Use case: Movement

Primary actor: Main Character (cow)

Goal in context: To allow the movement of the main character in the direction the player wants to (2d-plane) and other animate characters automatically

Preconditions: The map has been set up with a layout and all the entities (animate or inanimate) are positioned randomly (or deterministically) on it. There exists a path between the start and end points of the map

Trigger: The player presses one of the *'up' / 'w'* , *'down' / 's'*, *'right' / 'd'* or *'left' / 'a'* keys (for the main character only)

Scenario: 1. The game starts

                        2. Cow is on starting cell

                        3. Cow can choose to maneuver either up, down, right, left or remain stationary with each "tick" that is passed in game

Exception: 1. Game is paused

                        2. Game over

                        3. When cow is not on board (Before or after game starts)

Priority: Essential, must be implemented

When Available: End of phase 2

Frequency of Use: Frequent

Use case:	Interaction
Primary actor:	All Entities (main character, enemies, rewards, punishment, and barriers)
Goal in context:	To allow for accurate and defined behaviour between all the various entities on the map
Preconditions:	The map has been set up with a layout and all the entities (animate or inanimate) are positioned randomly (or deterministically) on it. There exists a path between the start and end points of the map
Trigger:	The main character and one of the other entities (enemies/rewards) are on the same cell on the map.
Scenario:	1. The game starts
	2. At first the cow moves into a reward cell, the reward is then accounted for and the reward is removed from board
	3. Then the cow moves into a bonus reward cell, the reward is then accounted for in the scoreboard by disturbing additional points and the bonus reward is removed from board
	4. Later the cow moves into a punishment cell, the punishment is then accounted for in the scoreboard by dedication in points and the punishment is removed from board
	5. Then the cow moves into a barrier cell, the barrier cell then prevents the cow from entering the cell, thus resulting in a tick (also known as neutral)
	6. Lastly the cow moves into a enemy cell, the enemy is then accounted for and the game outputs game over
Exception:	1. Cannot collect bonus reward if cow takes too long
	2. Cow with negative point due to punishment will result in game over
	3. Cow will not be able to interact with any entity if game is over
Priority:	Essential, must be implemented
When Available:	End of phase 2
Frequency of Use:	Frequent

Use case:	Start New Game
Primary actor:	Player
Goal in context:	To set up the map and position all entities randomly (or deterministically) across the map. There should be atleast one path from the start point to the end point
Preconditions:	The game has been freshly loaded.
Trigger:	The player clicks on the 'Play' button on the main.
Scenario:	1. Player is on menu screen
	2. Then the player clicks 'Play' button on the screen
	3. After the map and position of all entities are randomly generated across the map, and there should be atleast one path from start to the end.
Exception:	N/A
Priority:	Important, must be implemented
When Available:	End of phase 2
Frequency of Use:	Frequent

Use case:	Choose Difficulty
Primary actor:	Player
Goal in context:	To allow the player between three levels of difficulty, namely beginner, intermediate and professional
Preconditions:	The game has been freshly loaded or the player returns to the main menu after exiting an already started game without completion or the player has successfully completed the game
Trigger:	The player clicks on one of the three available difficulty buttons on the main or success screen
Scenario:	1. The player is on the main or success screen
	2. Then the player clicks one of the 3 available difficulty
	3. After the map and position of all entities are randomly generated across the map depending on the difficulty the map and entities will be more difficult (more punishment, barrier, enemies)
Exception:	N/A
Priority:	Important, must be implemented
When Available:	End of phase 2
Frequency of Use:	Frequent

Use case:	Game End
Primary actor:	Player
Goal in context:	To show the game end screen when the player either successfully reaches the end point, hence finishing the game or the player is captured by an enemy which leads to them losing the game. The game end screen shows the player's score for this match, a return to main menu button, and a high score table.
Preconditions:	The player was already in a newly started or restarted game.
Trigger:	Either the player reaches the end point after managing to complete the minimum number of required objectives or the player is captured by an enemy or the player's score falls below a certain threshold.
Scenario1:	1. Player starts game
	2. Player collects all regular rewards
	3. Player reaches the end cell
	4. Success screen pops up
Scenario2:	1. Player starts game
	2. Player collects all regular rewards
	3. Player moves into enemy cell
	4. Game over screen pops up
Exception:	1. Player will not be able to reach successful screen if game is over
	2. Player will not be able to reach game over screen if player beat game
Priority:	Important, must be implemented
When Available:	End of phase 2
Frequency of Use:	Frequent

Use case:	Pause Game
Primary actor:	Player
Goal in context:	To show the pause screen when the player pauses an already ongoing game. The pause screen shows a continue current game button, and a return to main menu button (exit game).
Preconditions:	The player was already in a newly started or restarted game.
Trigger:	The player presses the 'Esc' key in an ongoing game.
Scenario:	1. Player is in game
	2. Player presses Esc key
	3. Player is able to click on the following buttons, continue, and return to the menu.
Exception:	N/A
Priority:	Moderate priority, should be implemented
When Available:	End of phase 2
Frequency of Use:	Infrequent