

2.4 Functional Requirements

- **Functionality to be delivered.** You must list and number each functional requirement for your project. Note that some of this information can be copied/paraphrased from the [original project specification](#) provided to you. Again, at least some rewording will be necessary in doing this, as discussed above.
- You must also add any functional requirements that make sense for your specific game idea and the game mechanics involved. You should put yourselves in the place of the stakeholders (e.g. the player, etc.) and try to imagine what requirements they may have for your specific game.
- You may split the requirements listed in the specification into multiple requirements. Still, at a minimum, you should have one requirement for each requirement listed in the specification **AND at least 4 of your own functional requirements related to your specific game**. If you can identify additional functional requirements, feel free to list them here as well.

4.1 Functional Requirements

4.1.1 User Interface

Multiple Screens:

- **(4.1.3) Instruction/tutorial** – Display instructions, run through tutorial game
- **Main menu (4.1.2)**
- **Sub-menus**
- Select Single player/Multiplayer
 - Single player: Easy (single digit add/sub), Medium (double digit add/sub), Hard (mult/div/add/sub)
 - Create your own mode (input equations/answers for self-practice or parents input them for kids to practice)
 - Timed mode (Fastest player wins)
 - Countdown mode (60s to complete grid); Grid size increases each level; combos increase available time
 - Level selection menu
 - Multiplayer: Easy, Medium, Hard
 - Timed mode – 6x6 grid
 - Tournament mode – timed mode with 4 players, top 2 move to finals
 - Pyramid mode – each player does 2x2, 4x4, 6x6 grid, fastest player wins
 - **Gameplay screen** – Grids starting from 2x2, increasing each level; Name-top left; Time-top right; Quit bottom right
 - **(4.1.6) High score list** – Display highest level/fastest time for countdown mode; fastest time for timed mode
 - **Progress/Results screen** – Display correct pairs one after another if level is not completed, or in different colours

Mouse-Based Interaction:

- Use mouse to select menus, levels and game cards

Keyboard Shortcuts and Commands:

- Esc to exit game or return to previous screen
- S for single player, M for multiplayer, T for tutorial screen, H for high score list
- E for easy, M for medium, H for hard
- T for timed mode, C for countdown mode

Feedback Systems:

- Highlight menu buttons when being clicked or when shortcut commands are used
- Click game card to reveal equation/answer
- Cards and their borders disappear if correct; equation/answer disappears, border stays if incorrect match
- New level appears on level screen in countdown mode if level completed
- Highlight level being selected

- **Main Menu**

- Title
- 1) Start new game
- 2) Load game (Load a level that's previously been unlocked)
- 3) Tutorial/Instructions
- 4) High scores
- 5) Exit
- Game preview, Developers' names, Team # (38), Term created (Winter 2024), mention it was created as part of CS2212 at Western University
- Switch user

- **Instructions and/or Tutorials**

- Display instructions in text format
- Tutorial option to play a test game, explaining/highlighting what to do while a game screen is active

- **Game Play and Mechanics**

Scoring System:

- **Single player**
 - Display highest level completed and time it was completed in for each difficulty level
- **Multiplayer**
 - Display fastest time completed for each difficulty level

Progression Mechanics:

- **Single player**
 - Start with 2x2 grid, then 4x4, then 6x6, then 8x8; Increase when completed in given time limit

Educational Challenges:

- Varying difficulties, matching equations and answers by selecting cards in a grid to flip over, revealing their contents

Response to Player Input:

- Mouse clicks to select menu options as well as game cards
- Various shortcut keys (listed above – 4.1.1: Feedback Systems)

Feedback for Correct/Incorrect Actions:

- Correct answers disappear with card borders; Turn green as second card is clicked
- Incorrect answers disappear without borders; Turn red as second card is clicked

Time-Based Elements:

- Single player
 - 60-120 seconds to complete level of countdown mode
 - Timed mode uses time to complete a level as the score
- Multiplayer
 - Original
 - Timed mode - the time to complete the board is used as the score
 - Pyramid mode
 - Each player does a 2x2, 4x4, then 6x6 grid
 - Fastest player wins
 - Tournament mode
 - Timed mode with 4 players on 6x6 grid
 - Top 2 players move to finals, fastest player wins

- **Save/Load Game State**

- Whenever a new grid size is unlocked, automatically save
- Option to load previously unlocked levels in main menu

4.1.6 High Score Table

- Displays players name
 - Displays best time for timed mode
 - Displays highest level achieved, fastest time completed for countdown mode
- Accessible from main menu
- Updates automatically
- Displays at least 5 entries
- Gold/Silver/Bronze award next to top 3 names
- Stored on computer the game is installed on, stored in a file

4.1.7 Multiple Players

- Players enter name, initials, username, or email when starting new game
- Names included in save state, added to multiplayer high score list

4.1.8 Instructor Dashboard

- Includes users' name, highest level completed (countdown), current level they have are on but have not completed (countdown), fastest time completed (timed)
- Includes # of attempts for each level
- Password protected so players cannot access it (Password may be hardcoded or customizable. It does not require proper security (e.g. password hashing))

4.1.9 Debug Mode / Level Selection

- Mode for developers to jump to a specified level without completing previous levels to make testing easier
- Password protected (Password, key combination, cheat code, or secret button/UI element that normal players would not typically find)
- Includes editing/deletion of save states, level editor

4.1.10 Housekeeping & Error Handling

- Exit application cleanly, save data when doing so (instructor metrics must not be lost)
- UI elements scale correctly when the user minimizes/maximizes window (Can be disabled and only allow full screen or set window size)
- Any errors that the application could raise should be handled and clear messages in simple English shown to the user, that may help them understand and correct the error
- Ideally, avoid crashing without explanation
- Any user input that could potentially be incorrect or cause software faults, must be validated and checked for errors

Additional Functional Requirements

- **Practice mode**
 - A game mode for 4x4, 6x6 timed practise that doesn't get added to high scores; Used to become more familiar with the game
- **Make your own level**
 - Input 8 or 18 equations you would like to practise, along with their correct answers and they will be randomly placed in a 4x4 or 6x6 grid
 - Parents can use this to have their kids practice, or it can be used for self-practice
- **Easy, Medium, Hard**
 - Easy – Single digit addition and subtraction
 - Medium – Double digit addition and subtraction; (Answers must be below 100)
 - Hard – Double digit addition and subtraction, as well as multiplication and division (Answers must be below 100)
- **Tournament mode**
 - Original
 - Available for 3 or 4 players
 - 6x6 grid, timed mode
 - Lowest score is eliminated (slowest to complete), next round started
 - Pyramid mode
 - Each player does a 2x2, 4x4, then 6x6 grid
 - Lowest score is eliminated (slowest to complete), next round started
- **Countdown Mode Combos**
 - If 2 matches made within 3 seconds, 3 seconds added
 - If 3 matches made in 5 seconds, 5 seconds added
 - If 5 matches made in 10 seconds, 15 seconds added
- **Scenario model.** This section presents a model representing the functional requirements, in terms of actors, use cases, and activity diagrams. To carry out this modeling, you will need to analyze and understand these requirements, better preparing yourselves for what is to come. For this modeling, you need to document the following:
- **Actors.** You will need to describe each actor (user role, external system, etc.) involved in the use of the software. If your team is undertaking an extra feature that adds any additional actors, it must be included here. For each actor, you will need to provide the following information in the tabular format given below. Keep in mind that actors are not parts of the system itself (such as databases, GUI, etc.) and must be external entities such as a category or role of the user, external systems, etc.

Actor	Player
Description	the player interacts with the game using a keyboard and mouse with the option of an on-screen number pad. They are the end user of the system and are the users that control the gameplay. There may be more than one player depending on the settings of the game (single/multi-player)

Aliases	User/Student
Inherits	None
Actor Type	Person
Active /Passive	Active
•	
Actor	Instructor
Description	The instructor or teacher has access to a password-locked metrics page that can track player/student scores, levels, high scores, etc
Aliases	Teacher, TA
Inherits	None
Actor Type	Person
Active /Passive	Active
•	
Actor	Developer
Description	The Developer has access to a hidden debug mode that allows the actor to jump between locked levels/ modes, and possibly have access to cheats and other testing modes. This hidden mode is often protected by a password.
Aliases	Debugger, Admin
Inherits	None
Actor Type	Person
Active /Passive	Active

Use cases. Use cases describe specific usage scenarios of the software in question from the point of view of a particular actor. For each use case you identify, you will need to provide a formal definition of the use case. In most cases, you should have at least one use case per functional requirement. For the definition of the use case, you will need to provide the following information, and can use this tabular format:

Use cases: (delete before submission)

- Navigate the Main menu
- Selecting a level and difficulty (number of tiles and easy/medium/hard)
- Playing the game (clicking buttons in-game)
- Instructor dashboard (metrics dashboard that is password protected)
- make your own level (steps for creating your level)
- Tutorial
- High Score Table
- Progress/ Results Screen
- Multiple Players

Template:

Name	4.1.1 User Interface
Primary actor	User
Secondary actors	Developer and Instructor
Goal in context	Accessing the interface
Preconditions	Credentials accepted
Trigger	Login being accepted

Scenario	The user launches the interface and logs in with credentials
Alternatives	None
Exceptions	Possible Save state error
Priority	Medium
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.2 Main Menu
Primary actor	User
Secondary actors	Developer
Goal in context	Navigating the Main Menu of Math Match-up
Preconditions	User Login and information
Trigger	User accessing
Scenario	The user accesses the game, is prompted to the main menu, user navigates the main menu.
Alternatives	None
Exceptions	Possible Save state error
Priority	Medium
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.3 Instructions/Tutorial
Primary actor	User
Secondary actors	Developer
Goal in context	Option in the main menu to display instructions and run tutorial game for new user
Preconditions	User Login and information
Trigger	User accessing
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu and clicks the tutorial option
Alternatives	None
Exceptions	Possible Save state error
Priority	High
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.4 Game Play and Mechanics of Math Match-up
Primary actor	User
Secondary actors	Developer
Goal in context	Playing the level picked by the user
Preconditions	User Login and picking a specific level

Trigger	Users interacting with the different options on the Main Menu
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu, chooses the difficulty and plays the game
Alternatives	None
Exceptions	Possible error with the difficulty selection
Priority	Highest
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.5 Save and Load
Primary actor	User
Secondary actors	Developer
Goal in context	Saves the game of Math Match-up, updates any unlocked levels
Preconditions	The user plays the game
Trigger	The user completes the game
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu, plays the game and completes the level.
Alternatives	None
Exceptions	Possible Save/Load state error
Priority	Medium
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.6 High Score Table
Primary actor	User
Secondary actors	Developer
Goal in context	Navigating the high-score table after a game
Preconditions	The user playing the game
Trigger	User completing a game
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu, plays the game and accesses the high score table.
Alternatives	None
Exceptions	High score assertion done incorrect
Priority	Low
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.7 Multiple Players
Primary actor	User
Secondary actors	Developer

Goal in context	Ability to play with another user
Preconditions	Both the User's Login and information inputted
Trigger	User accessing
Scenario	The users access the game, are prompted to the main menu, the user navigates the main menu and both players click the multiplayer option.
Alternatives	None
Exceptions	Possible Save state error
Priority	Highest
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.8 Instructor Dashboard
Primary actor	Instructor
Secondary actors	Developer
Goal in context	Instruction Dashboard that holds the metrics of specific users.
Preconditions	The user playing the game
Trigger	User accessing
Scenario	The user runs a specific game, Metrics are saved from the User and stored in the Dashboard
Alternatives	None
Exceptions	Possible Save state error
Priority	High
...	Additional headings and information may be provided as you deem necessary.

Name	4.1.9 Debugging
Primary actor	Developer
Secondary actors	Instructor
Goal in context	To lock levels and edit save the state
Preconditions	Enter developer password
Trigger	Developer enters
Scenario	The developer logs in and navigates through the save state of the user or game.
Alternatives	None
Exceptions	Possible Save state error
Priority	High
...	Additional headings and information may be provided as you deem necessary.

Name	Difficulty Level pick
Primary actor	User

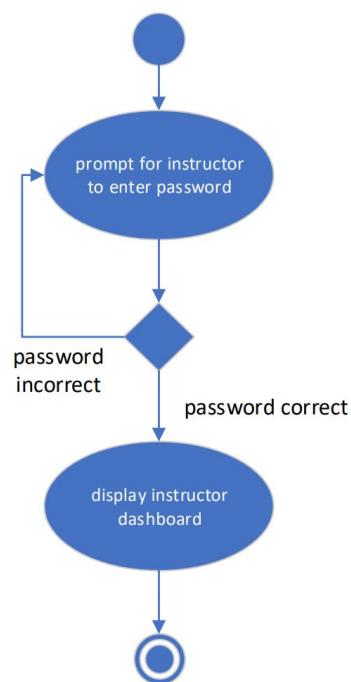
Secondary actors	Developer
Goal in context	Picking a specific tile level or difficulty
Preconditions	Accessing the Main menu
Trigger	User accessing
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu and chooses the difficulty of the game.
Alternatives	None
Exceptions	Possible error on level process
Priority	High
...	Additional headings and information may be provided as you deem necessary.

Name	Create Your Level
Primary actor	User
Secondary actors	Developer
Goal in context	The user creates their own level according to their preferences
Preconditions	User Login and main menu access
Trigger	User accessing
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu and selects the create your own option.
Alternatives	None
Exceptions	Possible error with creating tile pattern
Priority	Medium
...	Additional headings and information may be provided as you deem necessary.

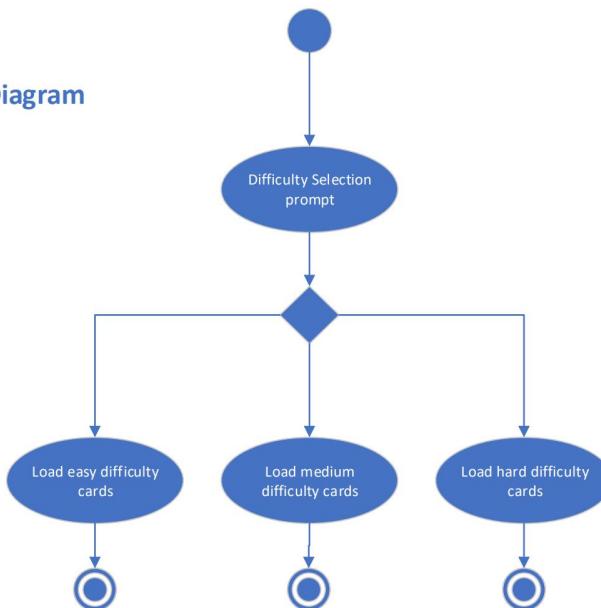
Name	Tournament Mode
Primary actor	User
Secondary actors	Developer
Goal in context	Playing the Tournament Mode
Preconditions	User accesses and is on the main menu
Trigger	The user selects tournament mode
Scenario	The user accesses the game, is prompted to the main menu, the user navigates the main menu and selects the tournament mode.
Alternatives	None
Exceptions	Possible Save state error
Priority	Medium
...	Additional headings and information may be provided as you deem necessary.

- **Activity diagrams.** For each use case defined above, you must provide an [activity diagram](#) that provides a graphical representation of the flow of interaction within that specific scenario. In a way, this can better capture the steps of the scenario (and their alternatives) using standard flowcharting symbols.

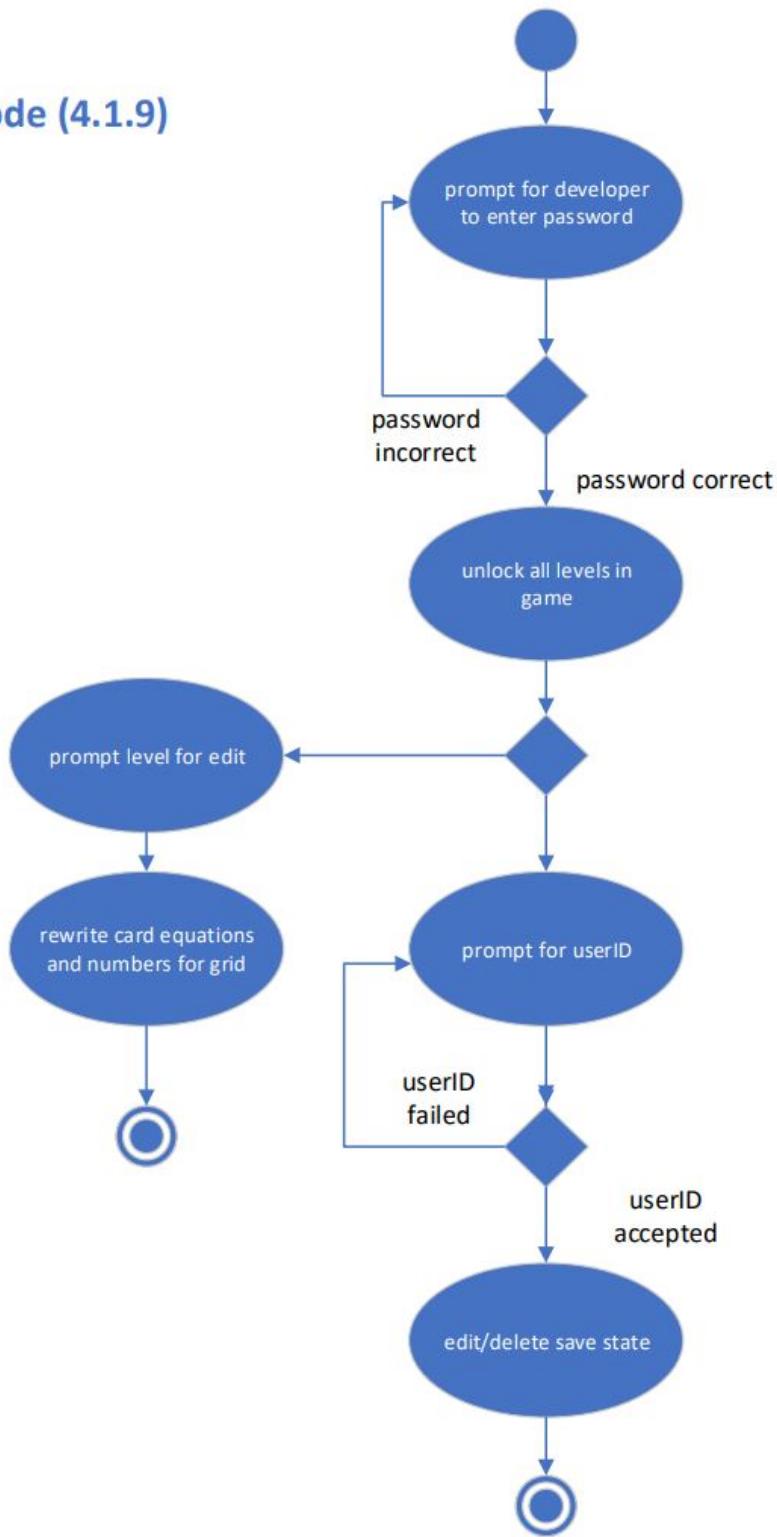
Instructor Dashboard (4.1.8)



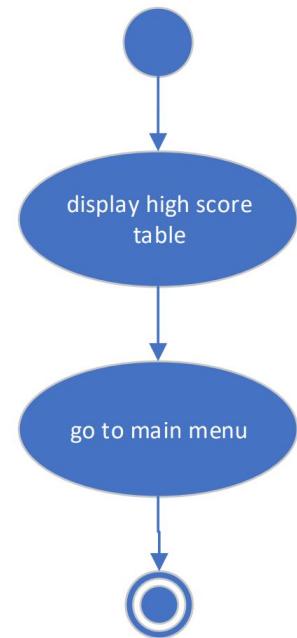
Selecting Difficulty Activity Diagram (additional)



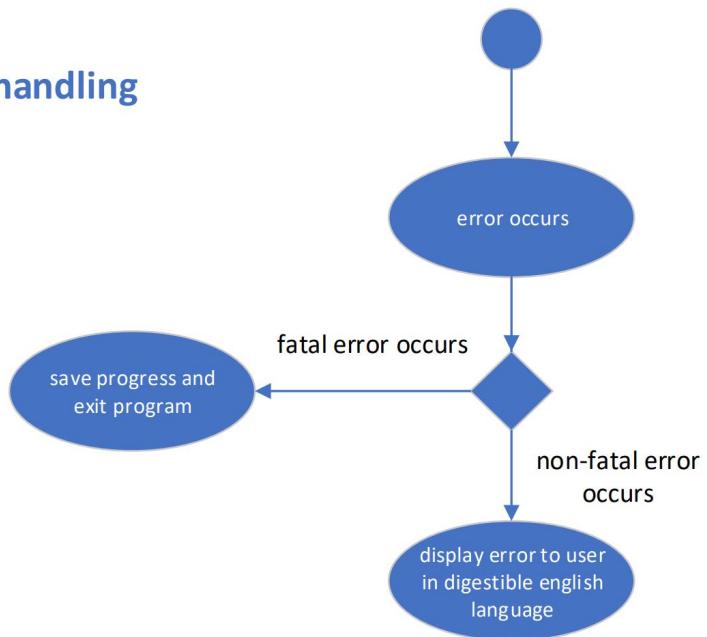
Debug mode (4.1.9)



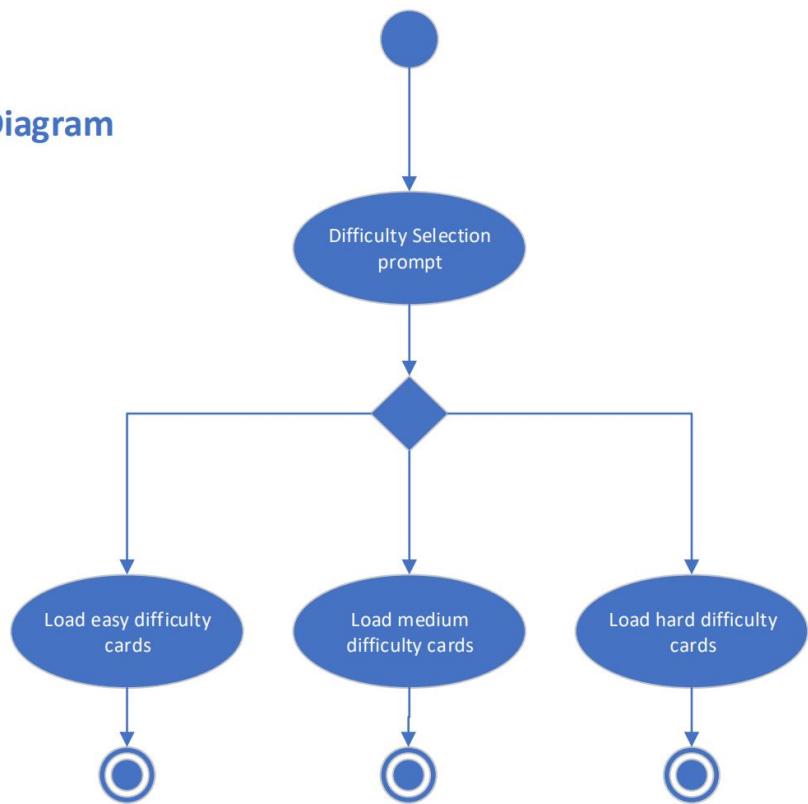
High Score Table activity diagram (4.1.6)



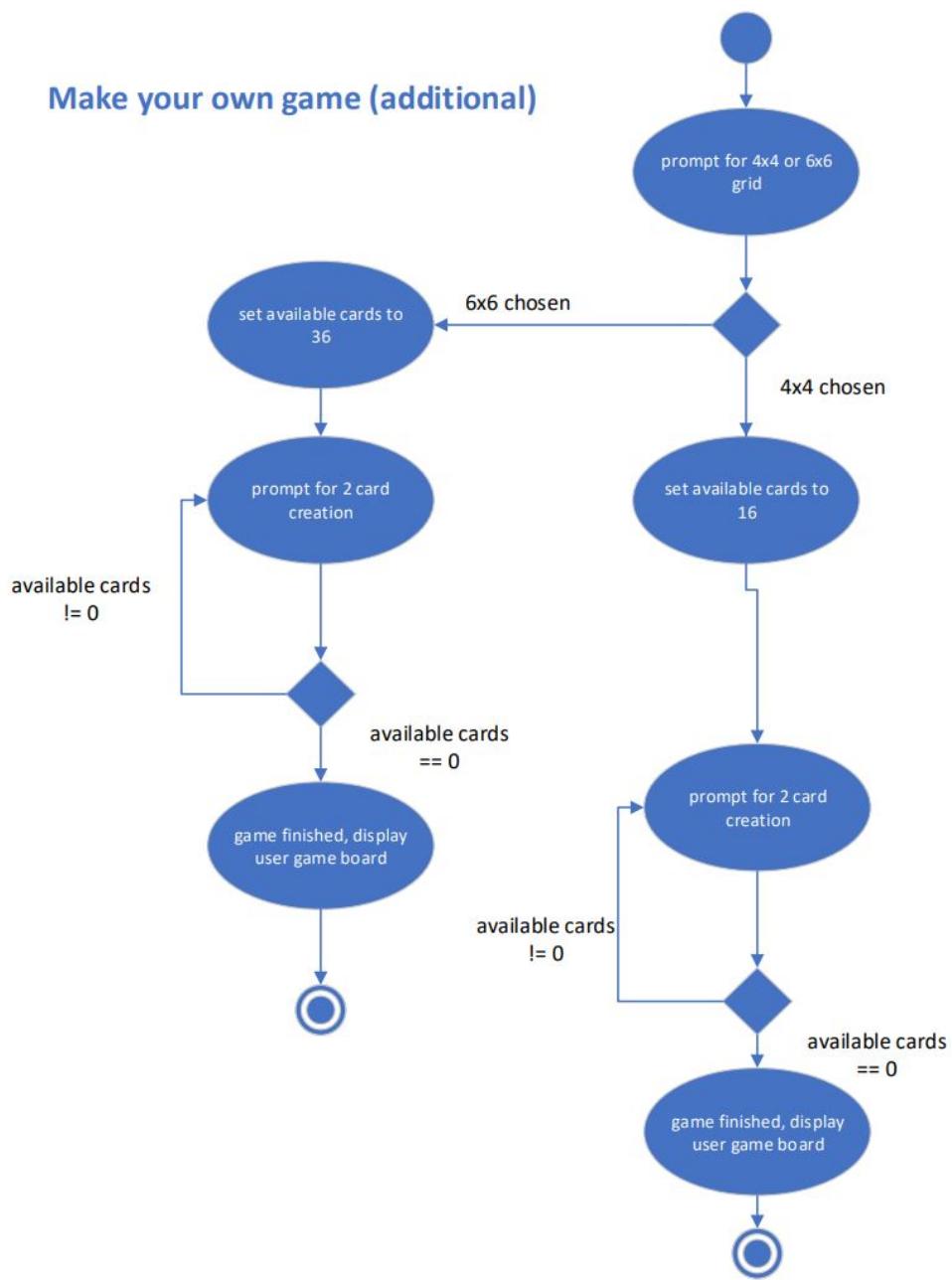
Housekeeping and error handling (4.1.10)



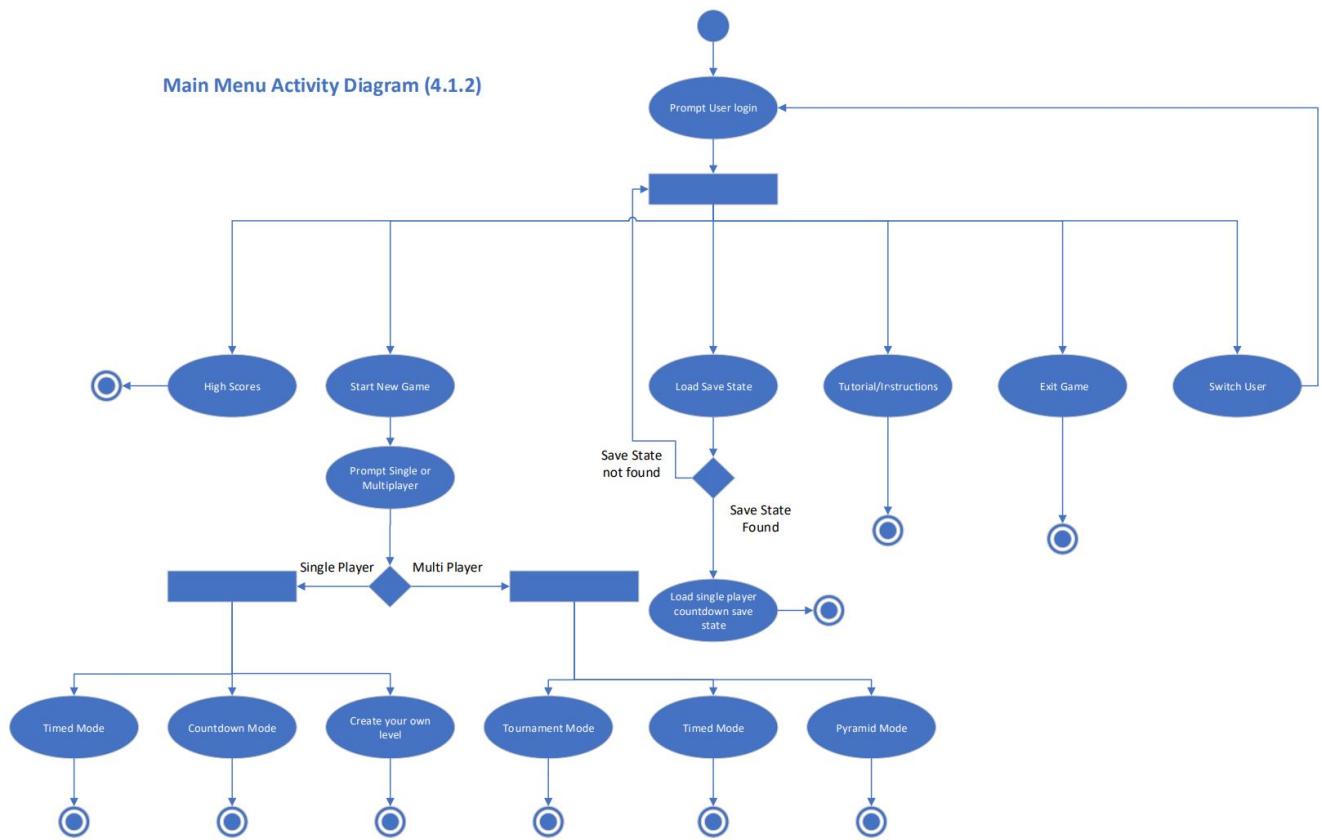
Selecting Difficulty Activity Diagram (additional)



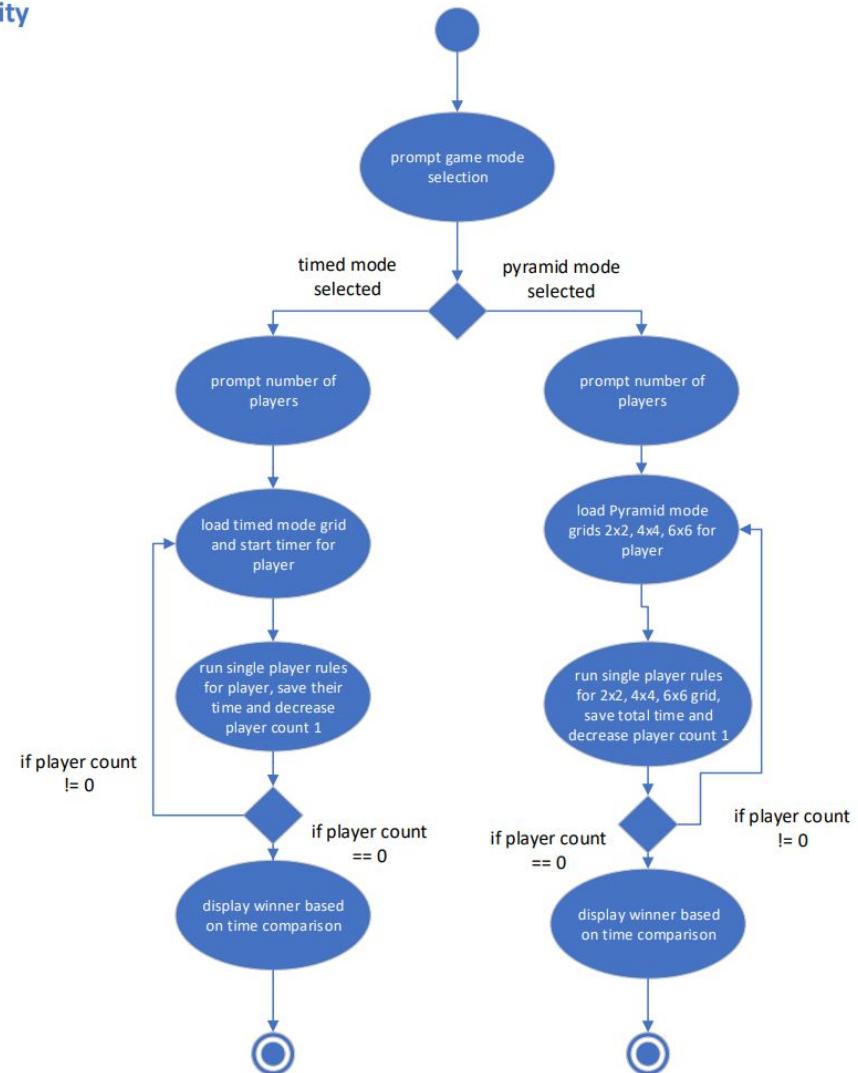
Make your own game (additional)



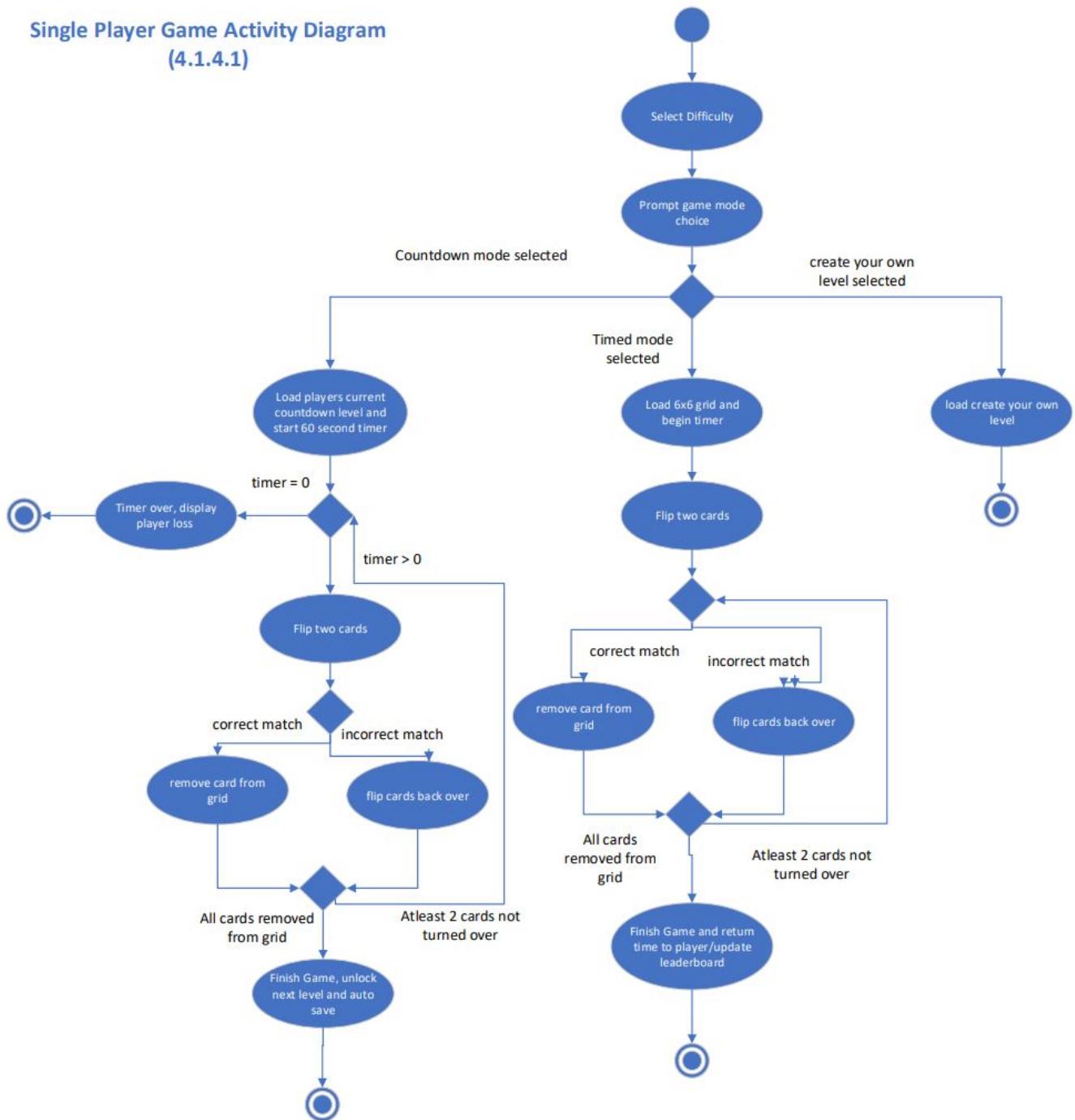
Main Menu Activity Diagram (4.1.2)



Multiplayer Game Mechanics activity
diagram (4.1.4.2)

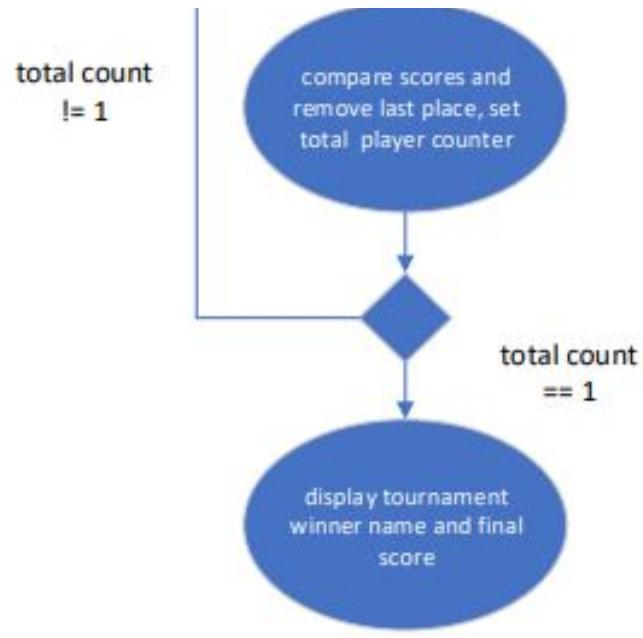


Single Player Game Activity Diagram
(4.1.4.1)



Tournament mode (additional)

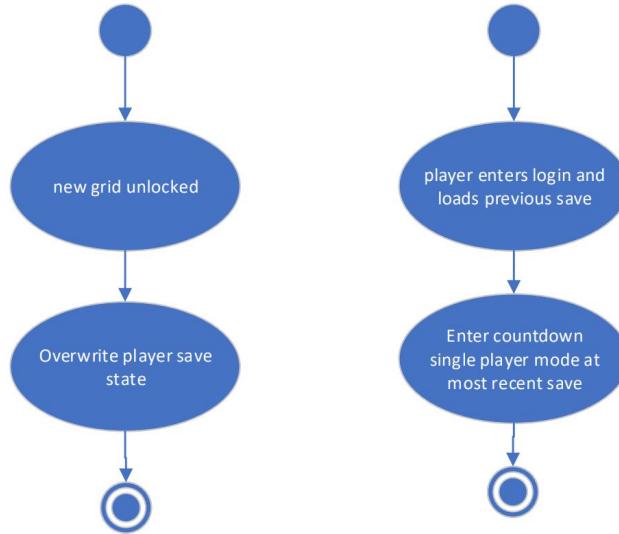


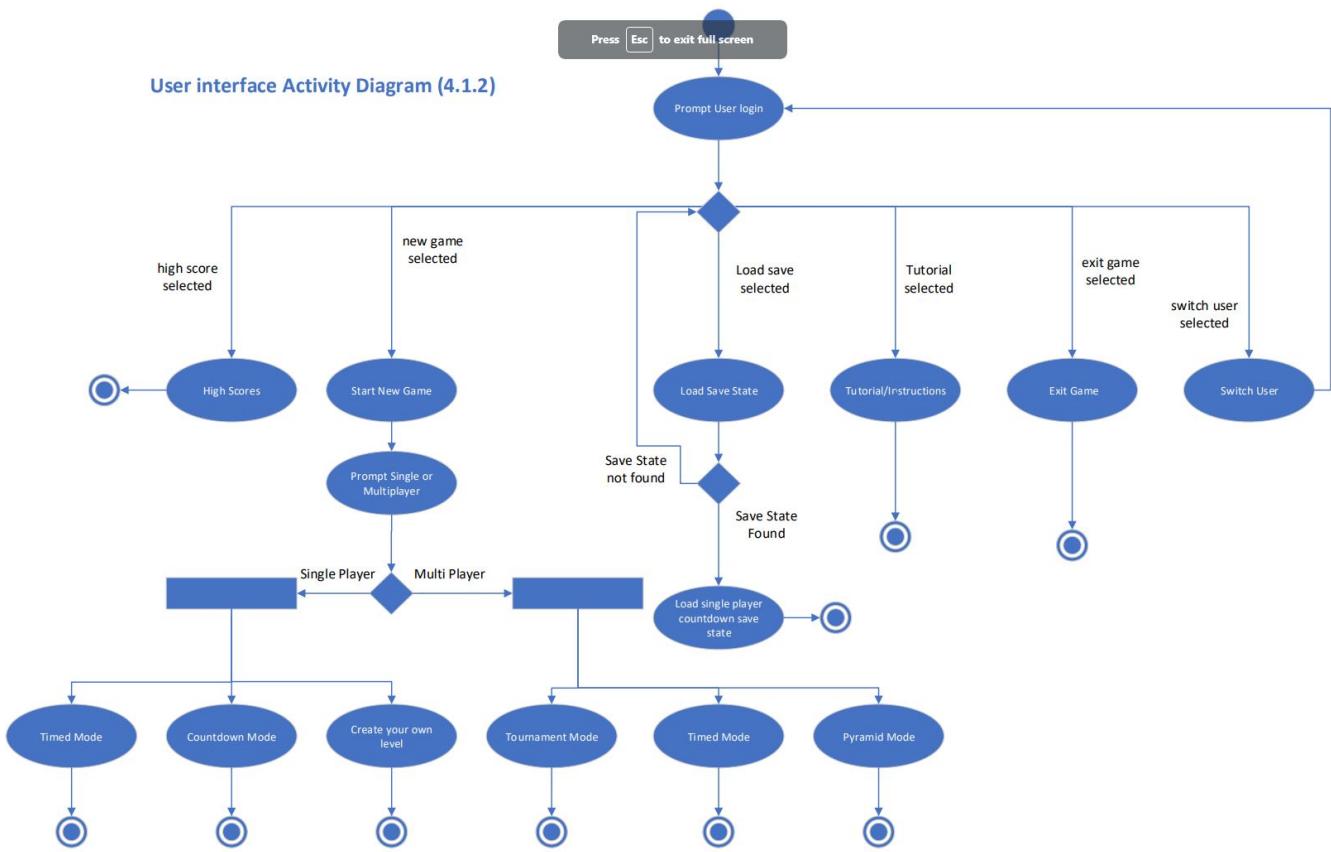


save state diagram

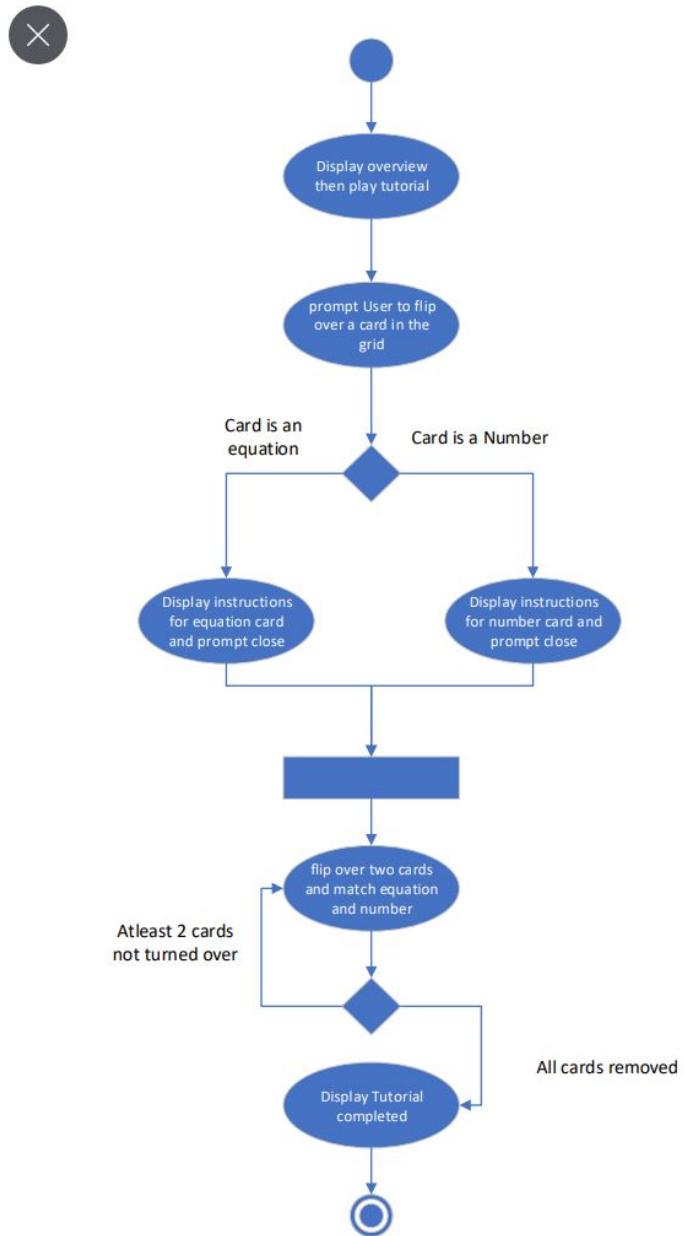
load state diagram

Save/Load Game State (4.1.5)





Tutorial Activity Diagram (4.1.3)



- **Use Case Diagram:** You must provide at least one (in total) [Use Case Diagram](#) that includes all of the actors and use cases in your system. You may split this diagram into multiple diagrams so long as it is clear how they are connected. Don't overcomplicate the diagram and try to keep it clear to stakeholders who may not be familiar with UML.

