CMSC 5233
Mobile Application Development
Game Collection Progress Report
Spring 2019

Paul Christy, Lee Shuman, Diane Truong

Lee Shuman's Progress

- Documented the high level design of the Gladiator Combat game.
- Researched leaderboard APIs available for free
- Determined that controlling our own API would provide a closer fit to our team's needs, due to the fact that authentication and account management was not in the scope of our project plan.
- Created a SQLite3 database project at (View source at https://glitch.com/edit/#!/join/1b1fa3e3-1394-4bcc-aafa-f62213f1f2cf)
- Populated database with player_id, game_id, and score_id tables and established primary/foreign key relationships.
- Created RESTful API through server.js on the glitch host. Basic API:
    - api/v1/player
        - POST – takes "player_id" (ie X'<GUID>') and "player_name" (string) parameters and creates an entry in the sqlite database. Returns status of 200 for success, or 400 for error.
        - GET – takes "player_id" and returns the matching database entry in a JSON string. Returns status of 200 for success, 400 for error, and 404 if the id is not in the database.
    - api/v1/score
        - POST – takes "game_id" (ie X'<GUID>'), "player_id" (ie X'<GUID>'), and "score" (positive number from one to 9 digits long) and creates an entry in the sqlite database. Returns status of 200 for success, or 400 for error.
        - GET – takes "game_id" and returns the top ten scores of that game in a JSON string. Status code is 200 for success, or 400 for error.
- Work Deviations:
    - Creating game has not started. Due to professor recommendation that game was ambitious, Gladiator Combat will start as a very basic RPG battle. I intend to take more of an Agile approach to the title and will just add functional pieces as I go.

Diane Truong's Progress

- Planned Angry Cats requirements and features.
- Tested out various JavaScript physics engines to see which is the most comfortable to work with.
- Implemented the basic collision boundaries for all Angry Cats entities.
- Implemented simple game physics.
- Implemented player control for Angry Cats.
- Researching how to add graphics to Angry Cats.
    - The p2.js physics engine does not support graphics so another library for graphics will be required.
- Work deviations:
    - Creating app images is slightly behind. It will be moved back a bit until the games are more completed.

Paul Christy's Progress

- Researched the sudoku rules
- Determined that using nested for loops would be the best way to check to see if the sudoku board is correct.
- Created a basic navigation system for the application
- Working on creating a sudoku board

Original Work Breakdown Structure for Progress Report

1. Game Project
    - 1.1. Week 1
        - 1.1.1. Document game rules — Lee Shuman
        - 1.1.2. Document game rules — Paul Christy
        - 1.1.3. Document game rules — Diane Truong
    - 1.2. Week 2
        - 1.2.1. Leaderboard Web API module — Lee Shuman
        - 1.2.2. Design app navigation — Paul Christy
        - 1.2.3. Create app images — Diane Truong
    - 1.3. Week 3
        - 1.3.1. Gladiator Combat – Design basic character stats — Lee Shuman
        - 1.3.2. Sudoku — Paul Christy
        - 1.3.3. Angry Cats – Create base game — Diane Truong