Introduction

Overview:

Our educational game uses the captivating nature of gaming to promote learning by acting as a link between the dynamic world we live in and the complex knowledge we aspire to acquire. Our game takes advantage of the new opportunities that digital media presents to combine learning and entertainment, providing a stimulating and educational experience. Players are invited into an interactive environment that goes beyond traditional learning methods, using gamification to heighten motivation, focus, and retention of information.

Specifically, our game immerses players in a geographic discovery journey, presenting them with images of various locations and challenging them to guess correctly. Each successful guess not only awards points but also unlocks subsequent, more challenging levels, effectively balancing the act of learning with the joy of gameplay. The game features a straightforward yet user-friendly graphical user interface (GUI), mouse-based interactions, and keyboard shortcuts to improve accessibility and engagement. This approach makes the game more accessible to the intended audience.

As players navigate through the game, they encounter educational challenges that deepen their understanding of geography, including countries, regions, and significant landmarks. A unique aspect of our game is the integration of a hint system. Players receive a set number of hints initially and have the option to purchase more by spending their accumulated points, adding a strategic layer to the gameplay. The game's numerous modes, which include player-made question sets and themed challenges, make sure that learning about our world is always engaging and dynamic.

Objectives:

- Project Objectives:
    - Apply and demonstrate software engineering principles throughout the project lifecycle.
    - Manage the collaborative aspects of the project professionally, ensuring effective team dynamics and communication.
    - Adhere to and interpret a detailed specification to create models of requirements and design that align with project goals.
    - Engage in a continuous reflective process to assess and learn from design decisions, recognizing successes and areas for improvement.
- Software Objectives:
    - Develop and deliver a game that imparts educational value to the target audience, utilizing the appeal of geographical exploration.
    - Execute the design and development of the game using Java, ensuring the application is scalable, maintainable, and functional.
    - Craft a graphical user interface that is both simple and easily navigable, focusing on providing a smooth user experience.
    - Commit to writing robust, efficient, good-quality, clean, and well-commented code that follows best coding practices, enhancing the game's reliability and maintainability.
- Educational Objectives:
    - Enhance the players' knowledge and understanding of geographical concepts, emphasizing the diversity and uniqueness of locations worldwide.
    - Utilize the game's interactive platform to illustrate the variance in geographical features and cultural landmarks from different locations, facilitating a comprehensive and comparative learning experience.

References:

- Servos, D. *CS2212 Group Project Specification*. https://owl.uwo.ca/access/content/group/aa2311c9-4cef-497f-8d9c-b502023be21c/project/CS2212%20Group%20Project%20Specification.pdf
- De Freitas, Sara. "Are games effective learning tools? A review of educational games." *Journal of Educational Technology & Society* 21.2 (2018): 74-84.

Domain Analysis

Software domain:

- This project's software domain is educational gaming, with a focus on geography. The project's goal is to create an interactive game in which players have to guess the location of given images using geographical hints and landmarks. This entails combining gaming elements with educational objectives to create an engaging and immersive experience for players, thereby improving their geographical knowledge in an enjoyable way.

Target audience:

- The target audience for the game consists of individuals interested in a combination of gaming and learning, particularly those with an interest in geography. The game is designed to appeal to a diverse audience, ranging from casual players to those with some geographical knowledge. The difficulty levels are designed to provide flexibility to accommodate players with varying skill levels and preferences. Overall, the target audience consists of individuals who seek an interactive and educational gaming experience focused on enhancing their understanding of geographical concepts in an enjoyable manner.

Subdomains:

- Within the broader software domain of educational gaming focused on geography, this project can be classified into the subdomain of for example, "Geography Quiz Games." The game's main objective is to educate players about different geographical locations, landmarks, and landscapes through an interactive guessing game. The subdomain focuses on the specific educational aspect related to geography, distinguishing it from general educational games or games in other subject areas.

Domain description:

- Educational gaming is basically a dynamic connection of technology and learning, with the goal of providing users with engaging and interactive experiences. This domain focuses on a variety of subjects and uses the attraction of gaming to provide educational content, making the learning process more enjoyable and effective. Educational games frequently use gaming principles, such as quizzes, challenges, and adaptive difficulty levels, to cater to a diverse audience with varying levels of knowledge. With the rise of digital media, educational gaming has evolved into a powerful tool for promoting knowledge acquisition and skill development in an immersive and entertaining manner.

Commonly encountered issues:

- Balancing Entertainment and Education: Balancing engaging gameplay with educational content can be challenging. If the game is too focused on entertainment, educational objectives may be overlooked, and overly educational approaches may make the game less enjoyable.
- Adaptability to Diverse Learning Styles: Educational games must accommodate a wide range of learning styles and preferences. Ensuring that the game effectively addresses different levels of expertise, adjusts to individual pacing, and provides a meaningful learning experience for all players can be difficult.
- Technical Challenges: Common technical issues in educational game development include device compatibility, ensuring a smooth user interface, and addressing potential bugs or glitches. Balancing the technical challenges with the game's educational goals is essential for an enjoyable experience for users.

Common solutions to issues:

- Balancing Entertainment and Education:
  Solution: To balance both entertainment and education, integrate a well-designed curriculum into the game's narrative and gather player feedback to achieve an optimal balance.
- Adaptability to Diverse Learning Styles:
  Solution: implement adaptive learning algorithms, provide multiple learning paths, and update game content on a regular basis to appeal to a diverse variety of audience.
- Technical Challenges:
  Solution: Conduct extensive testing on a variety of devices and platforms to identify and address technical issues early in the development cycle. Regularly update the game to address bugs and improve compatibility. Use user feedback and beta testing to gain insight into technical issues and boost overall performance.

Using domain understanding to improve or accelerate development:

- By using insights from educational gaming, this project can strategically advance development. Prioritizing a balance of engaging gameplay and educational content is critical for making appropriate design decisions. Early use of adaptive learning techniques improves the game's adaptability to various learning styles, allowing it to appeal to a wide range of audiences. Incorporating effective assessment and feedback mechanisms ensures an efficient in-game feedback system that aligns with educational goals. Proactive management of technical challenges, including thorough testing and updates, ensures a consistent user experience. Continuous user engagement through feedback loops speeds up development, refines the game, and maximizes user satisfaction for a successful educational gaming experience.

Functional Requirements

Required:

Player Interface: The game will feature a player-friendly Graphical Player Interface (GUI) that is easily navigable by the target age demographic. This interface will effectively communicate the current state of the game to players and react properly to input.

- Multiple Screens: There will be five different pages in the game. Note: Some pages like the tutorial screen, gameplay screen, leaderboards screen, will have seperate sub pages. These include:
    - A main menu screen
    - A gameplay screen
    - An instruction or tutorial screen
    - A high-score list screen
    - A progress/results screen
- Mouse-Based Interaction: The interface will allow players to use the mouse to navigate through the game's screens, select items when needed, and manipulate specific elements of the game using the mouse. The game's core gameplay will be played with the mouse.
- Keyboard Shortcuts and Commands: The interface will incorporate keyboard shortcuts and commands for frequently used actions to enhance the player experience. Keyboard controls for gameplay actions will be integrated where applicable, ensuring a comprehensive and accessible gaming experience.
- Feedback Systems: The game's UI delivers either visual or auditory feedback following player interactions, like button clicks or command inputs, to affirm that the intended action has been successfully executed. For example, there will be a specific sound that is made whenever the player chooses a correct or incorrect answer.

Main Menu: The main menu will be the first screen of the game. It will provide a list of these options for the player:

- Start a new game: Starting from the first level. Players will need to enter a username when starting a new game.
- Load a previously saved game: The player will have the option to load any saved game of their choice to continue.
    - Note: The saved game will include their previous records (Score, wins, losses, number of hints used, etc).
- Tutorial: The game will incorporate the option of showing the player basic instructions on how to play the game, including:
    - Controls and navigation: Players will be introduced to the basic controls, like how to move, interact with the environment, etc.
    - Shortcuts: The tutorial will also showcases list of written shortcuts, which will allow players to quickly grasp time saving and efficient way to navigate or performing game actions.
- Leaderboard: This will showcase the highest scores in order.
- Instructor dashboard: Allows instructor to monitor player progress.
- Debug Mode: (see debug mode requirement for details)
- Quit game: Exit the game
- The main menu will also display a background visual of Earth, a list of the developers, our team's number (43), the term it was created in (Winter 2024), a mention that this game was created for CS2212 at Western University.

Instructions and/or Tutorials:

- Players will be able to go to a tutorial screen from the main menu, which will provide the player with detailed instructions on how to play the game.
- This will be an interactive tutorial:
    - A practice question will be presented to the player
    - They will be guided through the geographic images, hint and answer options, and how saving their game works.
    - By doing so, the Player can have a good understanding of how the game works before actually playing it.

Game Play and Mechanics:

- Scoring System:
    - Points will be awarded to the player for each correct answer to a presented question.
    - The software will track player scores and display them to the player, updating scores as needed.
- Progression Mechanics:
    - Multiple levels of difficulty are available for the Player to progress through as their skills and knowledge improves.
    - Difficulty levels will unlock as the player progresses through easier levels.
    - The player will begin with only one level available.
    - Each level will consist of five questions and the player must answer at least four correctly to unlock the next level.
- Educational Challenges:
    - Require players to have a good understanding of geography, including knowledge of countries, regions, landscapes, and landmarks.
    - Present players with a random location and ask them to identify the location from multiple choices.
    - Challenge players to match landmarks to their country or city.
- Response to Player Input:
    - Player interactions with the game, using mouse clicks or keyboard inputs, will trigger corresponding actions within the game environment. These could include selecting options, maneuvering elements within the game, or moving through different layers of content.
    - The game will implement auditory cues when entering commands and clicking buttons to give affirmation that the input has been registered.
- Feedback for Correct/Incorrect Actions:
    - Players will be given immediate feedback after answering a question.
    - When the player inputs a correct answer, an encouraging message will be displayed along with a green checkmark to assure the player.
    - When the player inputs a incorrect response a consolation message will be presented along with encouragement to try again and a red X mark.

- Points system:
    - Every time the Player guesses the correct location, the system will reward them with some points.
    - The amount of points a player receives will vary based on the difficulty.
- Lives or Attempts:
    - Players will be given one attempt to answer questions.
    - After answering each question, players will be notified of whether their answer was correct or incorrect.
    - To complete a level, players must obtain the minimum score.

Save/Load Game State:

- Implement a Save Game State feature to capture and store the player’s last completed level.
- Enable players to resume gameplay from the latest checkpoint without needing to save every gameplay detail.
    - Save essential progress information, such as level number or stage, to streamline the save state.
- Provide options for automatic saving at checkpoints or manual saving.
- Ensure the load function restores the player to a logical point for seamless continuation of learning and play.
- Display a clear confirmation message post-save to confirm successful progress recording.
- Design the save system with multi-player functionality in mind:
    - Each player should have independent save capabilities.
    - Consider multiple save slots or a user/account system for individual progress tracking.

High Score Table (Leaderboard):

- Integrate a High Score Table to track and display top player scores.
- The High Score Table will foster a sense of achievement and provide a performance benchmark.
- Each entry in the leaderboard will have the player's name or initials and their score.
- The leaderboard will be accessible from the main menu.
- The table will store and show the top 5 scores and will be persistent across game restarts.
- It will be updated automatically as new high scores are achieved.
- A file in the local file system will store the high score data.

Multiple Players:

- The game will support offline multiple players functionality.
- The players will be required to enter a username when starting a new game.
- This name will be used in the high score table, saved in the save state, and it also will require the player to enter the username when trying to load a previously saved state.

Instructor Dashboard: The game will have a password-protected instructor mode (accessible from the main menu) that will display metrics to an instructor of each student in the game. These metrics will include:

- Student's current score
- Student's current level
- What levels the student has already completed
- Number of attempts per level

Debug Mode / Level Selection: This mode will be password-protected. The password will only be provided to testers and developers. This mode will enable them to:

- Unlock all levels, allowing them to navigate to any level without needing to complete the levels preceding it.
- Create, Edit or delete save states.
- Edit levels (for example, changing the options player can choose from):
    - The tester or developer has the ability to create their own levels which can involve different themes, levels, or objectives.
    - The developer can also customize control settings to match their preferences. This can include remapping keys, or choosing between different control schemes
- Adjust number of points and hints.

Housekeeping & Error Handling:

- Implement mechanisms to ensure clean exit from the application.
- Ensure all data, including save states and instructional metrics, is correctly saved upon exit for persistence in future sessions.
- The game will run at a fixed window size. Window resizing will not be permitted.
- Invalid inputs into username creation and password entry (debug/instructor mode) will be caught by the code and will throw an error on screen.
- Duplicate usernames will not be allowed. The application will throw an error if a duplicate is found, requiring them to pick a different username.

Additional:

Sound and music:

- Implement system where different scenarios trigger specific sounds
- The game will feature background music.
  - Menu Music: Music will be played in the menus before the Player chooses a mode to play. Players will have the option to change the music from a list of 5-10 different sounds.
  - Gameplay Music: Different levels will have different music playing in the background. This will be selected by the developers before release of the levels and players will be unable to change the music.

Hints System:

- Hints will be available during gameplay to assist players when they are unable to figure out a specific location.
- A set amount of hints will be provided at the beginning of each turn.
- Players can choose to use the given hints at any time in the game.
- There will be a limit of 1 hint per question.
- Once the player has spent all of their available hints, a "price system" based on points will be implemented. The player can either:
  - Continue the rest of the game without any hints, playing until they complete the game.
  - Use points that they have acquired thus far in the game (from their score) to purchase hints. This will reduce the player's score. There is no limit to the amount of hints a player can purchase. However, as players spend more on purchasing hints, their score will be suffer more deductions.

Theme based game mode: This is an extra mode where players will be tested based on a chosen theme. Some examples of the available themes are:

- <u>Landmarks:</u> This challenge will presented with landmarks (significant/ important feature of a place) from a location, which they will need to guess.
- <u>Languages:</u> The player will attempt to guess the location based on the given official language.
- <u>Cultural/ traditional practices:</u> The player will be given a cultural practice that is specific to a location and will need to guess based on this.

Fun Facts:

- After each question is completed (the Player submits an answer regardless of its correctness), the player will be presented with some fun facts about the correct location in the image along with the correct answer.
- This is done to ensure that players are educated properly about the locations they encounter through gameplay.
- Otherwise, the only take-away for players would be images of locations that they encounter.

Player-Created Questions Sets: Players will be able to create their own sets of questions, similar to the main gameplay mode. This can be used as a form of practice or to test friends on their knowledge of locations. The player creating the questions will be able to:

- Upload a maximum of 10 questions per set.
- Select a theme for the questions (if applicable).
- Upload an image and answer options (one correct option) for their questions.

Actors

Player

Description	The Player person who interacts with the game via the GUI, using mouse-based interaction and keyboard shortcuts to navigate through different screens, respond to game challenges, and receive feedback on their actions.
Aliases	Gamer, Student, Participant
Inherits	None
Actor Type	Person
Active /Passive	Active

Instructor

Description	Instructor is an individual who utilizes the instructor dashboard to monitor the progress and performance of students playing the game, accessing metrics such as scores, levels completed, and time spent.
Aliases	Teacher, Educator
Inherits	Player
Actor Type	Person
Active /Passive	Active

Developer

Description	A software developer who is responsible for building and maintaining the game. They have the ability to create levels, customize control settings, and manipulate various aspects of the game during the development process. This also includes ongoing game maintenance like fixing bug, adding new content or features, and fine tune control mechanisms.
Aliases	Game/level developer, game designer
Inherits	Player
Actor Type	Person
Active /Passive	Active

Tester

Description	A Tester is a individual who is focused on testing the game for bugs and user experience issues. They have access to a debug mode to thoroughly test all aspects of the game, including gameplay mechanics, level progression, interface design and overall game balance. In addition to the technical aspect, the tester also ensures the clarity of the in game menus and the responsiveness of the game mechanics.
Aliases	QA Tester
Inherits	Player
Actor Type	Person
Active /Passive	Active

Use Cases

Changing Background Music

Name	Change menu background music
Primary Actor	Player
Secondary Actors	Audio System
Goal in Context	The player wants to customize the background music played in the main menu, according to their preferences. This use case allows the player to select a different background music track from the available options provided in a audio popup menu.
Preconditions	• The player will be in the main menu • The player will select the option to change the background music and will select an audio from the provided audio options, in a popup.
Trigger	The player selects the option to change the background music
Scenario	1. The player selects the "Change Music" option from the main menu. This will open a small popup menu. 2. The popup presents the player with a list of available background music tracks. 3. The player selects the desired background music track from the list. 4. The game updates the background music to the selected track. 5. The game confirms the successful change of background music to the player.
Alternatives	None
Exceptions	• If the audio selected by the player is not working, the game will print an error message
Priority	Medium

Using hints

Name	Use hint
Primary Actor	Player
Secondary Actors	None
Goal in Context	To provide the player with additional clues or information to help identify the game's current location by using a hint.
Preconditions	• The player is engaged in a level where extra hints may be helpful. • The player has either unused hints available or enough points to purchase additional hints. • The game's hint system is accessible and operational.
Trigger	The player selects the option to use a hint.
Scenario	1. The player is unable to determine the location and selects to use a hint. 2. The system checks if the player has an available free hint for this location. 3. If a free hint is available, the system provides the hint without point deduction. 4. If no free hints are available, the system prompts the player to confirm the purchase of a hint using their points. 5. The player confirms the purchase. 6. The system checks if the player has sufficient points. 7. If the player has sufficient points, the system deducts the appropriate number of points and provides the hint. 8. If the player does not have sufficient points, the system displays a message informing the player of insufficient points and returns to the question.

Alternatives	• The player decides against purchasing a hint after the point deduction prompt. • The player opts to continue the game without using hints after expending the free ones.
Exceptions	• The player attempts to use more than one hint per question. • The system is unable to deduct points due to an error. • The system fails to display the hint after successful point deduction.
Priority	High

Accessing Instructor Dashboard

Name	Accessing instructor dashboard
Primary Actor	Instructor
Secondary Actors	None
Goal in Context	To provide the instructor with a password-protected overview of student metrics, including scores, levels completed, and time spent.
Preconditions	The instructor has a valid password and the game is installed and running.
Trigger	The instructor selects the "Instructor Dashboard" option from the main menu.
Scenario	1. The instructor selects the "Instructor Dashboard" from the main menu. 2. The instructor is prompted to enter a password. 3. Upon entering the correct password, the instructor is granted access to the dashboard. 4. The dashboard displays a list of students, their scores, levels completed, and other relevant metrics.
Alternatives	If the instructor enters an incorrect password, they are given the option to try again or to reset the password if they have forgotten it.
Exceptions	If the system fails to verify the password, an error message is displayed.
Priority	Medium

Accessing Theme Based Modes

Name	Access Theme Based Modes
Primary Actor	Player
Secondary Actors	None
Goal in Context	The goal is to allow the player to select and enter a game mode that is centered around a specific theme, such as landmarks, languages, or cultural practices.
Preconditions	The game is installed and running, the player is on the main menu, and theme-based modes are available for selection.
Trigger	The player selects the "Theme Based Modes" option from the main menu.
Scenario	1. The player launches the game and arrives at the main menu. 2. The player clicks on the "Theme Based Modes" option. 3. The game presents a list of available themes. 4. The player selects a theme of their interest. 5. The game loads the selected theme-based mode for the player to start playing.
Alternatives	None, as this is a direct selection without alternate pathways.

Exceptions	If the game fails to load the selected theme, an error message is displayed, and the player is returned to the main menu.
Priority (1-10)	High

Submitting Custom Question Sets

Name	Submit player created question set
Primary Actor	Player
Secondary Actors	None
Goal in Context	To enable players to submit their custom question sets for gameplay, promoting interaction and personalized gameplay experiences.
Preconditions	The player has created a question set and is logged into the game.
Trigger	The player selects the option to submit their question set.
Scenario	1. The player selects the "Create Question Set" option from the menu. 2. The player creates questions, uploads images, and provides answer options. 3. The player reviews their question set for accuracy. 4. The player submits the question set for inclusion in the game.
Alternatives	None
Exceptions	If there are issues with the submission, such as invalid data or server errors, the player is notified and prompted to correct the issues.
Priority	Low

Accessing Debug Mode

Name	Access debug mode
Primary Actor	Developer
Secondary Actors	Tester
Goal in Context	To provide developers and testers with access to a debug mode for game testing and level editing.
Preconditions	The developer/tester has the game running and possesses the correct password.
Trigger	Developer/tester enters a special key combination to access debug mode.
Scenario	1. The developer/tester starts the game. 2. They enter the key combination to access debug mode. 3. The game prompts for a password. 4. The developer/tester enters the correct password. 5. The game grants access to debug mode with all levels unlocked.
Alternatives	If the wrong password is entered, the developer/tester is prompted to try again.
Exceptions	If the system fails to authenticate the user, access to debug mode is denied.
Priority	High

View leaderboard

Name	View Leaderboard
Primary Actor	Player
Secondary Actors	None
Goal in Context	To provide players with the ability to view the High Score Table, displaying the top scores achieved.
Preconditions	The game has been played by one or more players, resulting in recorded scores.
Trigger	The player selects to view the High Score Table from the main menu.
Scenario	1. The player is on the main menu. 2. The player selects the option to view the high score table. 3. The system retrieves the high score data from the local file storage. 4. The system displays the High Score Table, showing at least the top 5 player scores with names or initials. 5. The system updates the table automatically if a new high score is achieved, replacing the lowest score if necessary.
Alternatives	Not applicable, as viewing the High Score Table does not involve alternate flows.
Exceptions	• The high score data file is corrupted or unreadable, preventing the display of scores. • There are fewer than 5 scores (or no scores at all) to display.
Priority	Medium - Useful for player engagement but not critical for basic gameplay functionality.

Load saved game

Name	Load saved game
Primary Actor	Player
Secondary Actors	Game system/database
Goal in Context	The primary goal of this is to provide the player with a seamless and efficient process to load a game that was previously saved with the same in game attributes. The system should authenticate the player, present all available saved games, and allow the player to choose and load the desired game state they want. Then, once the user has chose the game they want to load, the system should maintain the integrity of the saved game data and allow the player to resume exactly where they left off in terms of game progress, player stats, inventory, and any other game data.
Preconditions	▪ The player has previously created an account and has saved the game ▪ The database containing the saved game is available
Trigger	The player selects the 'Load Game' option from the main menu.
Scenario	1. The system displays the main menu for the game 2. the player select the load game option 3. The system prompts the player to enter in their username and password 4. The player has to enter valid credentials. However, if the credentials are not valid, then the player will be prompted to re-enter the correct credentials to proceed 5. Then, once authenticated, the system will retrieve the saved games associated with the player's account 6. If the player has multiple saved games, the system will open the game library 7. The player then selects the game they want to load. 8. The system loads the selected game and displays its details. These details include the current level, achievements, and unlocked content. 9. The system confirms that the game load is successful 10. The player chooses to continue the game
Alternatives	▪ If the player has only one saved game, the system will automatically load it without going to the game library ▪ If the player enters the wrong credentials, then the system prompts them to try again ▪ If the game ever fails to load, then the system offer options to retry or return to the main menu

Exceptions	▪ In the event the database is not accessible, loading will not proceed, and the player is informed of the system error. ▪ If the saved file is found unreadable, the system will notify the player and provide options to either delete the saved game, try again to load in the game, and if the loading is still not successful, a error message stating "Error: Game cannot open" will be displayed
Priority	High as this is a critical feature for player retention and engagement

Access tutorial options

Name	Access tutorial options
Primary Actor	Player
Secondary Actors	Tutorial system and the game database
Goal in Context	To provide the player with interactive learning resources that teaches them how the game works in depth. Options include, interface navigation, gameplay controls, scoring system, settings customization, and additional learning modes.
Preconditions	The game is started, and the main menu is accessible
Trigger	The player select the "View Tutorial" option from the main menu interface
Scenario	1. The system displays the main menu. 2. The player selects the 'View tutorial' option. 3. The system presents the tutorial options menu to the player. 4. The player then will have the ability to choose one of the following tutorial sections: • User Interface Walkthrough • Controls Explanation and How to Play • Scoring System and Progress Tracking • Settings and Customizations • Shortcut Details • Quiz or Practice Mode Explanation 5. Once the player has picked some starting point, the system provides the selected tutorial content to the player. 6. The player interacts with the tutorial content and has the option to return to the tutorial options menu. 7. After completing or exiting the tutorial, the player is finally taken back to the main menu where they choose the other options or start playing the game
Alternatives	The player can simply choose to exit the tutorial at any time or select another section without completing the current one
Exceptions	If the system fails to load the tutorial content, then the player is informed of the error and provide options to retry or return to the main menu
Priority	Medium to high because the tutorial is a key feature for players to help them understand the game mechanics, which in turn can enhance their gameplay experience. Bu having a tutorial that is well structured, it can help to reduce the learning curve and facilitate a smoother gameplay experience. This is important for the harder level in the game where players are often faced with more complex challenges that require a deeper understanding of the game mechanics.

Level selection option

Name	level selection
Primary Actor	Player
Secondary Actors	Game system
Goal in Context	To allow players to select and play a level within the game while still ensuring they only access levels that they have reached according to the game's progression rules
Preconditions	▪ The game is started, and the main menu is displayed ▪ The player has access to levels with respect to their progression

Trigger	The player chooses to select a level from the main menu
Scenario	1. The main menu is displayed. 2. The player chooses the level selection option. 3. The system displays the current level, completed levels, or all levels available to the player. 4. The player selects a difficulty level (n). 5. The system checks if the current level n is completed: • If the level is completed, the system then allows the player to select difficulty level (n). • If the level is not completed, the system prompts the player to complete the current level. 6. The player plays the level again or chooses to play more levels. 7. After playing: • If the player completes the level, the options menu is displayed. • If the player does not complete the level, the continuation menu is displayed. 8. The player chooses to end the game or continue playing: • If ending the game, an end screen pops up. • If continuing, the player can go back to the level selection or the main menu.
Alternatives	The player can adjust the mode from choosing levels to practice mode at the main menu
Exceptions	If the player tries to access a level beyond their progression list, they are notified with a message that states "Cannot proceed to the level, please successfully complete the preceding level". Hence, they cannot proceed.
Priority	High since level selection is a fundamental aspect of gameplay and player autonomy. It is high level priority because it includes a strict progression level system that ensures and prevents the player from bypassing the games naturaly difficulty escalation.

Start a new game

Name	Starting a new game
Primary Actor	Player
Secondary Actors	Game system
Goal in Context	To start a new game
Preconditions	▪ The player is at the games main menu, as this is the interface where they can start a new game session
Trigger	The player selects the start new game option from the main menu
Scenario	1. The system displays the main menu to the player. 2. The player has the following options: • View Tutorial: The player can choose to view a tutorial explaining game mechanics. • Load Saved Game: The player can load a previously saved game. • View Leaderboards: The player can view the game's leaderboards. • Quit Game: The player can choose to exit the game. • Start new game • Challenge mode 3. If the player selects 'Start New Game': • The player is prompted to enter a username and password • The system checks if the username is valid and not taken. • If the username is invalid or taken, the player is prompted to enter a different username. 4. Once a valid username is entered, the game starts at the respective starting level (n). Note: When the user first starts the game, they will begin on level 1 automatically. 5. The player attempts to complete level n: • If completed successfully, the player moves on to level n + 1. • If not completed successfully, the player has the option to use a hint. 6. If the player chooses to use a hint: • The system checks if a free hint is available. • If a free hint is available, it is given to the player. • If no free hint is available, the player can choose to buy one with points. 7. The player continues to play until: • The game is finished. • The final level is reached. • The player chooses to return to the main menu or quit the game.

Alternatives	N/A
Exceptions	If starting a new game overwrites existing data that is saved, then the system will provide a warning message to the player and notify them if they want to try again, or proceed.
Priority	The priority is high because beginning a new game is an essential first step for anyone playing the game. If there are any problems during this stage, it could really spoil the fun for the player right from the start.

Answering Questions

Name	Answering Questions
Primary Actor	Player
Secondary Actors	None
Goal in Context	Give players the ability to answer the question on screen during a level.
Preconditions	• The player is in a game level • The player has enough attempts and hasn't failed out of the level yet.
Trigger	The player selects an option on screen.
Scenario	1. The player is in the game screen and in a level 2. The player is asked a question 3. The player is given different options to choose, their selection will be their answer 4. Player selects an answer
Alternatives	The player quits the game
Exceptions	None
Priority	High

Player Attempts

Name	Player Attempts
Primary Actor	Player
Secondary Actors	None
Goal in Context	Give players one attempt or chance to answer a question before they fail the level.
Preconditions	• The player is in a game level • The players hasn't failed out of the level yet.
Trigger	The player selects an answer option on screen.
Scenario	1. The player is in the game screen and in a level 2. The player is asked a question 3. The player is given different options to choose, their selection will be their answer 4. Player selects an answer 5. If the answer is wrong, player fails the level.
Alternatives	The player quits the game
Exceptions	None
Priority	High

Signing in as a Player

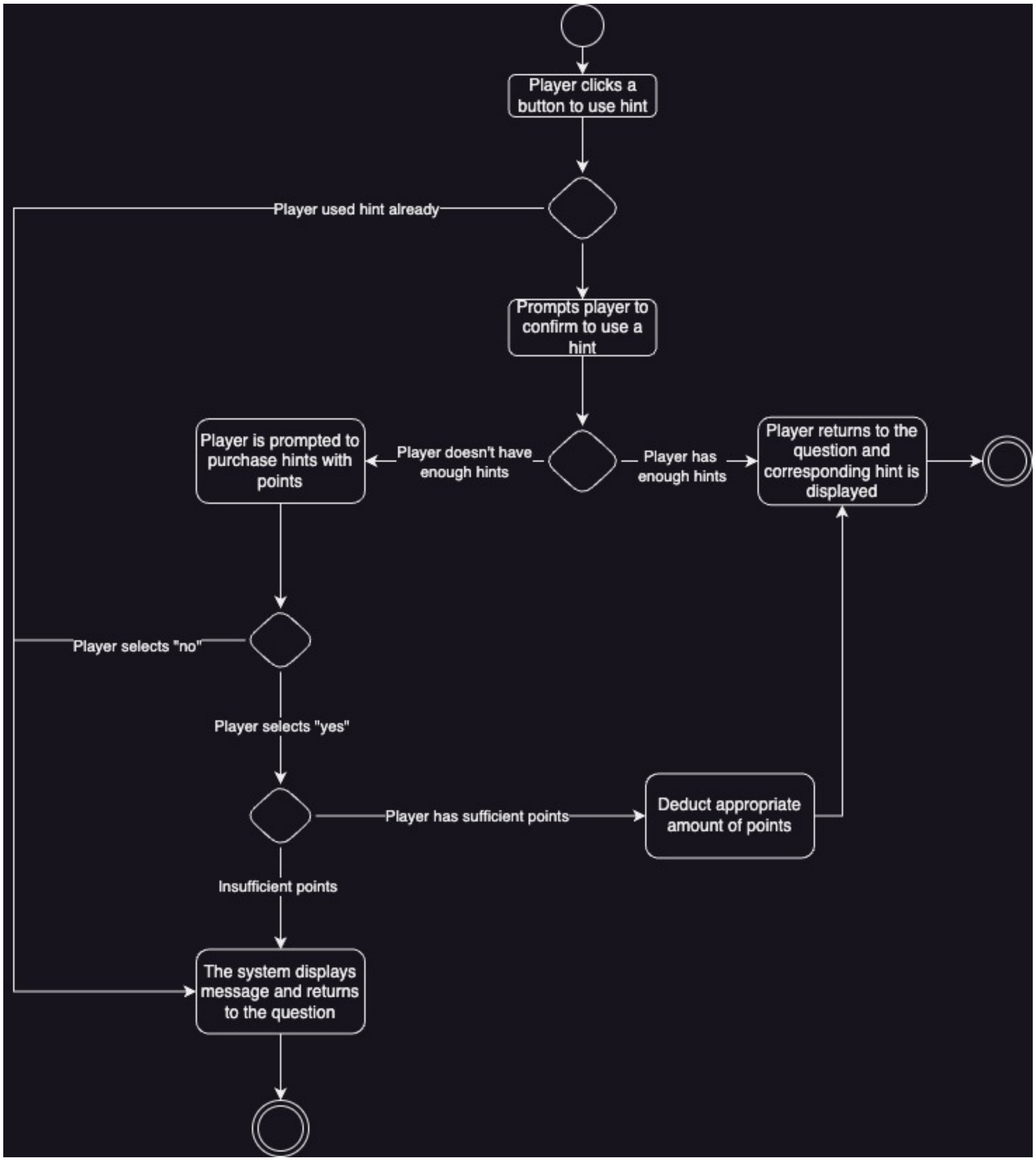
Name	Player Sign in
Primary Actor	Player
Secondary Actors	Database of player information
Goal in Context	Allow players to log in to their own account, in which they will have their own experience.
Preconditions	• The player has opened the game • The player has an account • The player isn't already signed in
Trigger	The player enters their account information to sign in
Scenario	1. The player opens the game 2. The game prompts them to input their sign in information 3. If the information is valid, they are signed in
Alternatives	The player quits the game
Exceptions	• The player inputs invalid input, such as invalid characters • The information the player inputted is incorrect, there is no account in the database with that info
Priority	High

Quitting the game

Name	Quit Game
Primary Actor	Player
Secondary Actors	None
Goal in Context	To allow the player to exit the game safely and ensure that their progress is saved before quitting.
Preconditions	The player currently has an instance of the game open.
Trigger	The player selects the "Quit Game" option from the game menu.
Scenario	1. The player opens the game menu during a session. 2. The player selects the "Quit Game" option. 3. The system prompts the player to confirm if they want to quit and save the game. 4. The player confirms the action. 5. The system saves the current game state. 6. The system closes the game application.
Alternatives	The player cancels the action when prompted to confirm quitting.
Exceptions	The game fails to save the current state due to an error, in which case the player is notified of the failure before the application closes.
Priority	High - Essential for a good player experience and game data integrity.

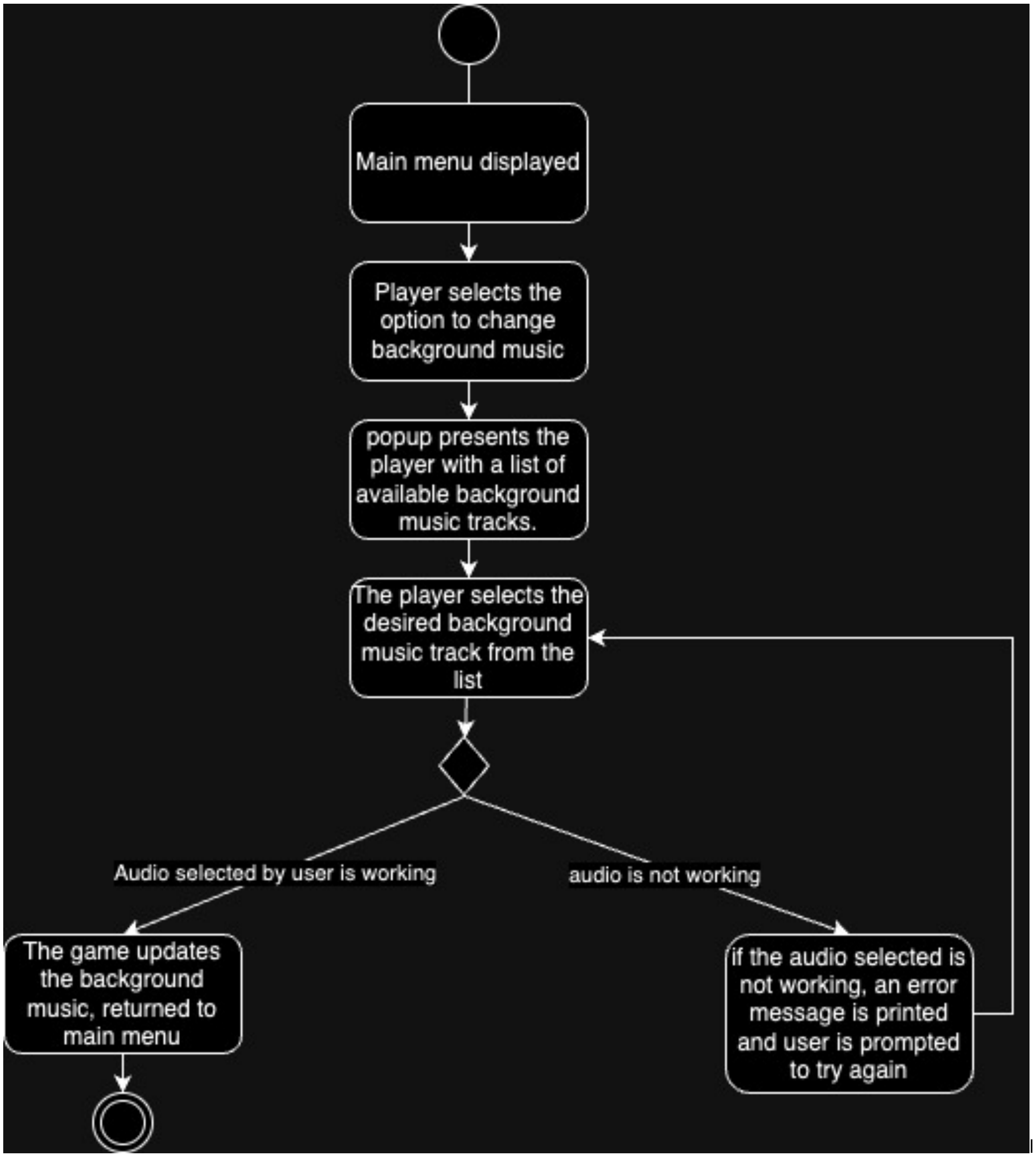
Activity Diagrams

"Use hint"

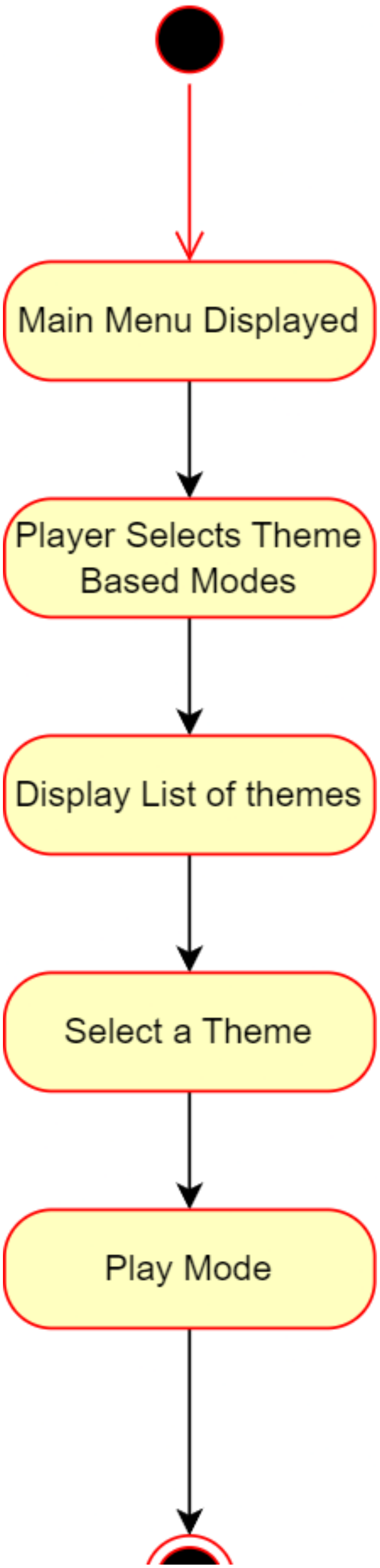


Player clicks a button to use hint
Player used hint already
Prompts player to confirm to use a hint
Player is prompted to purchase hints with points
Player doesn't have enough hints
Player has enough hints
Player returns to the question and corresponding hint is displayed
Player selects "no"
Player selects "yes"
Player has sufficient points
Deduct appropriate amount of points
Insufficient points
The system displays message and returns to the question

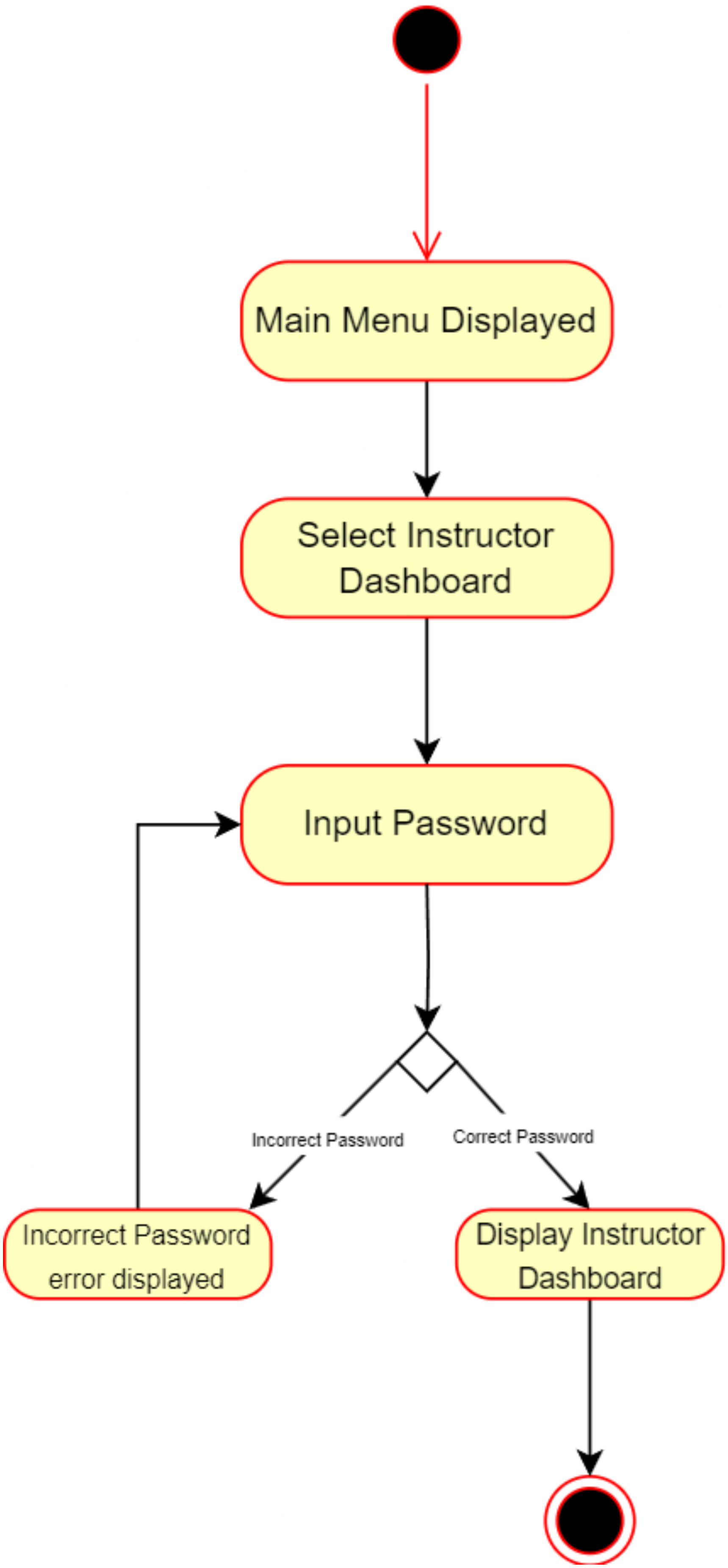
"Change menu background music"


Main menu displayed
Player selects the option to change background music
popup presents the player with a list of available background music tracks.
The player selects the desired background music track from the list
Audio selected by user is working
audio is not working
The game updates the background music, returned to main menu
if the audio selected is not working, an error message is printed and user is prompted to try again

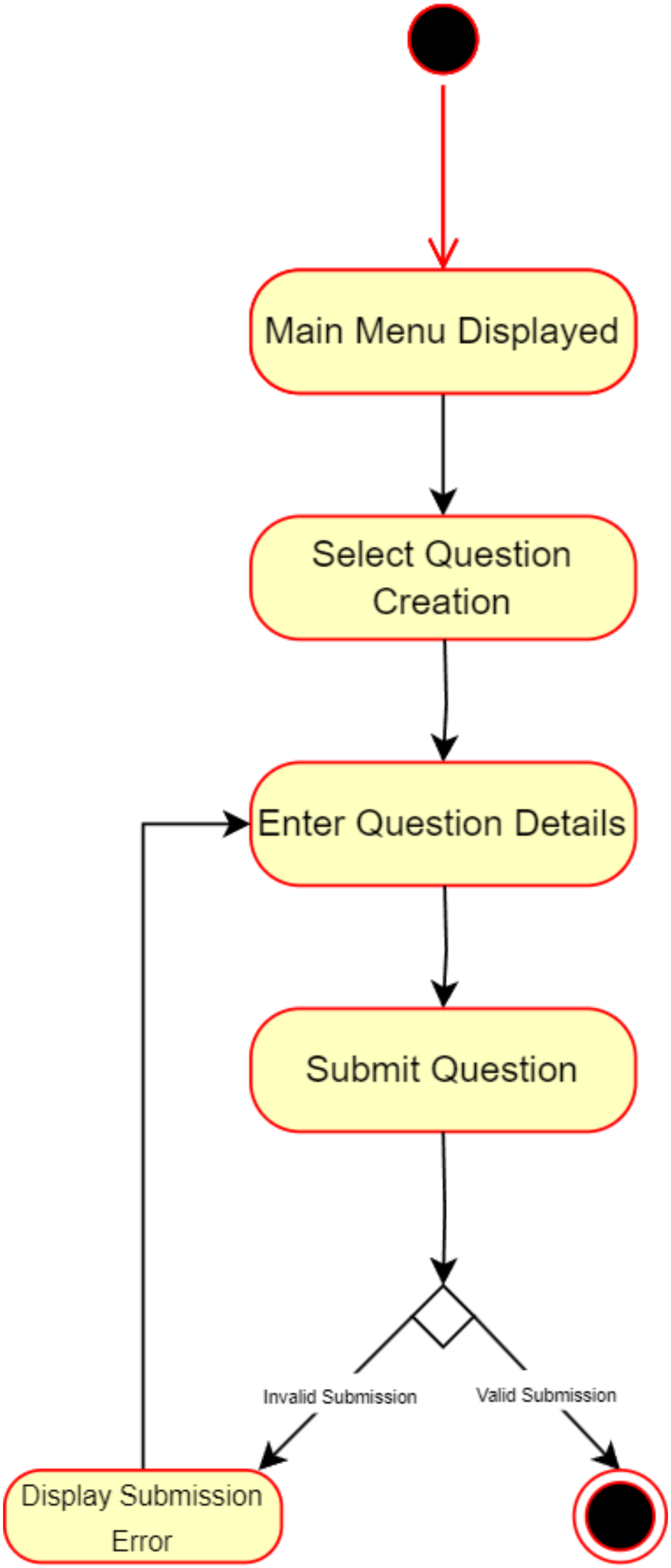
"Access Theme Based Modes"


Main Menu Displayed
Player Selects Theme Based Modes
Display List of themes
Select a Theme
Play Mode

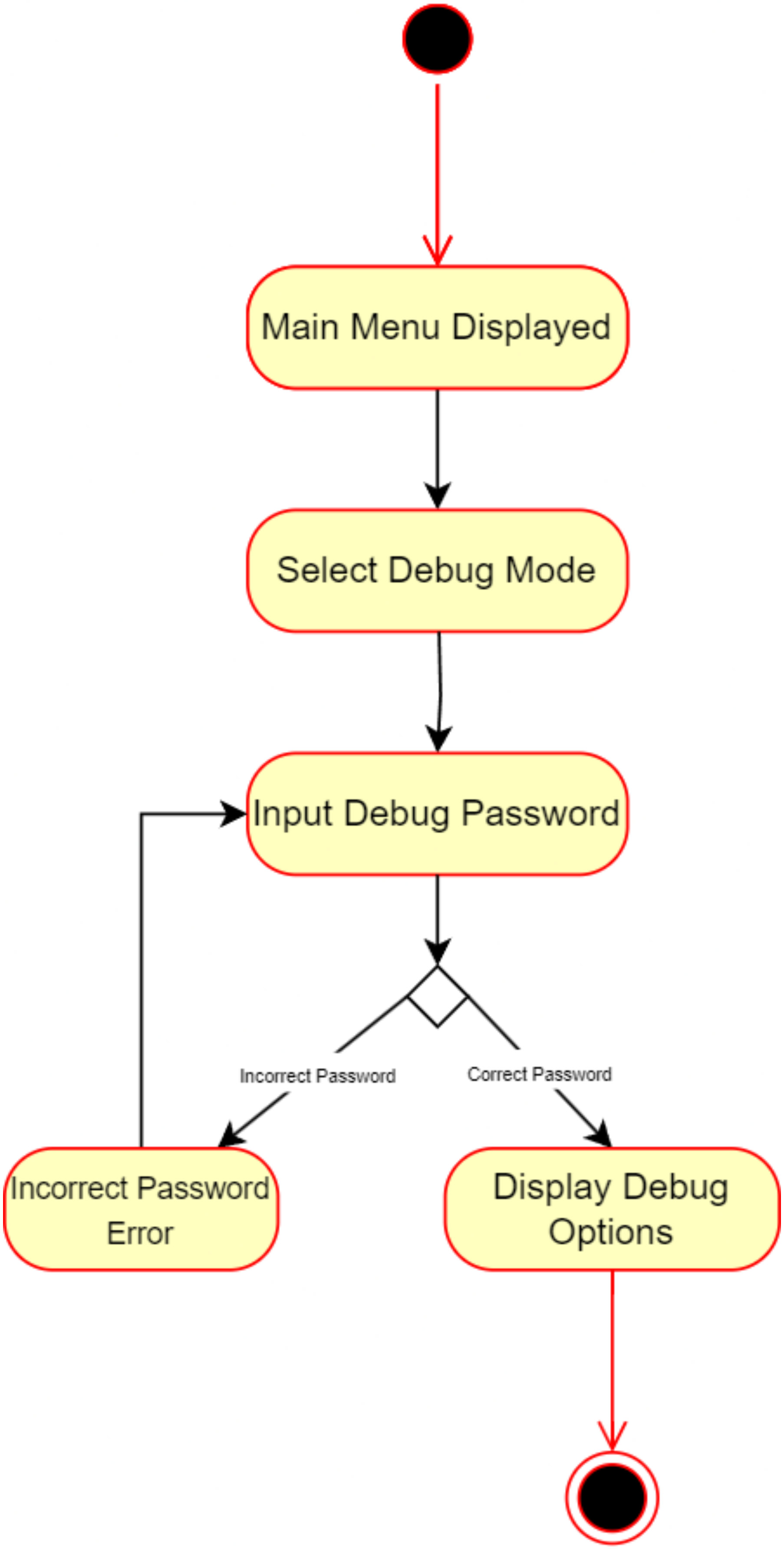
"Access Instructor Dashboard"


Main Menu Displayed
Select Instructor Dashboard
Input Password
Incorrect Password
Correct Password
Incorrect Password error displayed
Display Instructor Dashboard

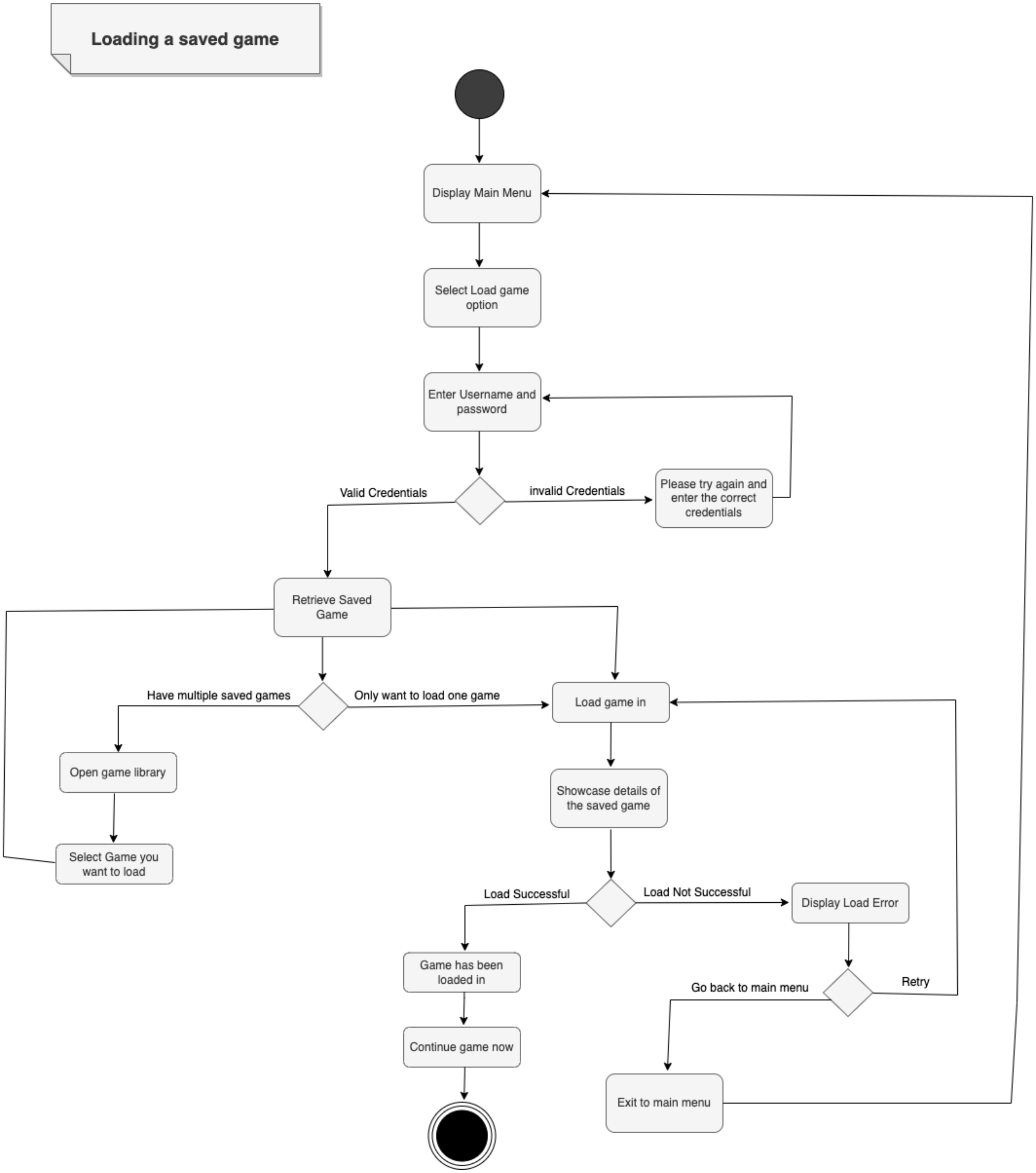
"Submit Question"


Main Menu Displayed
Select Question Creation
Enter Question Details
Submit Question
Invalid Submission
Valid Submission
Display Submission Error

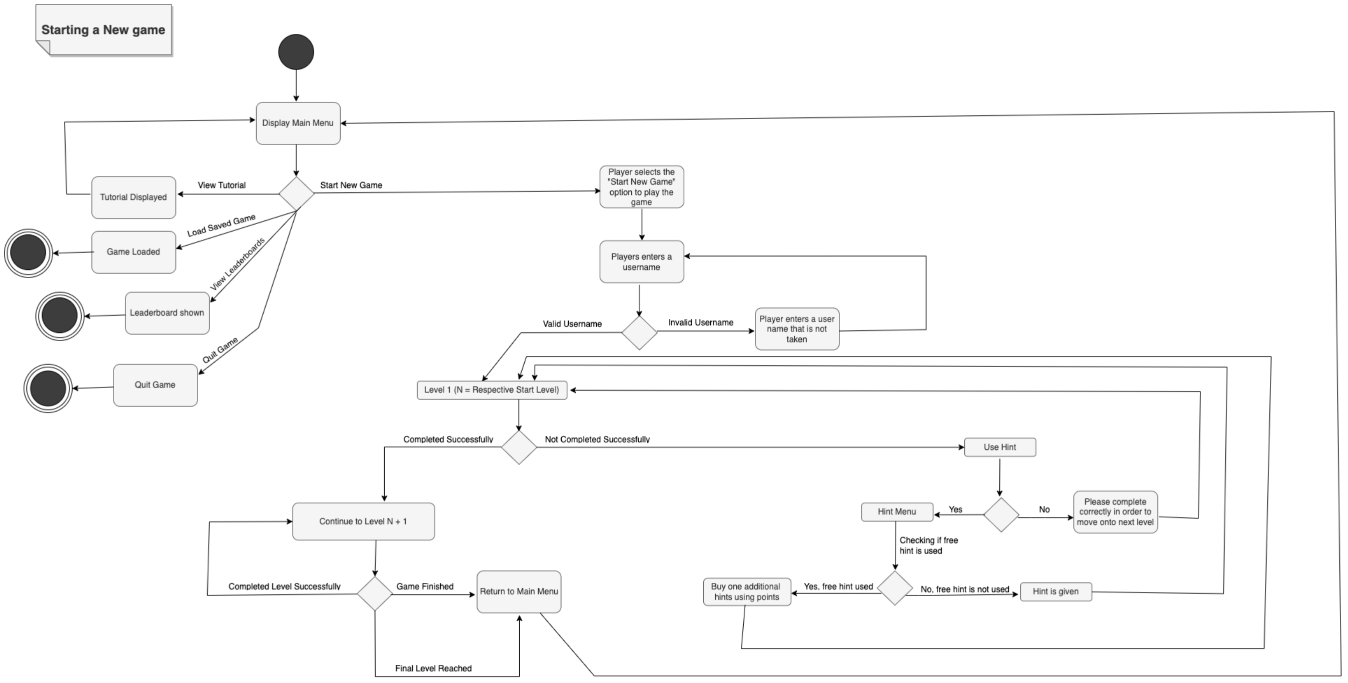
"Access Debug Mode"


Main Menu Displayed
Select Debug Mode
Input Debug Password
Incorrect Password
Correct Password
Incorrect Password Error
Display Debug Options

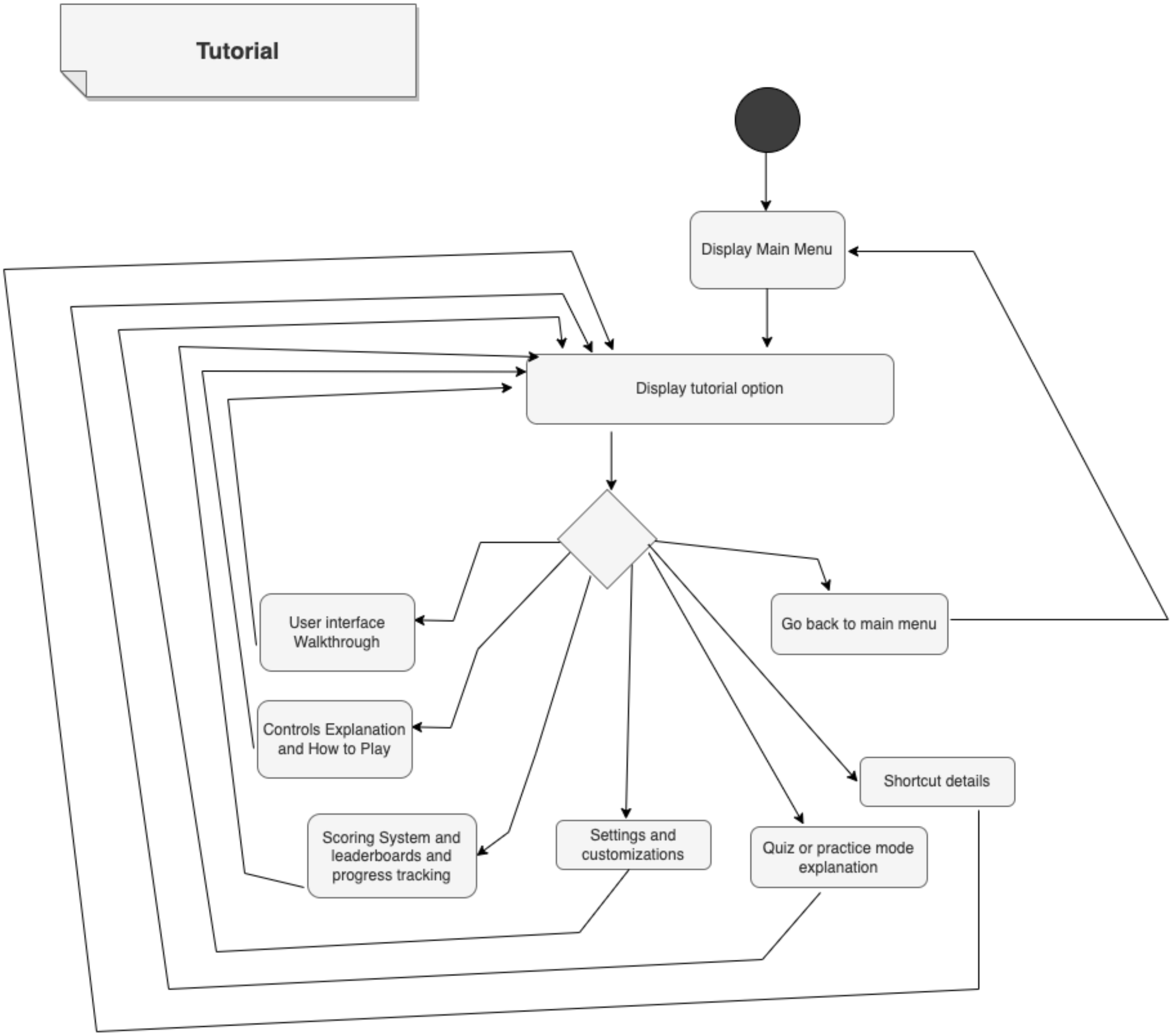
"Loading a saved game"



"Starting a new game"

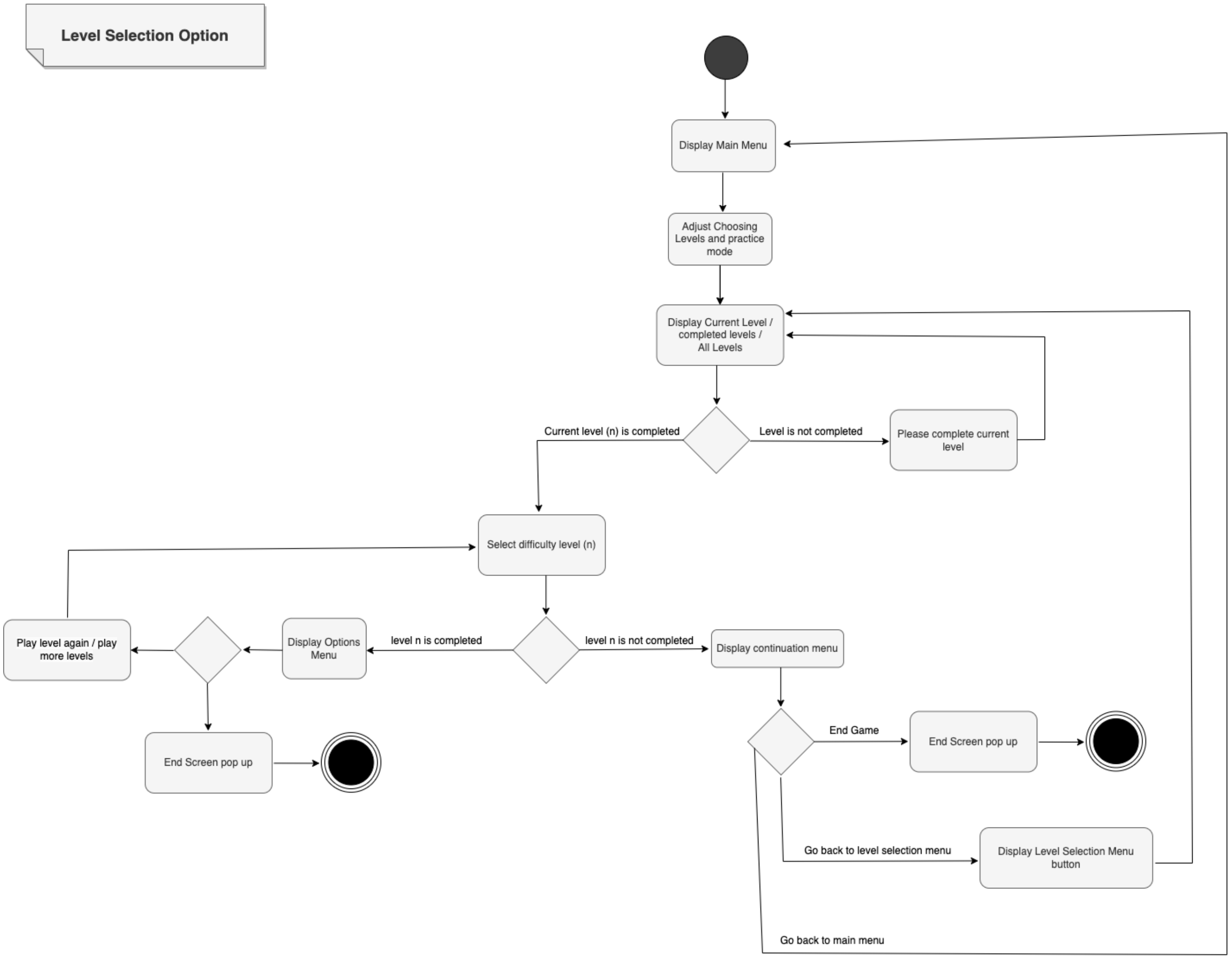

Starting a New game
Display Main Menu
View Tutorial
Start New Game
Tutorial Displayed
Load Saved Game
Game Loaded
View Leaderboards
Leaderboard shown
Quit Game
Quit Game
Player selects the "Start New Game" option to play the game
Players enters a username
Valid Username
Invalid Username
Player enters a user name that is not taken
Level 1 (N = Respective Start Level)
Completed Successfully
Not Completed Successfully
Use Hint
Continue to Level N + 1
Hint Menu
Yes
No
Please complete correctly in order to move onto next level
Checking if free hint is used
Completed Level Successfully
Game Finished
Return to Main Menu
Buy one additional hints using points
Yes, free hint used
No, free hint is not used
Hint is given
Final Level Reached

"Tutorial Menu and option"

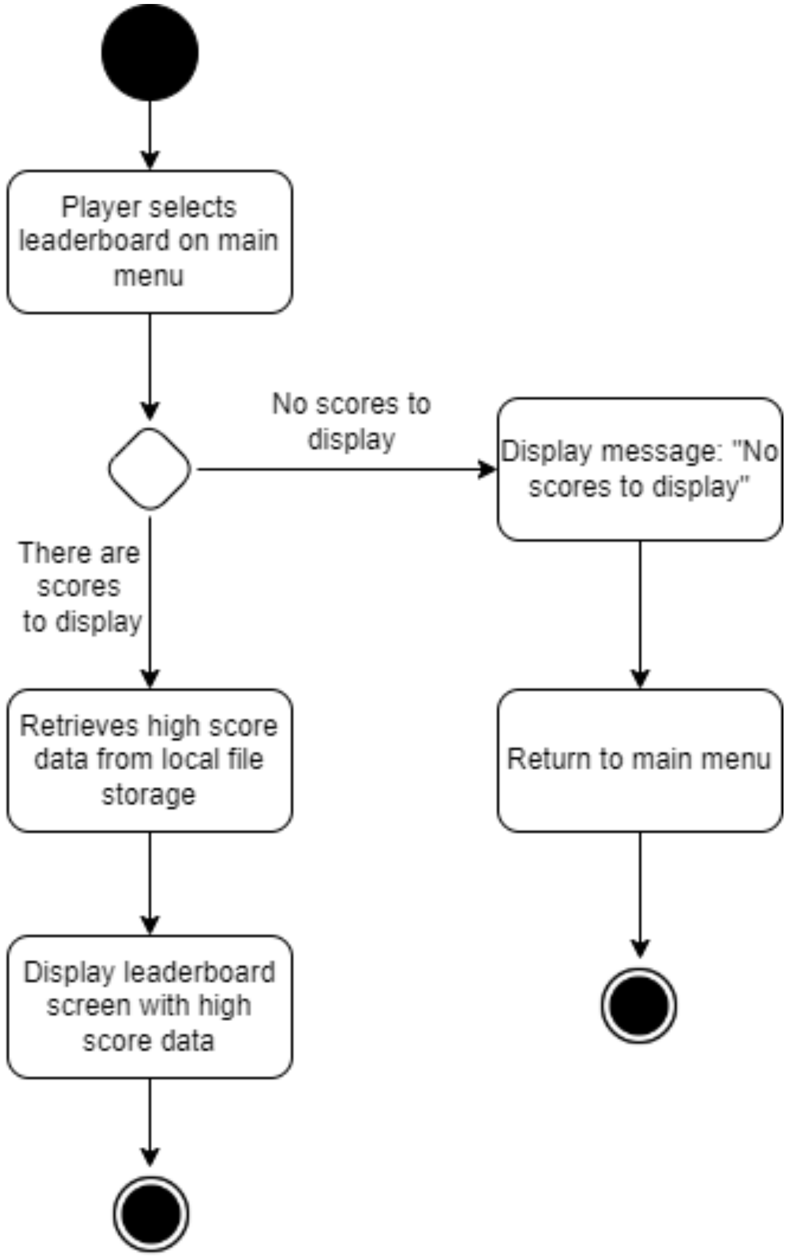

Tutorial
Display Main Menu
Display tutorial option
User interface Walkthrough
Go back to main menu
Controls Explanation and How to Play
Shortcut details
Scoring System and leaderboards and progress tracking
Settings and customizations
Quiz or practice mode explanation

"Level Selection Option"

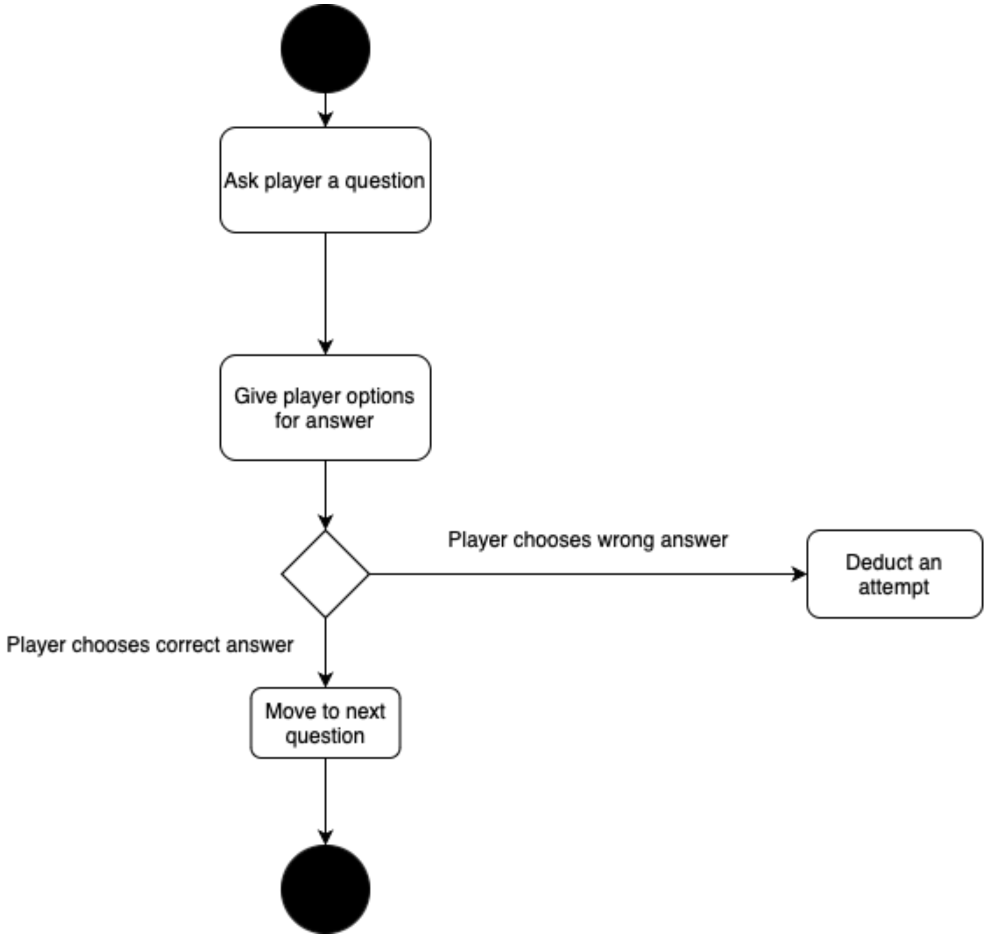
Level Selection Option


Display Main Menu
Adjust Choosing Levels and practice mode
Display Current Level / completed levels / All Levels
Current level (n) is completed
Level is not completed
Please complete current level
Select difficulty level (n)
level n is completed
level n is not completed
Play level again / play more levels
Display Options Menu
Display continuation menu
End Screen pop up
End Game
End Screen pop up
Go back to level selection menu
Display Level Selection Menu button
Go back to main menu

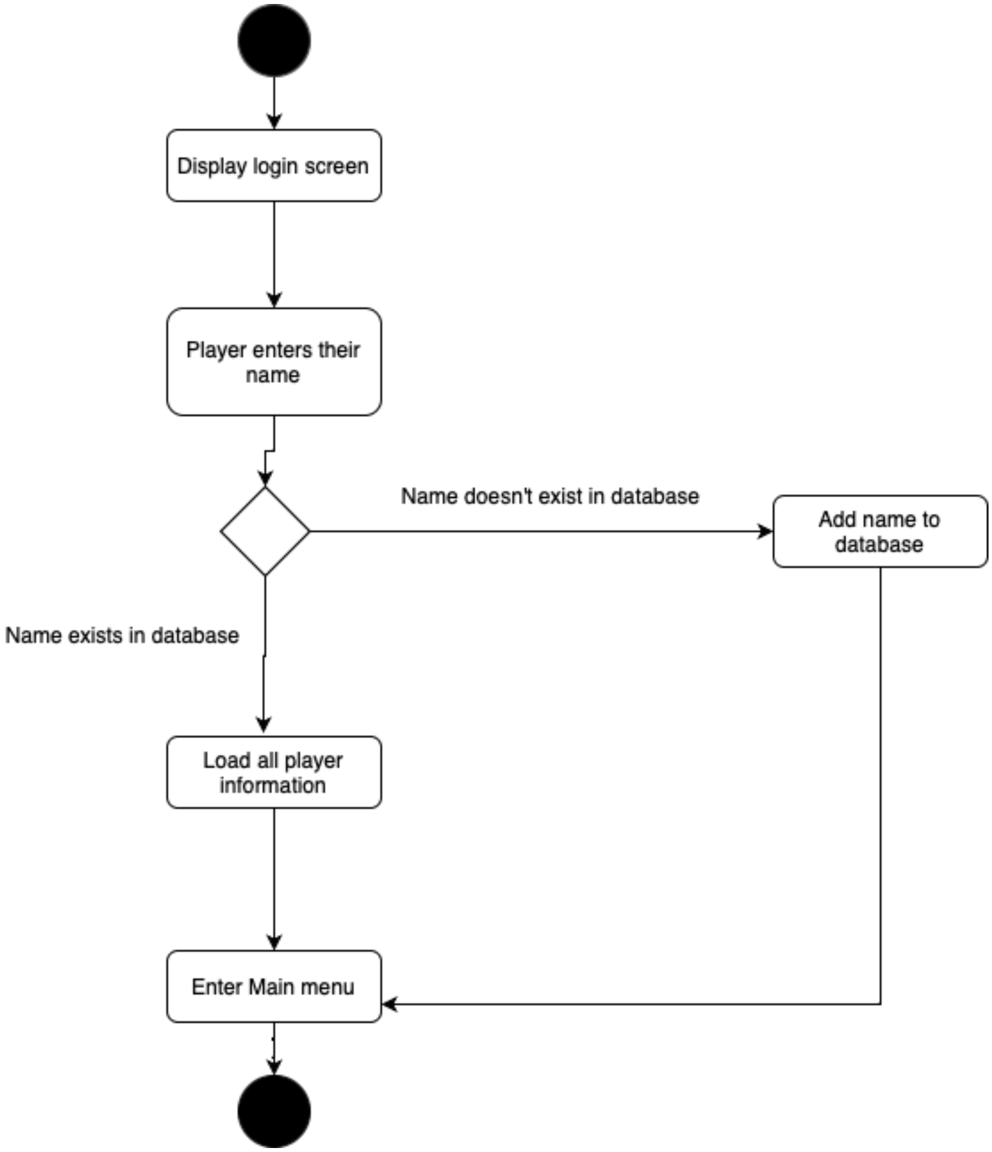
"Accessing the Leaderboard"


Player selects leaderboard on main menu
No scores to display
Display message: "No scores to display"
There are scores to display
Retrieves high score data from local file storage
Return to main menu
Display leaderboard screen with high score data

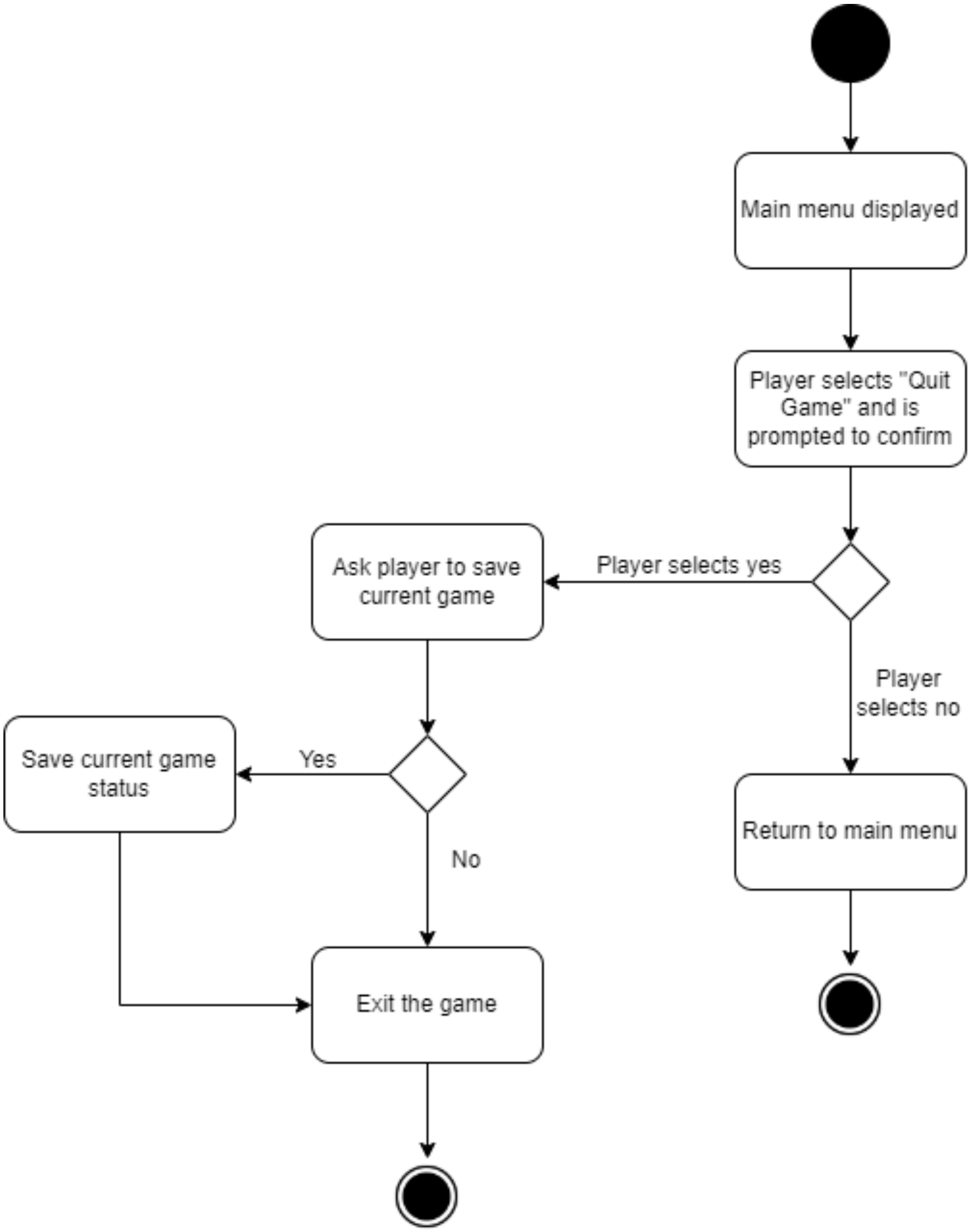
"Answering Questions"


Ask player a question
Give player options for answer
Player chooses wrong answer
Deduct an attempt
Player chooses correct answer
Move to next question

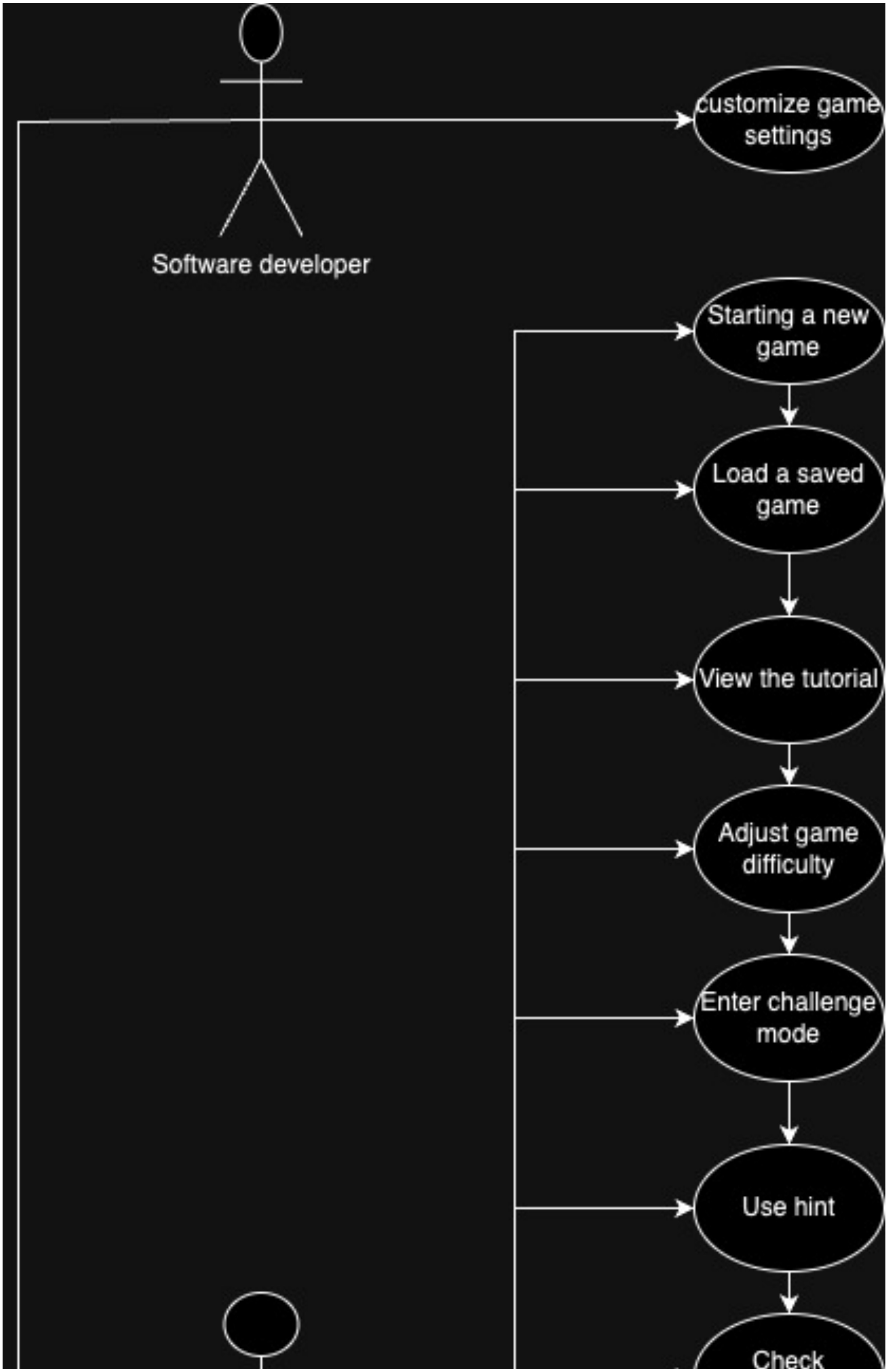
"Signing in as a Player"

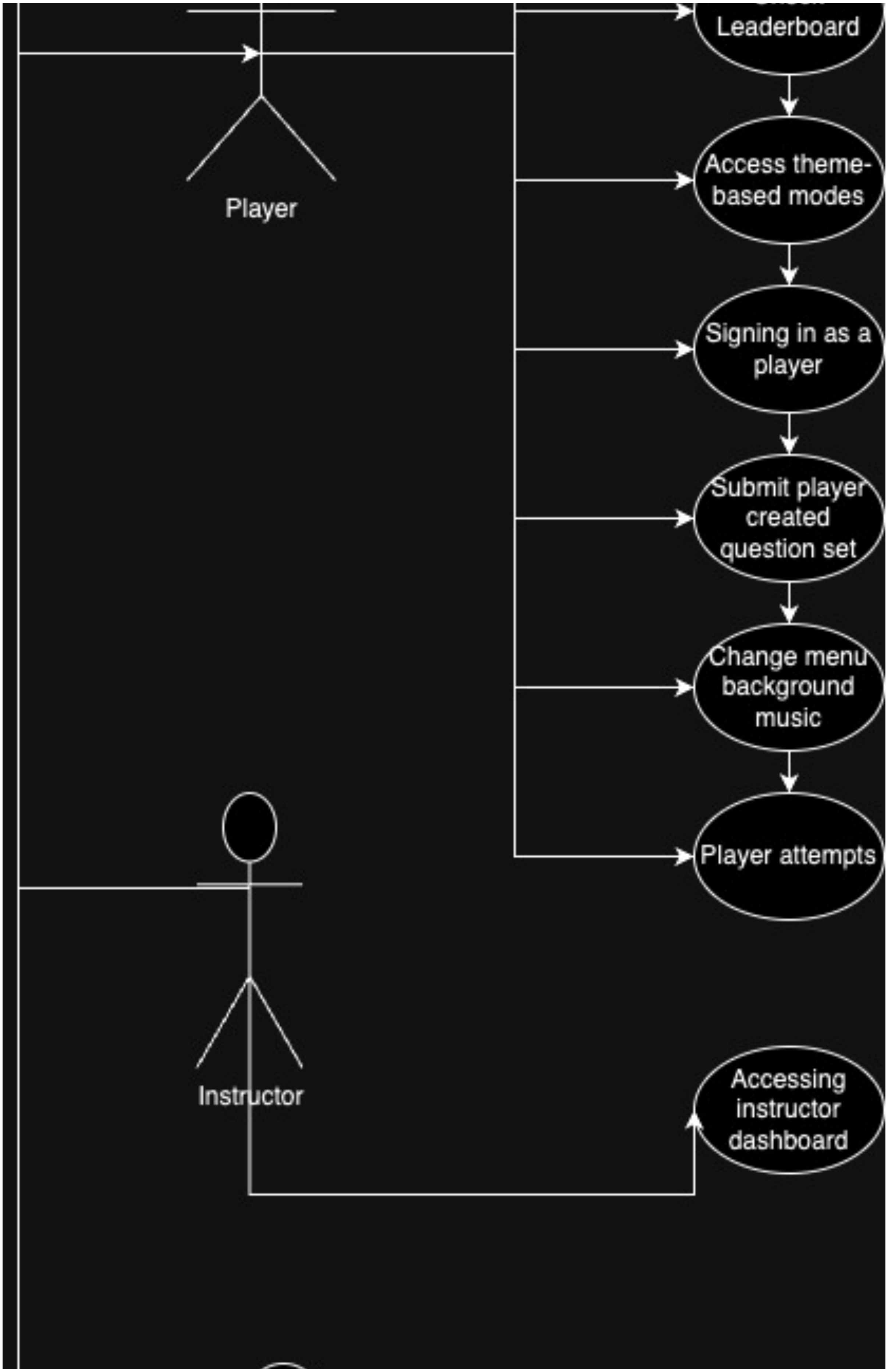

Display login screen
Player enters their name
Name doesn't exist in database
Add name to database
Name exists in database
Load all player information
Enter Main menu

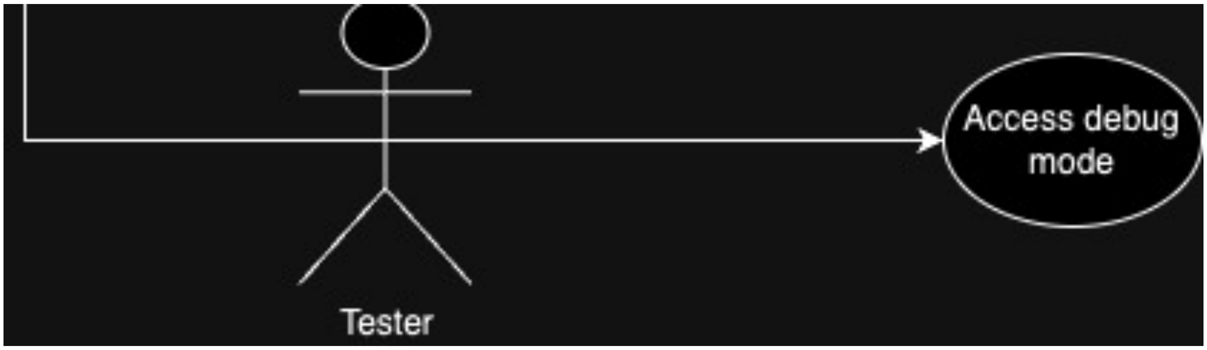
"Quit Game"


Main menu displayed
Player selects "Quit Game" and is prompted to confirm
Player selects yes
Ask player to save current game
Player selects no
Yes
Save current game status
No
Return to main menu
Exit the game

Use Case Diagram


Software developer
customize game settings
Starting a new game
Load a saved game
View the tutorial
Adjust game difficulty
Enter challenge mode
Use hint
Check


Leaderboard
Player
Access theme-based modes
Signing in as a player
Submit player created question set
Change menu background music
Player attempts
Instructor
Accessing instructor dashboard


Access debug mode
Tester

Non-Functional Requirements

The game will adhere to the following non-functional requirements:

1. The game will be developed in Java 19 or newer.
2. The game will be developed using an objected-oriented approach and design patterns.
3. The educational content of the game will be accurate, relevant, up-to-date and appropriate for the target age group. Any instructional material will be derived from credible sources and not contain any inappropriate content. This content will be presented in a manner that is engaging and fun.
4. The game will have a Graphical User Interface, which will be user-friendly and align with best UX practices to ensure intuitiveness and ease of navigation. It will present a cohesive design with clear labeling and a consistent visual language to minimize user effort and facilitate engagement.
5. The game will store all data locally and not require an Internet connection. The data will be stored in a JSON file format.
6. The game will not use and third-party libraries that are not freely available. All tools and build instructions will be included with the application.
7. All code and files for the game will be stored in the Bitbucket Git repository created for the team. Files will be actively committed throughout completion of the project.
8. All design work and diagrams will be stored and developed on Confluence.
9. All tasks and issues related to the game will be tracked on Jira, and be updated as the project progresses.
10. The code will be commented using Javadoc. Each file and each function will have a comment explaining its purposes and author.
11. All code except for some that can only be tests via GUI actions or events will be tested using JUnit5 tests.
12. The team will choose and follow coding conventions and styles that we adopt, and reman consistent in applying these across all files in the application.
13. The game will be executable on a Windows 10 system with a standard Java installation, and each team member will be able to compile it and run it from a development environment they have access to. All team members must use the same development environment and tools as agreed to in the team contract.
14. The game will be well self-contained and not create, modify, or delete files outside of the directory in which the application is installed, and subdirectories of this directory.
15. The game will present a visible response to every user action. Actions that could not succeed for some reason will be met with a useful and professional error message.
16. The file size of the project as a whole will be under 1 gigabyte.
17. The game will run efficiently and not use unnecessary computing resources. The interface will be responsive and not lag while the user is playing the game.
18. The game will take accessibility into account when designing the UI. Tasks will be allowed to completed with both a keyboard and mouse, and UI elements will have a logical tab order. Colour use in the UI will be considered for colour blind players.
19. The code will be maintainable and as reusable as possible, being structures to allow for easy updates and maintenance.
20. All content in the game, documentation, design work, comments, etc. will be written in English and will be clear, comprehensible, or otherwise easily decipherable. All communications between the team members will also be in English.
21. The game will otherwise be designed with sound software engineering principles in mind as discussed in the course.

Summary

The requirements documentation for our educational game project lays out the overall framework and expectations for the development process. Starting with the team contract, we establish clear guidelines for team interaction, meeting protocols, work division, and adaptability. This contract serves as the foundation for assessing team performance and ensuring mutual agreement on responsibilities and workflows. Moving forward, our requirements documentation describes the essence of our project, including objectives, domain analysis, and functional and non-functional requirements. We identify the software domain, target audience, and potential challenges through in-depth analysis, allowing for more informed development decisions. Functional requirements define what the software must accomplish, with a focus on our game's unique mechanics and educational objectives.

With the help of scenario models such as actors, use cases, and activity diagrams, we can clearly envision user interactions and system functionality. Non-functional requirements support functional aspects by addressing the how of our development process. These include factors related to usability, security, and performance, guaranteeing a comprehensive approach to software design and implementation. Our goal is to create an educational game that not only fulfills stakeholder expectations but also offers our users richer experiences by carefully following these precisely defined requirements. We aim for cooperative, effective, and successful project development execution through constant iteration and adherence to our requirements documentation.