Use Case (A): Zombie Dash Game

Description: Move, collect rewards, and encounter enemies in the 2D arcade-style game.

Iteration:	Phase 1 – October 22, 2021 by Group 6	**Prority:**	Moderate
Primary Actor:	Human Player	**When Available:**	Phase 4
Goal in Context:	For the player to collect all the regular rewards and reach the end cell to win the game as the main character.	**Frequency of Use:**	Dependent on the player's desire to play the game.
Trigger:	The player begins the game by clicking the "Start" menu.	**Channel to Actor:**	Via desktop application.
Pre Conditions:	The main character, enemies, and environment (such as walls and obstacles) of the game, as well as the appropriate menus, must be fully configured for the player to begin the game.	**Open Issues:**	1. What minimum system requirements are recommended to play this game?

Scenario:	**Exceptions**:
1. The player presses the "Start Button" on the game's start menu interface. 2. The system's game environment and scorekeeping windows are loaded in the game's interface and the main character starts at its initial location (first cell). 3. The player controls the main character's movements and actions through button presses on their keyboard. 4. The player moves the main character up/down/left/right to an adjacent cell. If there are no obstacles on the target cell, the main character can walk to a neighbouring cell. At each "tick" of the game, the player can only move one cell. 5. Periodically, bonus rewards spawn randomly throughout the game, but only last for several ticks before disappearing. 6. The main character further progresses by picking up a regular reward by successfully avoiding enemies, and receives additional points to their score if a bonus reward is collected. 7. After the player completes their action, the moving enemy will also move one cell. 8. The player wins by making the main character collect all the regular rewards. 9. The system shows the player's total score on the user interface.	1. The player encounters an enemy and receives a punishment - see use case **Punishment**. 2. The player encounters an enemy, receives punishment, and loses the game due to negative score— see use case **Player Loses**. 3. The player does not hit a valid movement key (WASD) during their turn—see use case **Validate User Movement Input**. 4. The player moves to an invalid cell (i.e. barrier)—see use case **Validate User Movement Input**. 5. The game crashes unexpectedly and displays an error message— see use case **Game Crashes**.

Use Case (B): Punishment

Description: Case where the player encounters an enemy.

Iteration:	Phase 1 – October 22, 2021 by Group 6	**Prority:**	Moderate
Primary Actor:	Human Player	**When Available:**	Phase 4
Goal in Context:	For the player to avoid damage from moving enemies and punishments that would cause a negative score.	**Frequency of Use:**	In-game dependent.
Trigger:	Player encounters an enemy.	**Channel to Actor:**	Via desktop application.
Pre Conditions:	While the game is running. See **Use Case (A) – Zombie Dash Game**	**Open Issues:**	1. How many moving enemies and punishments should be placed on each map of the game? 2. How much damage will the player incur? 3. Will damage differ between different enemies?

Scenario:

1. Enemy successfully catches the player's position and attacks
2. System indicates the player received damage
3. Player's score is reduced by the amount of damage received.
4. System records the player's updated score.
5. Player continues the game from Use Case (A) Scenario #4 and on.

Use Case (C): Player Loses

Description: Case where the player encounters an enemy and loses as a result of negative damage.

Iteration:	Phase 1 – October 22, 2021 by Group 6	**Prority:**	Moderate
Primary Actor:	Human Player	**When Available:**	Phase 4
Goal in Context:	For the player to restart the game and attempt the map again.	**Frequency of Use:**	In-game dependent.
Trigger:	System detects if the main character total score drops below 0 from punishments.	**Channel to Actor:**	Via desktop application.
Pre Conditions:	While the game is running. See **Use Case (A) – Zombie Dash Game**	**Open Issues:**	1. How many tries will be given to the player after losing 'x' amount of times? 2. Will the user have to restart the entire progress or will they be able to continue where they had left off before losing?

Scenario:

1. Player encounters enemy
2. System indicates the player received damage
3. Player's score is reduced by the amount of damage received and is negative.
4. System displays the message "Game Over".
5. System records the player's updated score.
6. Player is presented with a screen containing two buttons connecting to the following options: "Restart the game" or "Exit the game".

Use Case (D): User Movement Input

Description: The system reads the input from the player and validates it.

Iteration:	Phase 1 – October 22, 2021 by Group 6	**Prority:**	Moderate
Primary Actor:	System	**When Available:**	Unknown
Goal in Context:	For the system to read if the player had given a valid input to act in the game.	**Frequency of Use:**	In-game dependent.
Trigger:	System detects invalid input from the player during their turn.	**Channel to Actor:**	Via desktop application.
Pre Conditions:	While the game is running. See **Use Case (A) – Zombie Dash Game**	**Open Issues:**	
Scenario:			

1. System displays an error message in the game, “Invalid movement by player.”
2. System skips the player’s turn and allows the enemy to move.

Use Case (E): Game Crashes

Description: A failure scenario where some aspect of the game has been misconfigured.

Iteration:	Phase 1 – October 22, 2021 by Group 6	**Prority:**	Moderate
Primary Actor:	System	**When Available:**	Phase 4
Goal in Context:	For the system to detect what type of error occurred and notify the user.	**Frequency of Use:**	In-game dependent.
Trigger:	System detects invalid input from the player during their turn.	**Channel to Actor:**	Via desktop application.
Pre Conditions:	The game was in the process of running.	**Open Issues:**	1. How will these crashes/bugs/errors be handeled by the system? 2. Will the system be programmed to automatically rerun the application? 3. Will the system be able to save the player’s progress before the crash occurred?
Scenario:			

1. Desktop displays the appropriate error message notifying the player that the game has crashed.
2. System detects the crash and prompts if the player would like to rerun the game.