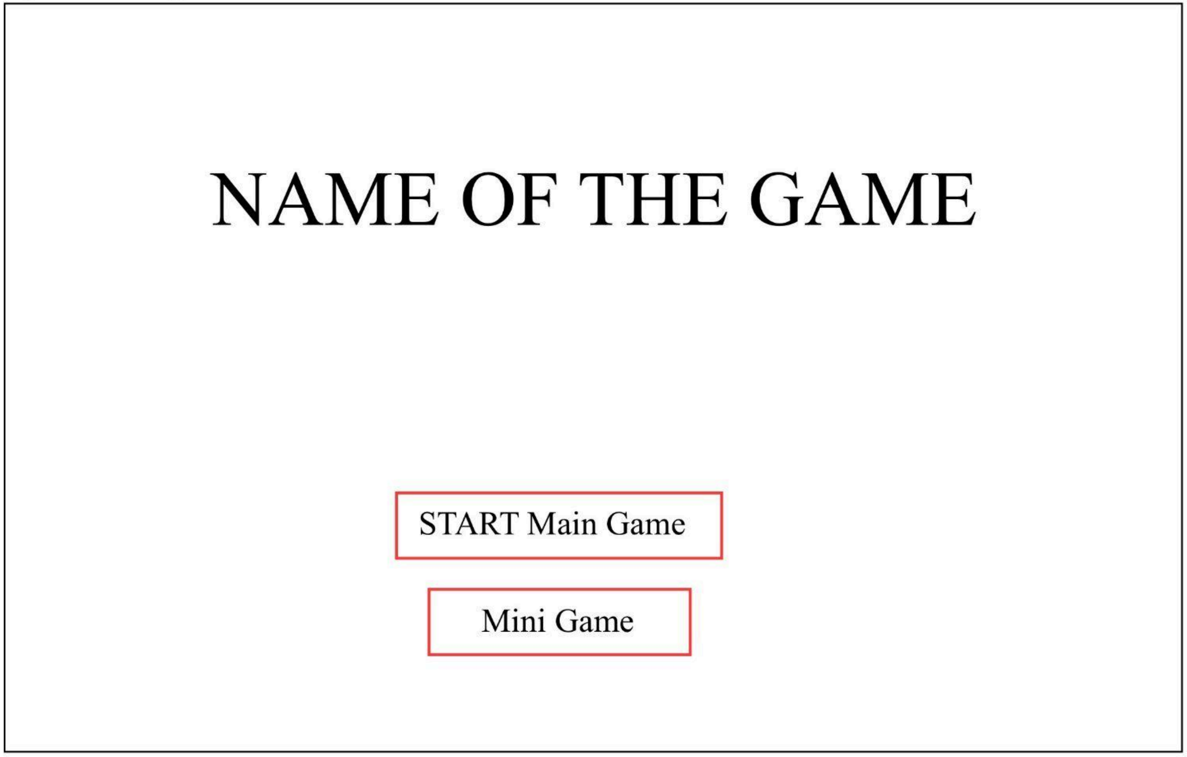
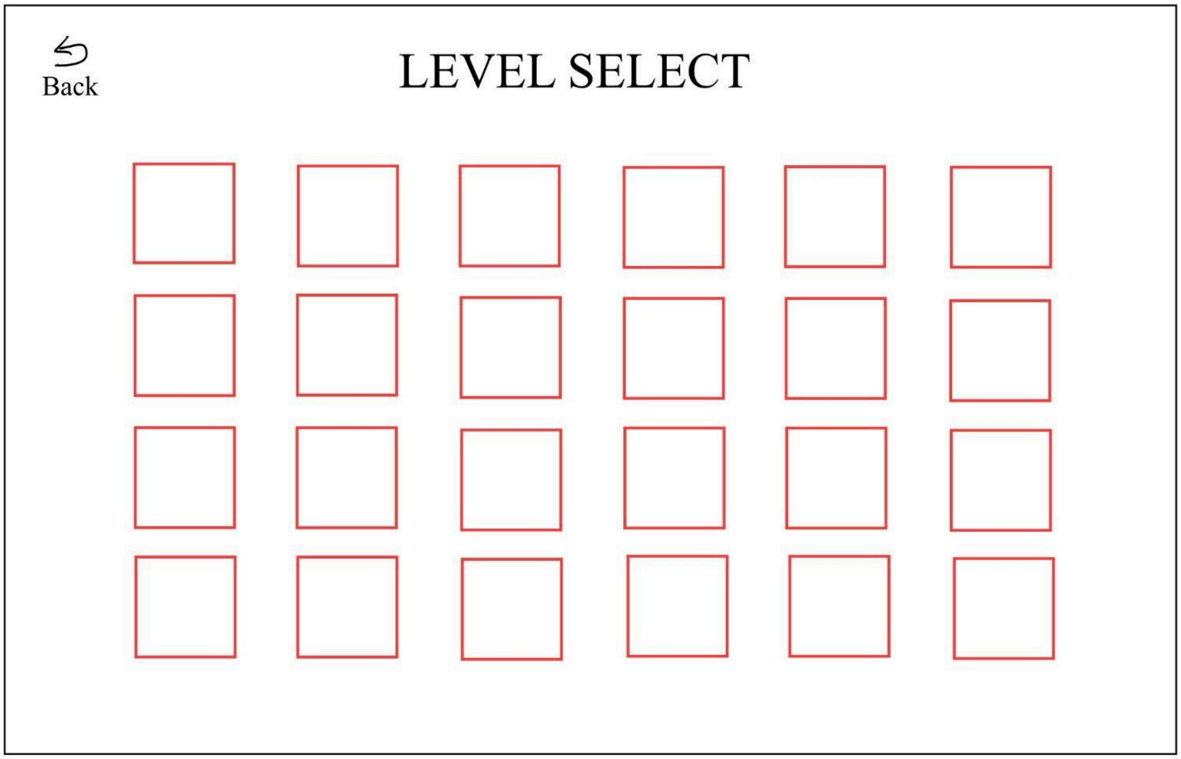
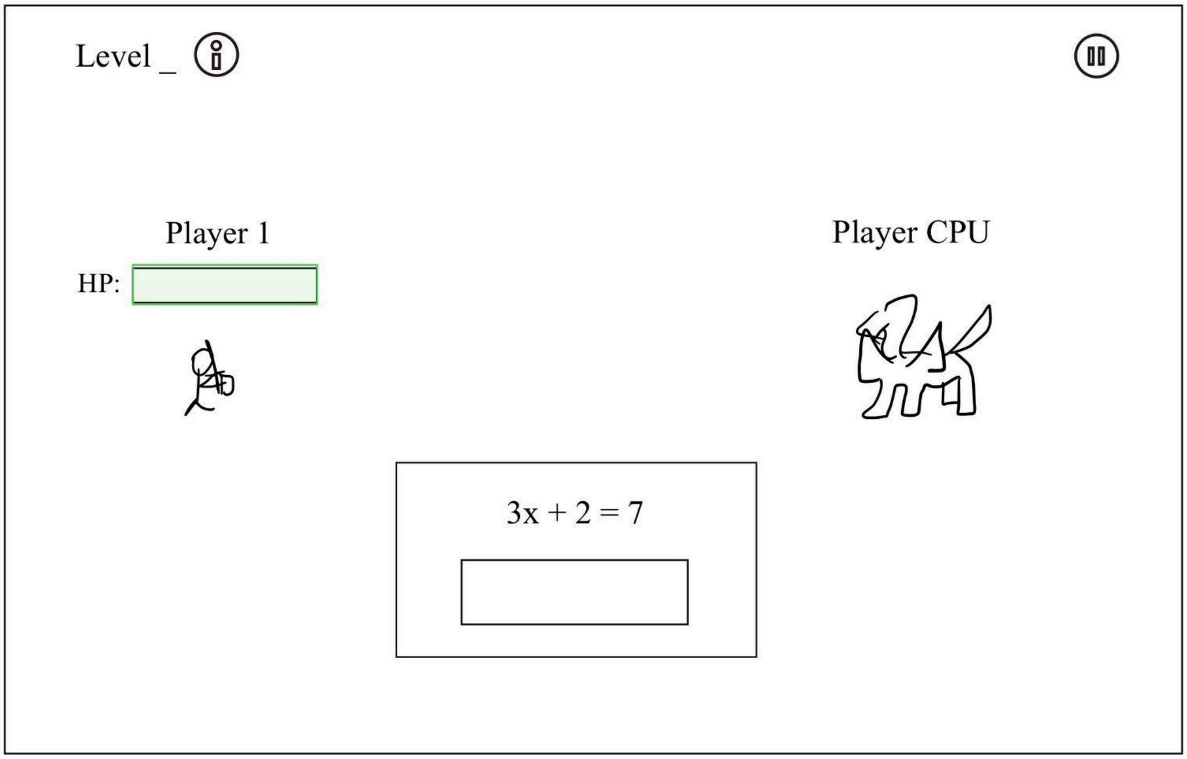
Requirements Document - CSE 110 Group 38

1. As a user, I want to be able to smoothly navigate through the game so that I can choose and play each level at will. (8 day estimate)
    - Design the main menu screen, which includes options for starting the main game (level select) and starting the mini-game. (1 day)
    - Design the level select screen, which includes selectable levels and an option to go to the main menu screen. (1 day)
    - Design the level screen, which includes options to go to the level select screen and go to the main menu screen. (1 day)
    - Implement navigation logic by linking these three screens together appropriately based on the available options of each screen established in Tasks 1-3. (2 days)
    - Create navigation flow tests by writing tests that test every individual option on every screen to be sure of its functionality. (1 day)
    - Run tests that simulate a user navigating through all the different screens and using all the different options to see if the program responds appropriately. (1 day)

1) Game Navigation

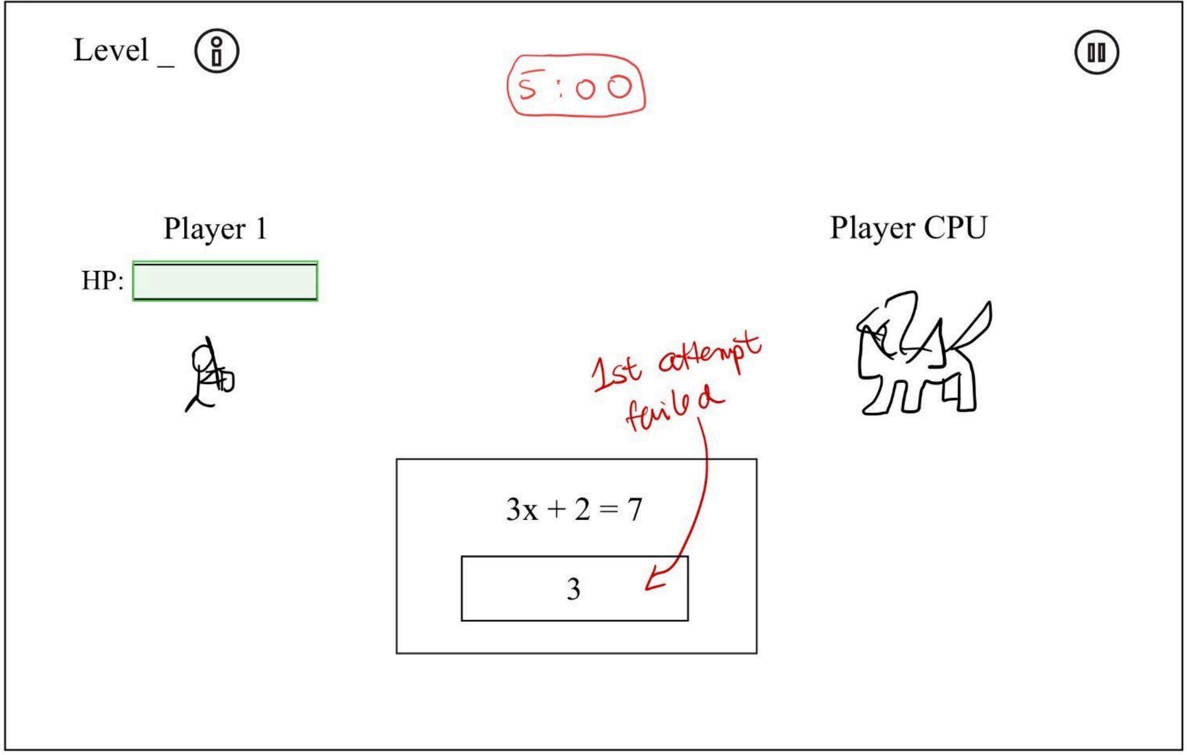


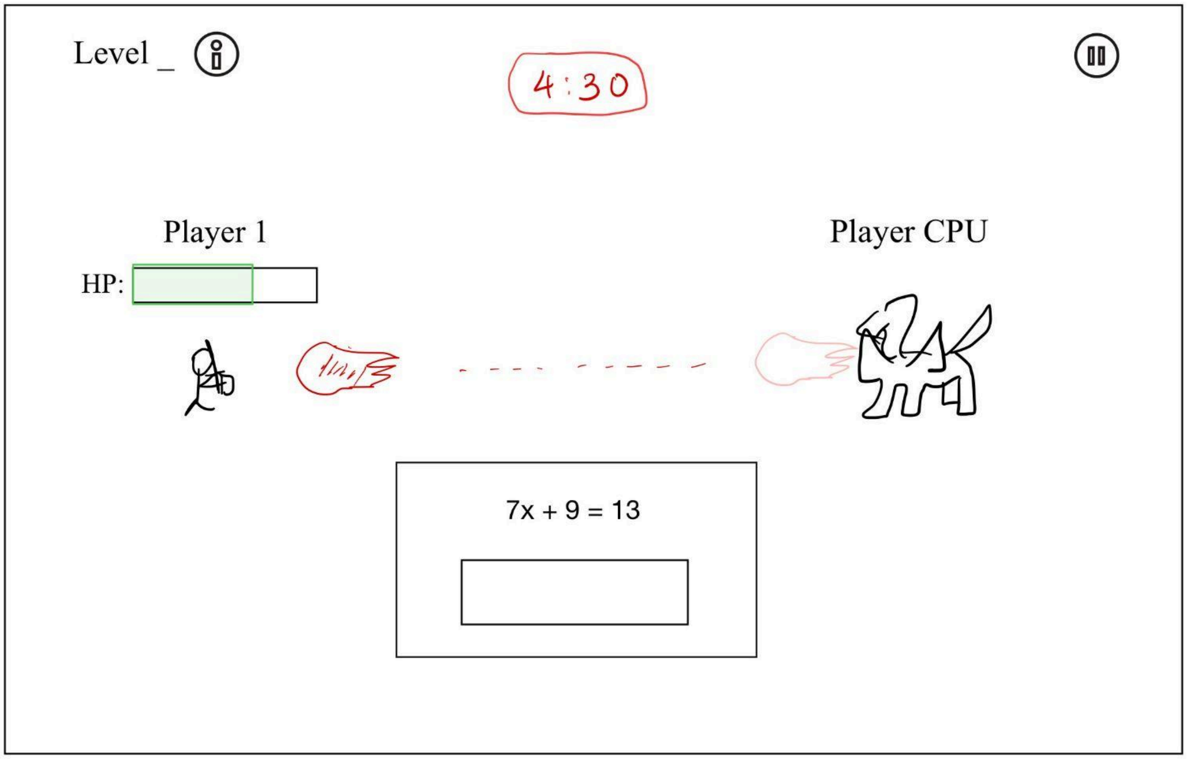



Level _
Player 1
HP:
Player CPU
3x + 2 = 7

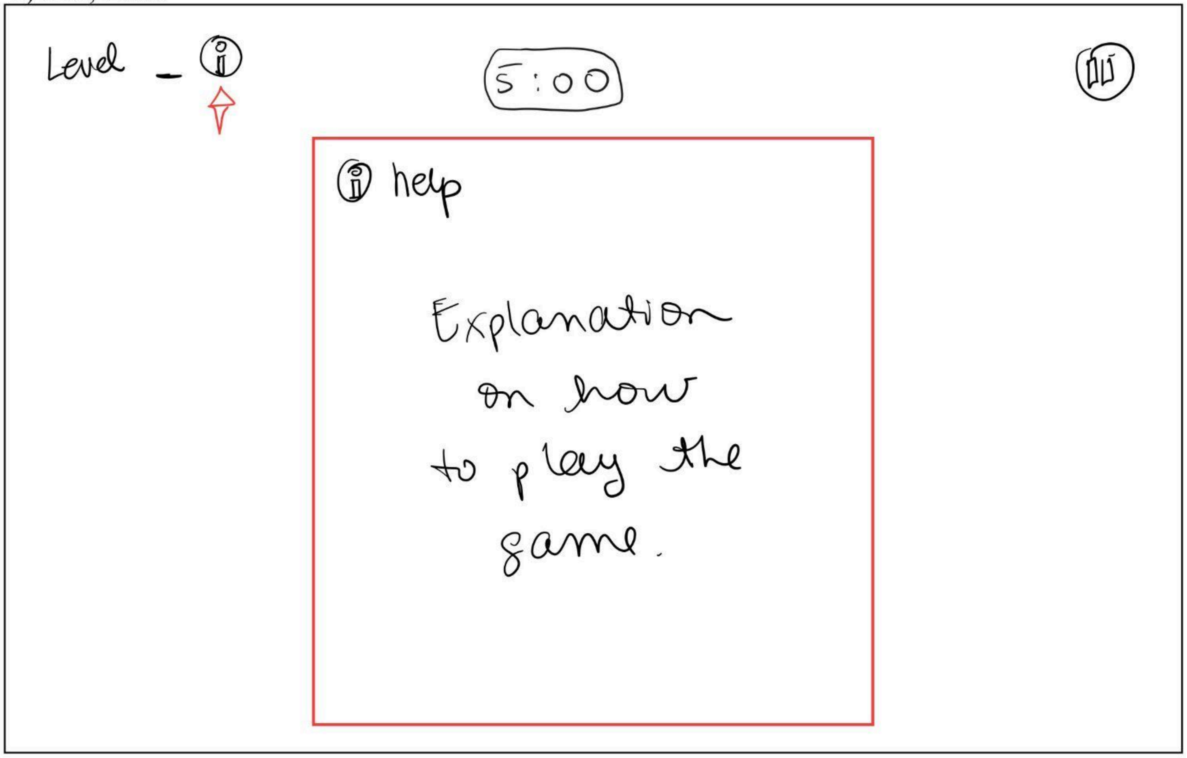
2. As a user, I want the gameplay sections to handle my actions and in-level progression, responding accordingly based on my input and how well I performed. (15 - 20 days)
    - Implement player actions and handling systems, which includes user input, question responses, attacking enemies. (3 days)
    - Add a pause overlay, which has options to resume, retry, go to the level select screen, and go to the main menu screen. (1 day)
    - Add a how to play (information) overlay, which has an option to resume. (1 day)
    - Implement enemy and boss defeat logic, which includes correctly detecting conditions for defeating an opponent. (3 days)
    - Implement progression system which progresses the user to the next enemy or area upon completing the current task. (2 days)
    - Add event and state tracking which keeps track of the current character and enemy states. (1 day)
    - Design winning and losing overlays for levels, which will both have options to retry, go to the level select screen, and go to the main menu screen. The winning screen will additionally have an option to go to the next level. (1 day)
    - Integrate with navigation and level systems so the levels are properly connected to the other screens. (1 day)
    - Implement obstacles for users such as enemy attacks and health loss. (2 days)
    - Create a test which attempts and checks enemy defeat conditions. (3 days)
    - Write tests to verify and validate all user input as well as correctly recognize keyboard input. (1 day)
    - Write tests which can edit and correctly detect and verify edited play values. (1 day)

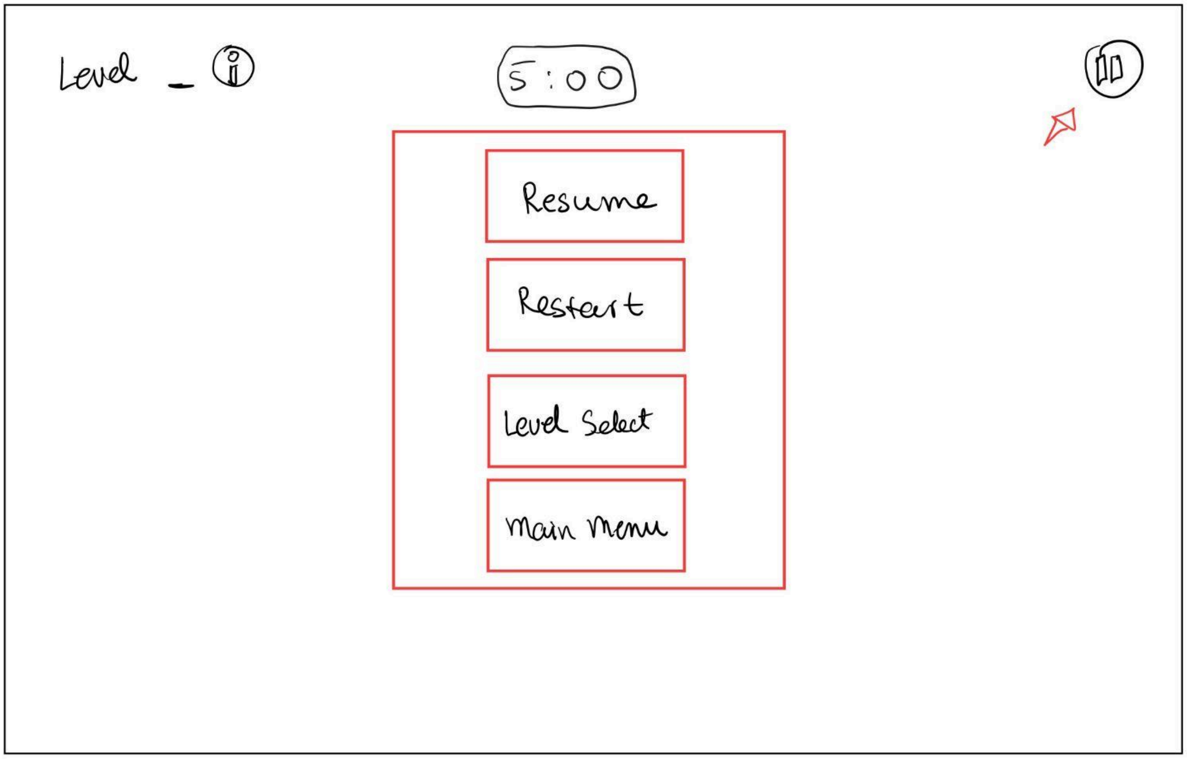
2) In-Game Mechanics



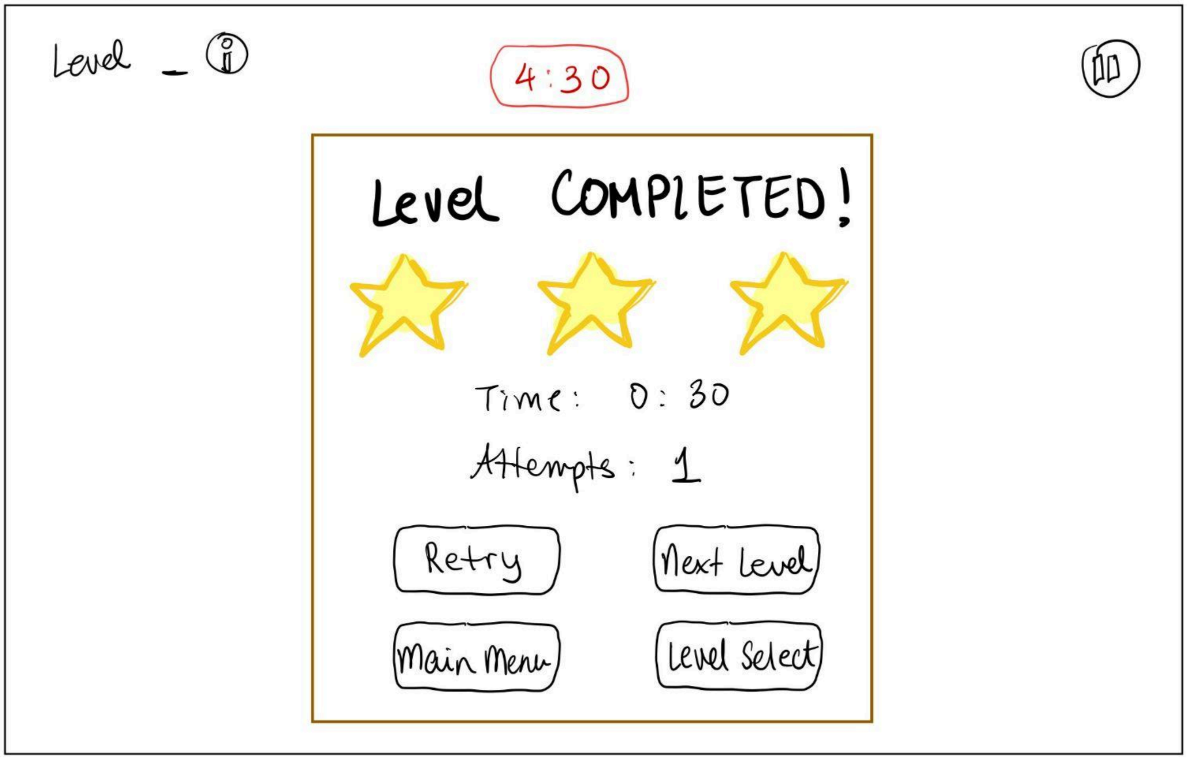


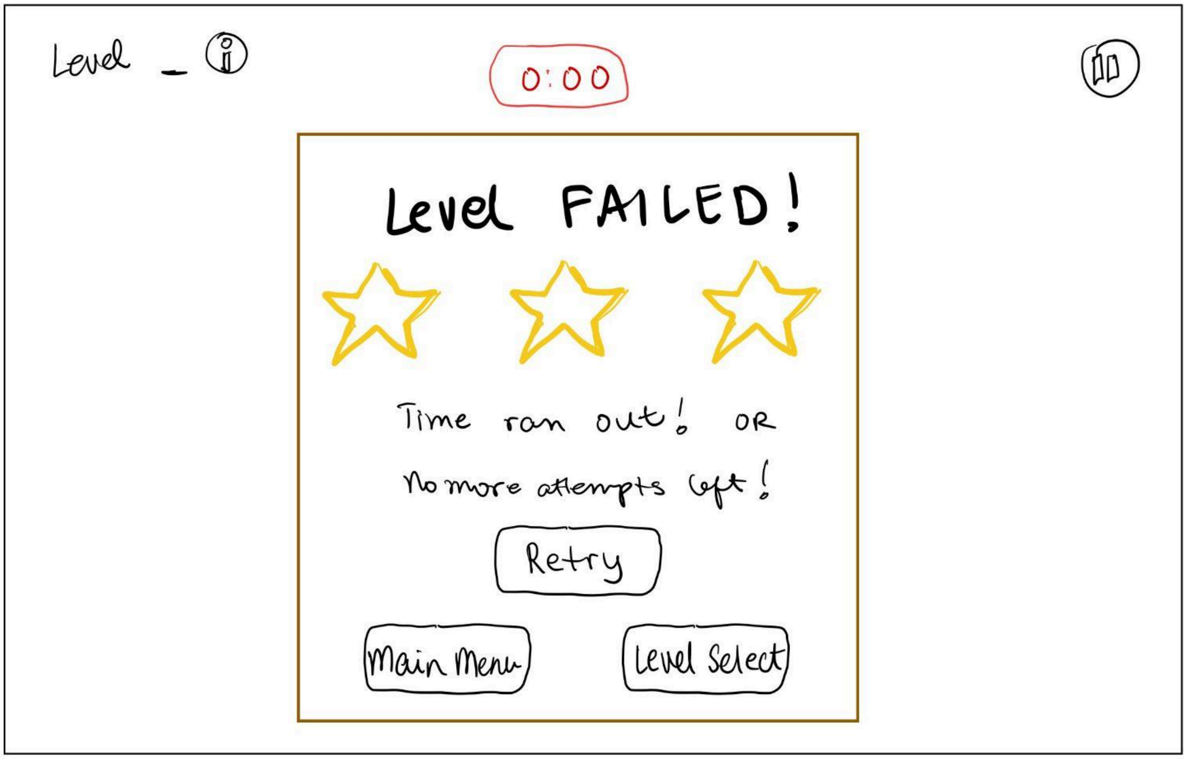
2) Info, Pause


Level _
i
5:00
i help
Explanation on how to play the game.


Level _
i
5:00
Resume
Restart
Level Select
Main Menu

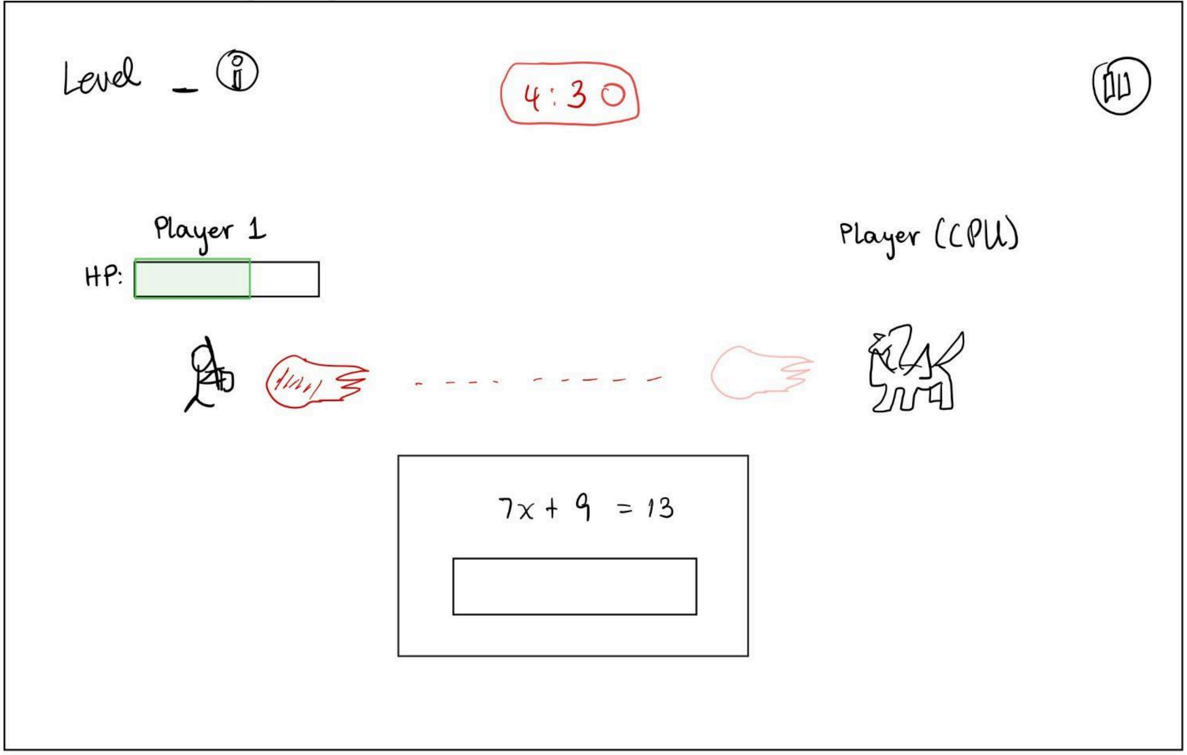
2) Winning / Losing Screens

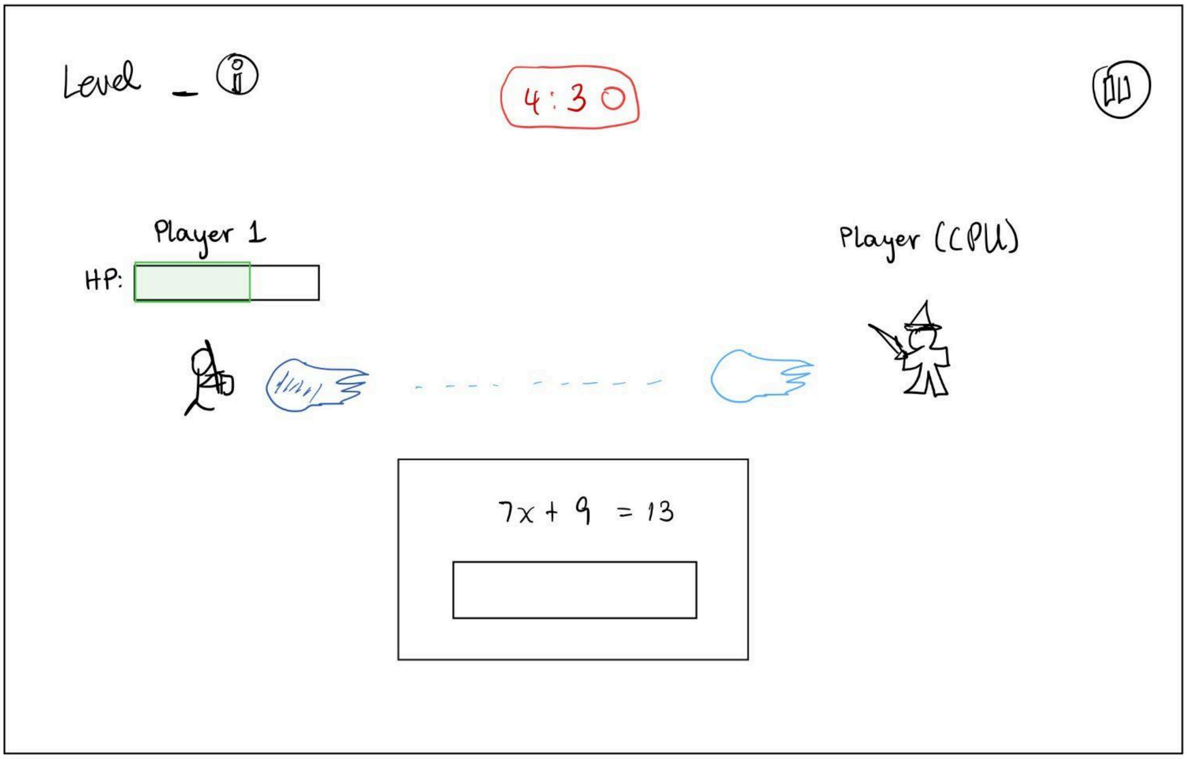




3. As a user, I want each level in the game to feature unique enemies, themes, and attack patterns so that gameplay is varied and enjoyable. (10 - 25 days)
    - Plan and design level themes as well as level progression structure such as increasing difficulty as the user progresses. (3 - 15 days depending on # of levels made)
    - Implement a basic level template which contains the base layout and structure for each level along with enemy logic. (3 days)
    - Implement unique enemies for user to interact with. (4 days)
    - Implement unique attack patterns between enemy and user. (2 days)
    - Write gameplay and progression tests which test for enemy creation, enemy defeats and selection of level themes. (2 - 5 days)
    - Run said tests in parallel with a visual analysis to ensure the game is handling the graphics as intended. (2 - 5 days)

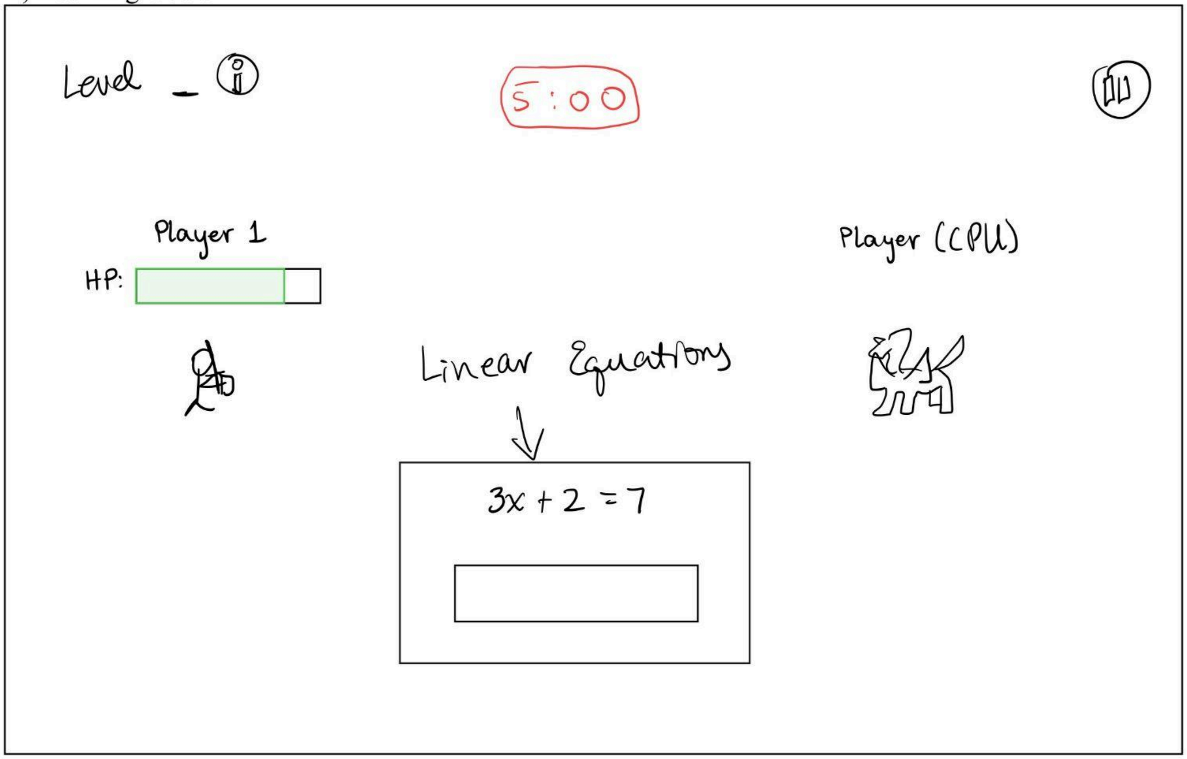
3) Different and Unique Graphics

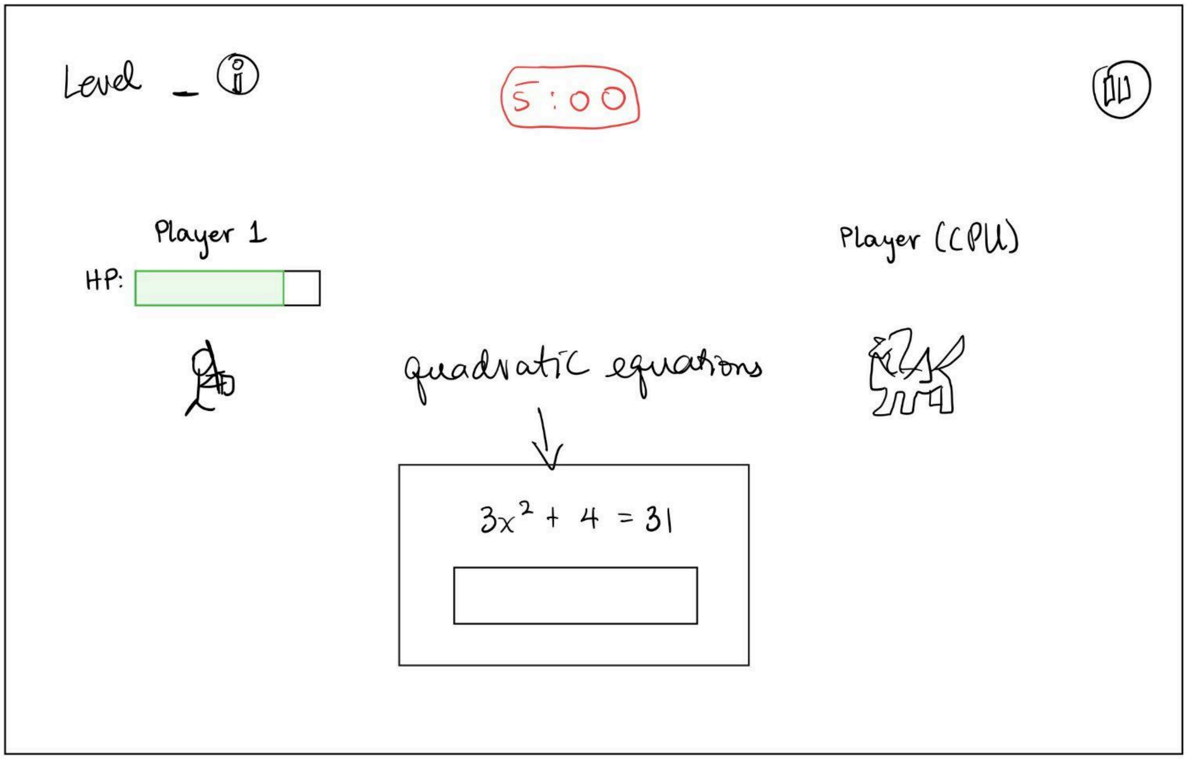




4. As a 7th-9th grade student, I want the game to help me practice systems of equations and linear and quadratic equations and inequalities so that I can solidify my understanding of these core concepts in Algebra 1. (15 days)
    - Research Algebra 1 Common Core on the above concepts to pinpoint what to teach. (2 days)
    - Prototype questions with varying difficulty for each concept. (1 day)
    - Standardize prototypes to a format that allows us to generate random questions at a given difficulty. (3 days)
    - Assign questions to levels in a manner to ensure all the concepts are covered and include problems ranging from easy to hard difficulty. (2 days)
    - Integrate question formats into their assigned levels. (1 day)
    - Add an appropriate solution/answer (if any) to each level to give users feedback if they get that problem wrong. (2 days)
    - Write tests to verify the random generation of questions and the correctness of their solutions. (2 days)
    - Run tests and pass them if questions are in the intended scope of the design, and solutions are correct based on the assigned values of that question. (2 days)

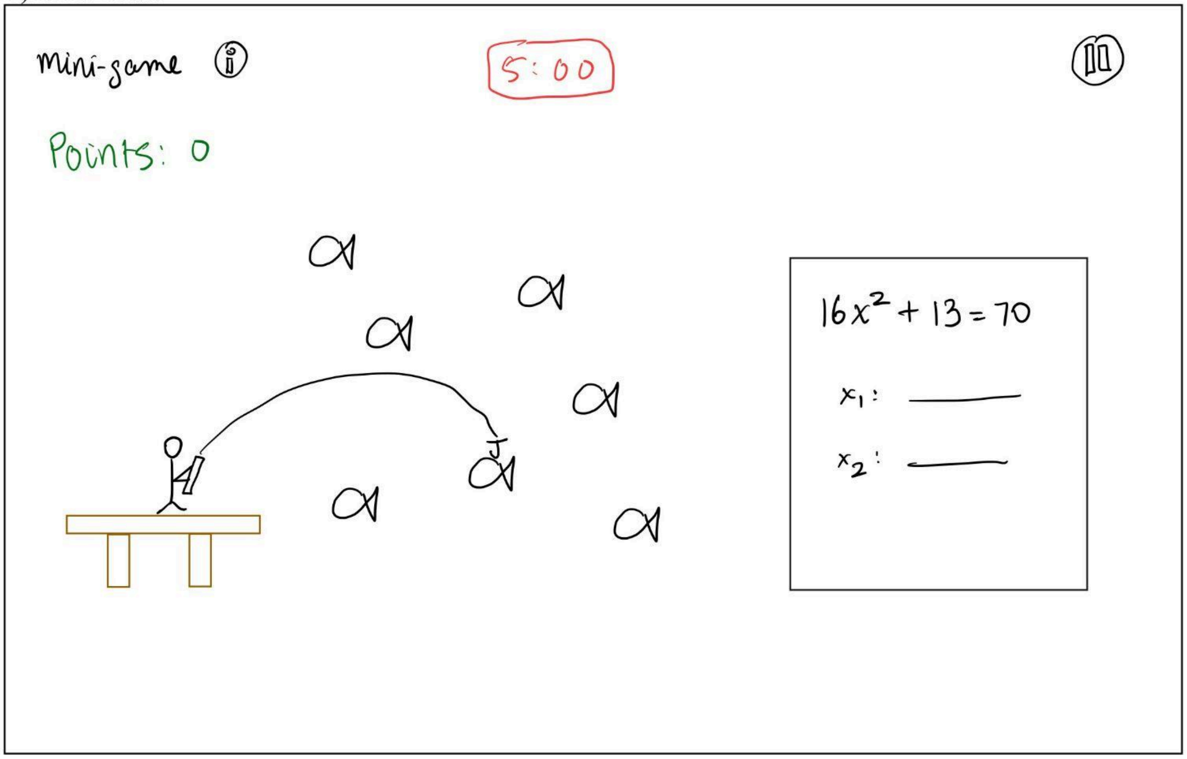
4) Learning Goals

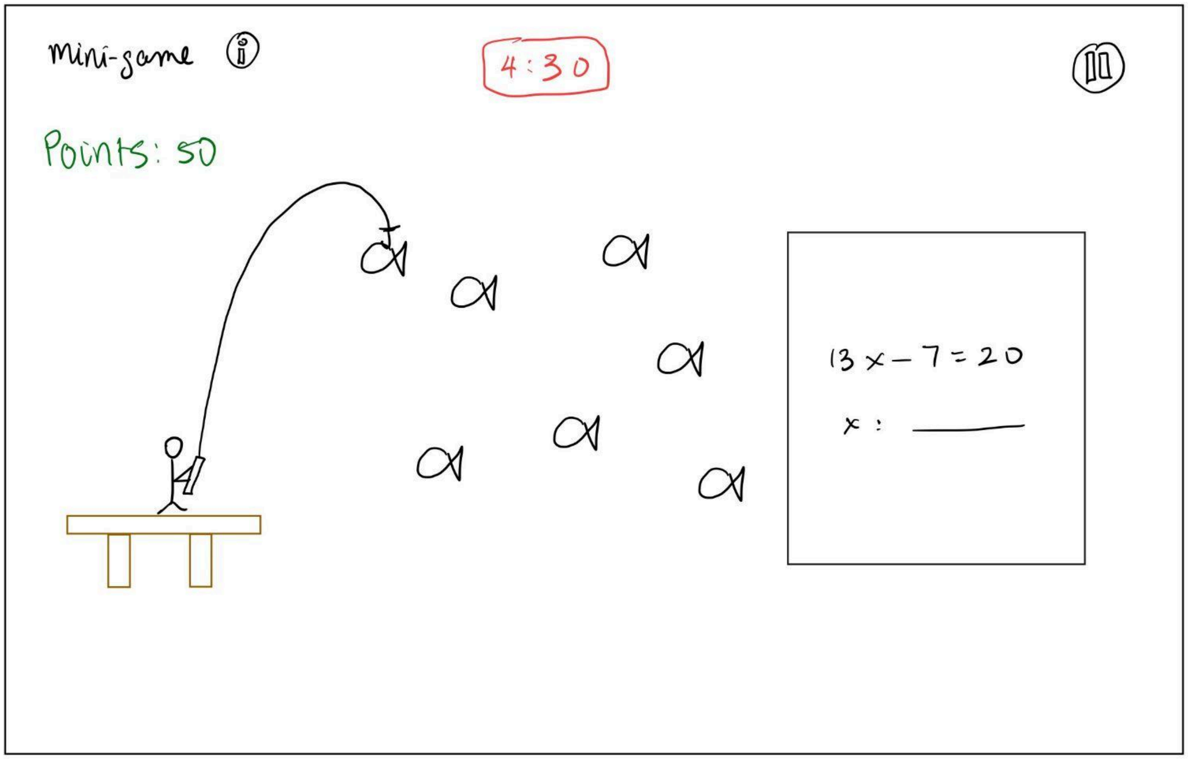




5. As a user, I want a mini-game with different mechanics from the main game so that I can stay engaged and continue practicing my skills in a fun, unique way. (12 days)
    - Design Mini game UI and core mechanics. (3 days)
    - Implement a grid for the fishing UI for fish to appear randomly on the screen. (2 days)
    - Implement a fish catching mechanic which calls an event for the user to complete in order to catch a fish. (2 days)
    - Add feedback and animations for catching and selecting fish. (3 days)
    - Integrate fishing Minigame into main game flow by being able to transition from main game to Minigame either through an option or in between levels. (1 day)
    - Write integration tests which test the functionality of catching and storing fish as well as transitions. (2 days)

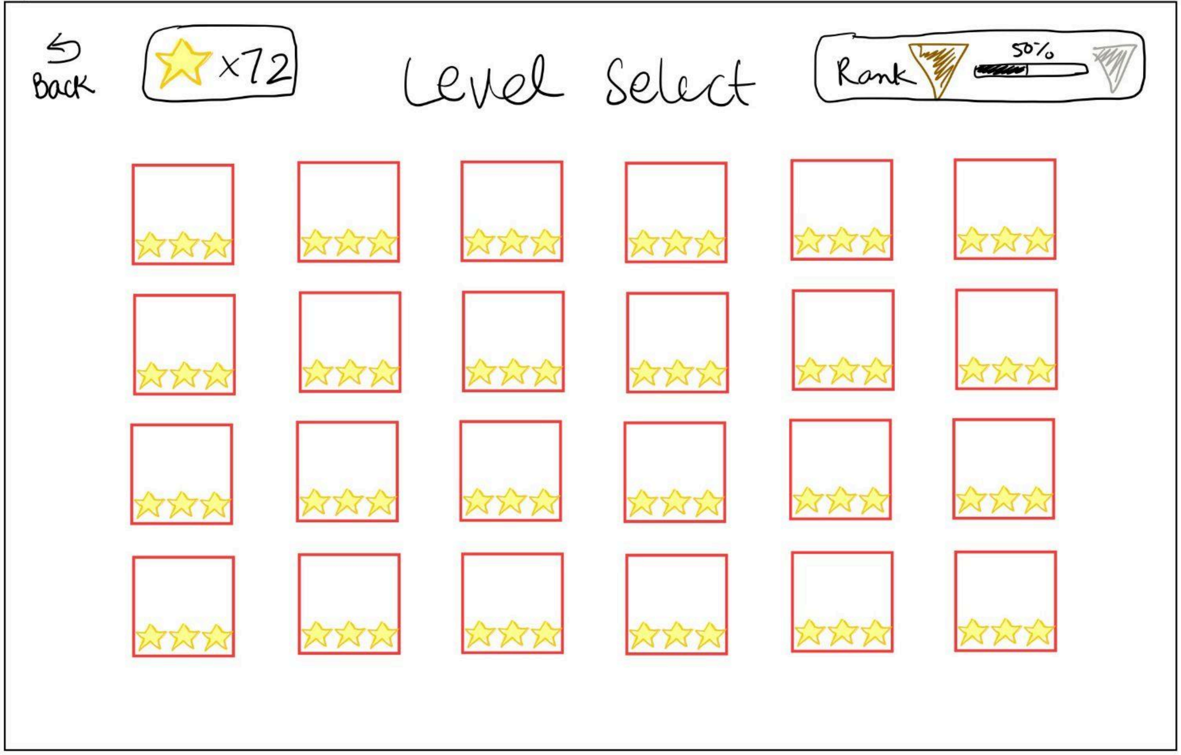
5) Mini-Game


mini-game
5:00
Points: 0
$16x^2 + 13 = 70$
x_1 :
x_2 :


mini-game
4:30
Points: 50
$13x - 7 = 20$
x :

6. As a user, I want to earn rewards and track my progress so that I can feel motivated to keep improving and complete more levels. (15 days)
    - During each level calculate the player's performance based on the level's selected measurement (correctness/time/both). (3 days)
    - Create a star system for levels that rates the players performance during the level and assigns them 1, 2, or 3 stars. (2 days)
    - Create a system that measures their total completion of the game measured by rank (bronze/silver/gold/diamond) based on how many of the game's total stars they have collected. (2 days)
    - Display player's total stars and current rank on the level select screen. (1 day)
    - Run unit tests that simulate playthroughs of each level, testing different combinations of correct and incorrect answers, as well as different time thresholds. (2 days)
    - Create expected star results for these playthroughs based on how we feel the game should be balanced for the average 7th - 9th grader (7th should probably find the questions and time constraints a bit difficult, 9th should feel quite comfortable) to compare to simulations, checking for accuracy. (3 days)
    - Run tests to calculate the player's rank at each possible star count, ensuring that each one matches our determined ranking scale. (2 days)

6) Rewards


Back
×72
Level Select
Rank
50%