Use Cases

Use Case: Play Game

- Actor
  - Player (Dinosaur)
- Goal in context
  - To start and play the game.
- Preconditions
  - Player has to launch the game.
- Scenario: (greyed out part should probably be refactored in the use cases listed below)
  - Player launches the game
  - The main menu screen is displayed that gives them the option to start the game, go to the level selection menu, or exit the game.
    - By selecting "Level Selection", the level selection menu will be displayed where they can pick the level (easy, medium, hard) that they wish to play. Players must have completed easier levels before more difficult levels are unlocked.
    - By selecting "Start Game", the level they selected will be loaded. The player will then be placed into the game.
    - By selecting "Exit", the system will terminate and close the game.
  - The Player maneuvers around the map, using keyboard keys w, a, s or d to move up, left, down and right respectively.
  - While in the game, players can view their health, stamina, amount of keycards and eggs, the amount of time spent in the level,and any current buffs they may have collected from the map.
  - The player has to navigate their character through the maze and collect all keycards to unlock the exit.
  - They can also collect eggs scattered around the maze to increase their score.
  - Animate enemies will actively seek out the player if the player enters the detection zone of the animate enemies.
  - If the player interacts with an enemy, they will lose health and if their health goes under a certain threshold, a "game over" screen is displayed and is given the option to retry or go back to the menu screen.
  - While in the game, the player can also pause the game by pressing the "esc" key. This will bring up an in-game settings menu where they will get an option to continue, go back to the main menu or exit and close the game.
  - The player can only save their progress by completing a level. The information that will be recorded is the amount of time they spent and the number of eggs collected.
- Exceptions
  - Higher Difficulty Levels cannot be played if the previous level has not been completed.
    - The Player must go back and successfully pass the lower level before the system unlocks the next level.

Use Case: Moving Character

- Actor
    - Player (Dinosaur)
- Goal in Context
    - Player wants to move the character around the maze
- Preconditions
    - Player must be loaded into a map
- Scenario
    - User launches the game and selects a level to play.
    - User starts the game.
    - Player needs to move their character (Dinosaur) around the map to collect keys to unlock the door to escape the lab.
    - Player uses keyboard keys w (up), a (left), s (down) or d (right) to move around.
    - Player collects keys, power-ups, and dinosaur eggs while avoiding scientists and spike traps placed around the map.
    - After collecting all the keys Player maneuvers to the unlocked door to win the game.
- Exceptions
    - When a Player is next to a wall and they try to maneuver into it, they cannot walk through them, the walls are solid.
        - The Player must choose a different direction to walk in.
    - If a Player tries to pass through the door before collecting all the keys, they can not as the door will be locked.
        - Player must go back to collect keys around the map, before the system unlocks the door.
    - If a Player walks into a Scientist or fails to out maneuver a scientist
        - The player dies, the game ends, and the player fails the level.
        - The End Game menu will be displayed with player stats.
        - The Player can choose to restart the level, go back to the main menu to choose a different level, or exit the game.
- Priority
    - Essential, must be implemented

Use Case: Auto Save Progress

- Actor
    - Player (Dinosaur)
- Goal in Context
    - Saving the score and progress of the player in order to access further levels.
    - Saves players “high” score
    - Saves that player passed a level so they can unlock the next difficulty level
- Preconditions
    - The Player needs to pass the level in order for their progress in the game to be saved.
- Scenario
    - Player launches the game.
    - Player enters the initial easy level to play.

    - ➢ After the player has passed the level, the game progress will be auto saved.
        - ■ If their current score is higher than their previous attempt, then the new score will be recorded.
    - ➢ System will then unlock the next difficulty level until all levels have been unlocked

- ❖ Exceptions
    - ➢ The game can not be saved while the player is playing a level.
        - ■ The Player can choose to exit the game from the in-game menu, where all current session progress will be lost.
        - ■ The Player can choose to continue playing until they pass the level.
- ❖ Priority
    - ➢ Essential

Use Case: Spawn Enemies

- ❖ Actor
    - ➢ Inanimate: Spikes
    - ➢ Animate: Scientists
- ❖ Goal in Context
    - ➢ Populate map with obstacles that the Player must avoid.
- ❖ Preconditions
    - ➢ Starting a game
- ❖ Scenario
    - ➢ The Player launches the game.
    - ➢ The Player selects a difficulty level to play.
    - ➢ Upon entering the map, the system populates the map with a set amount of enemies.
        - ■ The higher the level of difficulty the more set enemies will spawn.
    - ➢ The Player runs around the map trying to avoid spawned enemies.
- ❖ Exceptions
    - ➢ Only spawned at the beginning of the game.
        - ■ If the player runs into the animate Scientists, the player fails the level and dies and the game ends.
        - ■ If the player runs over spike traps, their life decreases by a set amount but the traps remain in the same location.
    - ➢ Can not spawn on another inanimate object (power-up, spikes, and walls etc.)
        - ■ The location of each item that spawns on the map is pre programmed and remains the same every time you play the same level.
    - ➢ Can not spawn on an animate objects (player, other scientists)
        - ■ The location of each item that spawns on the map is pre programmed and remains the same every time you play the same level.
- ❖ Priority
    - ➢ Essential, must be implemented

Use Case: Enemy Interaction

- ❖ Actor
    - ➢ Scientists
    - ➢ Spikes
- ❖ Goal in Context
    - ➢ The enemy AI purpose is to prevent the player-controlled characters from reaching their destination
- ❖ Preconditions
    - ➢ Enter new game
- ❖ Scenario
    - ➢ The Player launches the game.
    - ➢ The Player selects a difficulty level to play.
    - ➢ Upon entering the map, the system populates the map with a set amount of enemies.
    - ➢ The Player runs around the map trying to avoid spawned enemies.
    - ➢ The system detects and resolves enemy interactions, by using set detection zones, when the player enters the neighbouring cells the scientist is standing or walking in, the scientist becomes alerted.
    - ➢ Scientists: walk back and forth on a set path until player enters their detection zone
        - ■ Once alerted, the scientist will chase and attempt to catch the player
    - ➢ Spikes Traps: have a set position
        - ■ when player walks over the spikes, they receive damage to their health
    - ➢ If the player runs over enough traps that their health drops below zero and the player dies.
    - ➢ If the player runs into or is caught by a scientist, their health instantly drops to below zero and the player dies.
    - ➢ The player loses the game and the game ends.
- ❖ Exceptions
    - ➢ Scientist: if player doesn't enter alert zone, scientist is not triggered to leave their original path
- ❖ Priority
    - ➢ Essential, must be implemented

Use Case: Key Cards

- ❖ Actor:
    - ➢ Player
- ❖ Goal in Context:
    - ➢ Unlocks door to allow player to win the game
- ❖ Preconditions
    - ➢ Game has started
    - ➢ Set amount of keys will spawn on the map in set locations
- ❖ Scenario
    - ➢ The Player launches the game.
    - ➢ The Player selects a difficulty level to play.
    - ➢ Set amount of keys will spawn at the start of game

    - ➢ Number of keys depends on level of difficulty (the more difficult the the level the more keys will spawn)
    - ➢ The Player runs around the map trying to collect all the keys to unlock the door.
    - ➢ Player must run over the key to pick it up.
    - ➢ Player must collect all keys to unlock the door.
    - ➢ Player must maneuver to the unlocked door to escape the lab.
- ❖ Exceptions
    - ➢ Keys can not spawn one on top of other inanimate or animate objects
        - ■ The location of each item that spawns on the map is pre programmed and remains the same every time you play the same level.
    - ➢ Enemies can not collect keys
        - ■ If a scientist walks over a key, the key remains in the same place unchanged
- ❖ Priority
    - ➢ Essential, must be implemented

Use Case: Power-Ups

- ❖ Actor:
    - ➢ Health packs
    - ➢ Stamina Replenishments
    - ➢ Speed potions
    - ➢ Invisibility potions
- ❖ Goal in Context:
    - ➢ Provides player with and edge for winning the game
    - ➢ Health packs: replenishes health points
    - ➢ Stamina Replenishments: replenishes stamina points (if player runs out of stamina they can no longer sprint)
    - ➢ Speed potions: boosts the players run speed for a set time
    - ➢ Invisibility potions: allow the player to become invisible for a short period of time
- ❖ Preconditions
    - ➢ Game started
    - ➢ Set amount of power-up will spawn on the map in set locations
- ❖ Scenario
    - ➢ The Player launches the game.
    - ➢ The Player selects a difficulty level to play.
    - ➢ Game starts.
    - ➢ Power-ups get spawned at set locations.
    - ➢ Player must run over power-up to pick it up and receive benefits
    - ➢ Depending on power-up, player will receive a boost of either health, speed, or stamina
        - ■ Health and stamina boosts are cumulative and are added on to players current health and stamina points.
        - ■ Speed and invisibility potions are time limited and will expire off the player within a set time.
- ❖ Exceptions
    - ➢ Power-up can not be placed one on top of another, or other animate or inanimate objects.

        - The location of each item that spawns on the map is pre programmed and remains the same every time you play the same level.
    - Scientists can not collect power-ups
        - If a scientist walks over a power-up, the power-up remains in the same place unchanged
- Priority
    - Medium priority

Use Case: Dinosaur Eggs

- Actor:
    - Dinosaur Eggs
- Goal in Context:
    - Provides player with and edge for getting a higher score
- Preconditions
    - Game started
    - Set amount of dinosaur eggs will spawn on the map in set locations
- Scenario:
    - The Player launches the game.
    - The Player selects a level to play.
    - The Player begins a game.
    - Power-ups get spawned at set locations.
    - Player must run over power-up to pick it up and receive benefits
    - The Player can run over dinosaur eggs to collect extra points.
    - Each egg picked up increases the score of the player by ten points.
- Exceptions
    - Dinosaur eggs can not spawn one on top of another or other animate or inanimate objects.
        - The location of each item that spawns on the map is pre programmed and remains the same every time you play the same level.
    - Scientists can not collect dinosaur eggs.
        - If a scientist walks over a dinosaur egg, the egg remains in the same place unchanged.
- Priority
    - High priority

Use Case: View Level Stats

- Actor:
    - Player
- Goal in Context:
    - View the highest number of eggs obtained and how long the player took to complete the level.
- Preconditions
    - The player must have completed the level they wish to view.
- Scenario
    - Player starts the game.

- ➢ Clicks on “Play” on the menu screen to view the level select menu.
  - ➢ Then proceed to click the level for which they wish to see the stats of.
  - ➢ Finally by clicking a level, they will see a screen that contains their stats.
- ❖ Exceptions:
  - ➢ Stats of a level cannot be seen if the player has not completed the level
    - ■ If this occurs, the player will be shown a screen that shows they have not completed the level
- ❖ When available
  - ➢ After phase 2

Use Case: Main Menu

- ❖ Actor:
  - ➢ Player
- ❖ Goal in Context:
  - ➢ Allows the player to use and navigate the game.
- ❖ Preconditions
  - ➢ The player must launch the game
- ❖ Scenario
  - ➢ Player launches the game.
  - ➢ The main menu is loaded.
    - ■ Options Start, Level Selection, and Exit buttons are displayed.
  - ➢ If the Player selects “Play” the selected level will load.
  - ➢ If the Player selects “Level Selection” they will be taken to the level selection menu.
  - ➢ Finally if the Player selects “Exit”, the game will terminate.
- ❖ Exceptions:
  - ➢ Main menu can not be viewed while the player is actively playing a level.
    - ■ An alternate in-game pause menu can be viewed by pressing the pause button while playing the game.
- ❖ When available
  - ➢ After phase 2

Use Case: Pause Button and Pause Menu

- ❖ Actor:
  - ➢ Player
- ❖ Goal in Context:
  - ➢ Allows the player to use and navigate the game.
- ❖ Preconditions
  - ➢ The player must have started playing a level
- ❖ Scenario
  - ➢ Player launches the game.
  - ➢ The main menu is loaded.
  - ➢ Selected level is loaded.
  - ➢ The Player can press pause from in-game.
  - ➢ Pause Menu will then pop up.

■ The button Options Main Menu, Continue, or Exit are displayed.

➢ If the Player selects “Main Menu”, the game exits without saving and returns the Player to the main menu.

➢ If the Player selects “Continue”, the game will resume.

➢ Finally if the Player selects “Exit”, the game will terminate without saving.

- Exceptions:
  - Pause menu can not be viewed while the player is not actively playing a level.
    - An alternate menu can be viewed before a game begins (main menu) and just after a game ends (level cleared menu).
- When available
  - After phase 2