to keep with the progression speed of the game.

Different Approaches to Completion

Upon gathering some money to buy some upgrades, the player quickly realises that they have two main routes to follow; a hands on approach, or an idle approach. This relates to the choice of which upgrades to pair if the player wants to progress quickly.

For a hands on approach: The player must pair **Increase Apartment Size** with **Increase Your Power**, so to earn more money per apartment built, while simultaneously building many apartments per click.

For an idle approach: The player instead pairs **Increase Apartment Size** with **Hire a Builder** so to earn more money per apartment built, while having multiple apartments being built per second due to the higher builder count. This approach pairs extremely well with the timed event to be discussed below.

Timed Event

As mentioned before, *Konkos* incorporates a timed event mechanic upon the purchase of the expensive upgrade: **Deal with a Politician**.

This upgrade, for thirty seconds, grants the player a 10x multiplier of the current amount of builders they have, which is why it pairs very effectively with an idle approach of playing. Due to the very high resource counts generated by this event given that it is purchased late in the game due to a high price, there is a cool-down period of 60 seconds to prevent "bulk buying" of the upgrade.

Timers are indicated near the bottom of the screen to indicate that there is the event currently going on, and that there is a cool-down.

Achievements

The achievements are based on the three resources of the game: Apartments, Money, and Builders. Additionally, there is an achievement granted to the player upon purchasing the upgrade of activating the timed event twice.

Some achievements are earned very early on in the game to give the player a sense of motivation in the early stages, and then are fairly evenly distributed throughout different stages of progression. The achievement designed to be the hardest to achieve is the achievement unlocked when 200 builders are purchased, as since prices increase per upgrade, the 200th builder cost quite a high amount compared to other upgrades.

5. Server Implementation

Overview

The technologies used were Node.js for server creation, and express.js for building APIs used as listed below. It is a simple server made to only function locally on one's device and does not handle public hosting of the web application. It does function properly in terms of saving progress of the player through API calls mentioned below.

POST request

When the browser (client) makes a POST request to the URL `/SaveGame` defined by express.js, it writes to `SavedState.json` via `fs.writeFile` the current data for counters and achievements. This was also made possible through the File system module which allows the code to interact with files within the player's file system.

GET request

When the browser makes a GET request to the URL `/LoadGame` defined by express.js, it reads `SavedState.json` via `fs.readFile` the saved data for counters and achievements, and updates the UI with the saved values. The File system module is also necessary here.

Auto-saving

From the client-side, a function `SaveGame()`; is defined which fetches `/SaveGame` via the fetch API. This function is set to an interval of 10 seconds, thus the game state is saved constantly.

6. Conclusion

This project was delightful to work on, and increased my appreciation in multiple facets of development; both in the web development area as well as in the game development area.

Thinking of a way how to put an idea from a simple drawing on a piece of paper to a responsive web layout, with functional logic handling all of the buttons and counters helped me to develop my way of thinking in terms of programming, and made me more aware about how even the simplest of web pages function. Additionally, it helped increase my awareness on the importance and thought needed behind accessibility considerations and helped me understand the workings of web applications in more depth.

While *Konkos* is a simple idle clicker game, I hope that players will appreciate the message it aims to convey.