**15112 Design Proposal: 112Jump\***

Roy Huang

**Project Description**

112Jump is based on the popular mobile game Doodle Jump. Players control a jumping character (referred to as the Doodle) and navigate it through a never-ending series of platforms. The Doodle must jump onto these platforms, some of which have special properties, and must avoid falling off of the screen or bumping into monsters. As the Doodle goes higher and higher, the player’s score increases and the levels get more challenging. Players can save their high scores and compete with others to claim the title of 112Jump!

**Competitive analysis**

There have been many emulations of Doodle jump made in the past. Similar to their projects, 112Jump will implement basic game functions that are found in the Doodle Jump game, like jumping onto platforms, scoring, graphic elements, and sound effects. Many of these projects were programmed using Pygame. This project will be using CMU graphics instead. Aside from basic game functions found in every other Doodle Jump python project, 112Jump will feature extra features. Multiplayer data will be saved so that a leaderboard can be formed, similar to the scores page found in Albert Haque’s[[1]](#footnote-1) Doodle Jump project. Aside from basic game functions found in every other Doodle Jump python project, players will be able to customize their Doodle. Players will also be able to adjust parameters for their game, like a Zen mode with unlimited Doodle lives and Night mode with less player visibility.

**Structural Plan**

Specific components will be separated into separate files. Different components and functions related to those features will have their separate classes:

* A Doodle class keeps track of character data, like position and Doodle color
* A Monster class keeps track of the movements and whereabouts of monsters
* A Platform class with special children classes based on different properties (like moving, one-use, or spring platforms)
* A Main file for drawing and checks for game conditions (platform collisions, game over, points etc.)
* Files for multiplayer data writing and retrieval

**Algorithmic Plan**

A challenge for this project will be implementing realistic physics for the Doodle character. The Doodle must jump upwards and fall back down as if it was experiencing gravity. To tackle this challenge, an algorithm to calculate acceleration and velocity must be implemented. As well, any collisions between the Doodle and objects will be checked using a method in the Doodle class. Another feature will be keeping track of players and their high scores. A dictionary system will be used to input and retrieve scores, saving data in a format that will be retained even after the application is closed. Based on the current score of the user, the intelligent map generator will increasingly make levels more difficult, modifying the number of platforms and monsters that are generated.

**Timeline Plan**

11/18/21 - Basic jumping doodle, moveable by a player on screen, platform generation and interaction

20/11/21 - Follow doodle up (“camera”), implement scoring and game conditions

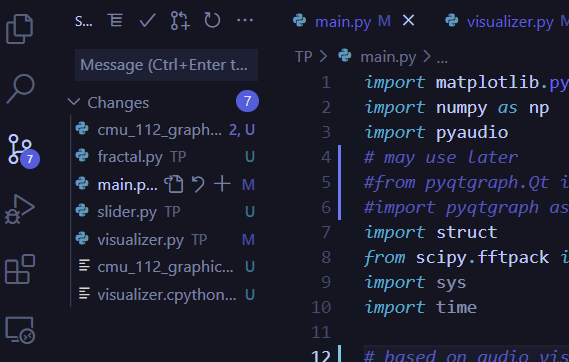
22/11/21 - Add special platforms and monster, backtracking algorithm

23/11/21- intelligent map generation

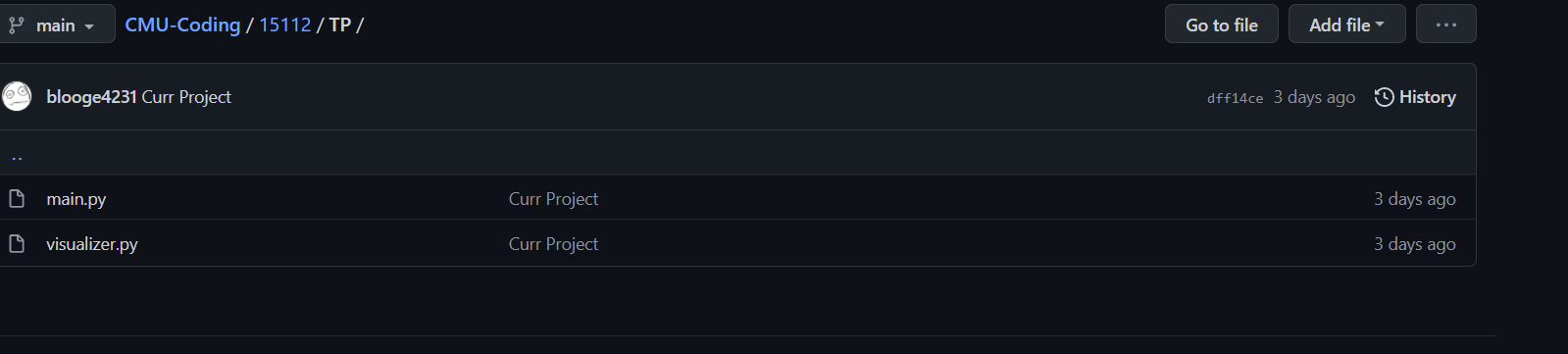
29/11/21 - Doodle customization and UI update

**Version Control Plan**

Using VS Code github extension, projects will be backed up onto the Github cloud, accessible from any device with an internet connection.



*Merge and commit changes to master branch, can also sync changes from Github onto local*

*Example project saved on Github*

**Module List**

Pyaudio (post MVP)

\*This was a pivot project, some files included were developed earlier in the week for the previous TP, but will not be used for this project

**TP 2 Update**

112Jump will feature a backtracking based algorithm that gives users move suggestions. Optimal moves will be decided on how close platforms are to the user, as well as how close platforms are to monsters on screen. Struggling on deciding whether to jump on that moving platform or spring platform to get past a monster? Fear not, the algorithm will suggest a move for you!

To focus on developing algorithmic complexity (move suggestion), Doodle customization and Zen/Night game modes will be delayed or removed from the project.

**TP 3 Update**

Doodle customization and Zen/Night modes have been removed from the project. New UI elements were added, specifically multiple screens (main screen, game over, leaderboard) and in console gameplay instructions. As well, users can enable test functions by pressing ‘t’ while in game, allowing them to manually jump, generate gameplay elements, and print out the distance of each platform to the user. Sound effects have been added.

1. https://github.com/ahaque/Doodle-Jump [↑](#footnote-ref-1)