

**Multiplayer Grid Based Simultaneous
Movement Skill Game
NEA: Georgiy Tnimov**

0.1. Abstract

0.2. Client

0.2.1. Client Synopsys

The Client is Alexander Tahiri, a software developer at Studio Squared and the developer of Tapp, a game based on a 4x4 grid, which consists of 12 inactive tiles, and 4 active tiles. Players use the mouse cursor to click on an active tile, which then deactivates that tile and activates a new, currently non-active tile. the objective of Tapp is to achieve as high a score as possible, without making any mistakes. The Client requires a derivative of this game, which tests simultaneous dexterity of both hands, additionally The Client wants to incorporate a competitive aspect to the game, which consists of a leaderboard section, allowing players to see their position within the rankings and a Tetris-99-esque game mechanic, where players compete to either achieve the highest score, or last the longest in a mass multiplayer format. The Client has specifically asked for the Catpuccin colour scheme to be used, and that a mobile friendly version of the game is created. The Client has sufficient computing power to host both the client, server and database, which will be provided free of charge.

0.2.2. Interview Notes

0.3. Research

0.4. Prototyping

A rudimentary prototype has been made, which tested out multiple different input methods for simultaneous inputs, which has finalized in a "cursor"-based system, where you have two cursors controlled by Wasd-like movement, with each set of controls representing their respective cursor, additionally it has been decided that both cursors need to be on individual Tiles, to prevent copying movements on each hand. this prototype also implemented server-side move verification, making it more difficult to cheat. Finally, the UI design of the prototype will be used in later iterations of the project.

0.5. Outline Design

0.6. Objectives

0.6.1. User Interface

1 user can interact with the grid

1.1 user can move both cursors using keyboard on the grid

- 1.2 user can "submit" moves using a keybind
- 1.3 user can reset game (in single player) via a keybind
- 2 user can change gamemode (singleplayer,multiplayer) on the main page
 - 2.1 user can change grid size (4x4,5x5,6x6)
 - 2.2 in singleplayer, user can change time limit
- 3 user can access settings
 - 3.1 user can modify keybinds for each action in the game
 - 3.2 user can change DAS
 - 3.3 user can change ARR
 - 3.4 user can log out of account
 - 3.5 user can reset all keybinds to a sane default
- 4 user can play the game
 - 4.1 on game start, user sees cursors are positioned on opposing sides of the board
 - 4.2 on game start, user sees the starting active tiles
 - 4.3 user can view current game score
 - 4.4 in singleplayer, user sees time remaining
 - 4.5 user can "submit" their move
 - 4.5.1 user can interactively see if the move was valid via a colour interaction which flashes green or red depending on if the move was valid, a valid move is when the two cursors are on two active grid tiles within the grid boundary and they are distinct active tiles
 - 4.5.2 on successfull submit, user sees two new tiles become active, which were previously inactive and are not on current cursor location
 - 4.6 cursors are rendered via two different colours, with the two cursors being visually distinct but symmetrically consistent
- 5 user can see statistics post singleplayer game end
 - 5.1 user views their score
 - 5.2 user views if their score was validated by the server
 - 5.3 user views their leaderboard position
 - 5.4 user can copy their game statistics to the clipboard for sharing
 - 5.5 user can see if it is their personal best
 - 5.6 user has the option to start a new game from the results menu
- 6 user can view leaderboard
 - 6.1 user can view the top 10 highest scoring games
 - 6.2 user can view their personal top 10 highest scores
- 7 user can play the multiplayer gamemode
 - 7.1 user can see the other players movements on other grids in the game

7.2 user can see their remaining score quota for each 5 second interval period

7.3 after a user has been eliminated by not reaching the quota,the user can view their position in the game

7.4 after a user has been eliminated, the user can view the remaining game being played

8 user can log in to the application

0.6.2. Functionality

0.6.3. Usability

0.6.4. Data Management

0.6.5. Server Side

1 User CRUD

1.1 user authentication via google oauth

1.2 user schema contains references to game history

1.2.1 simple user statistics

2 Game Verification

3 Multiplayer implementation

4