Team: Kevin Flynn
Andrew Callahan
Hayden Reich
Jonathan Vu

Title: Side Scrolling Game

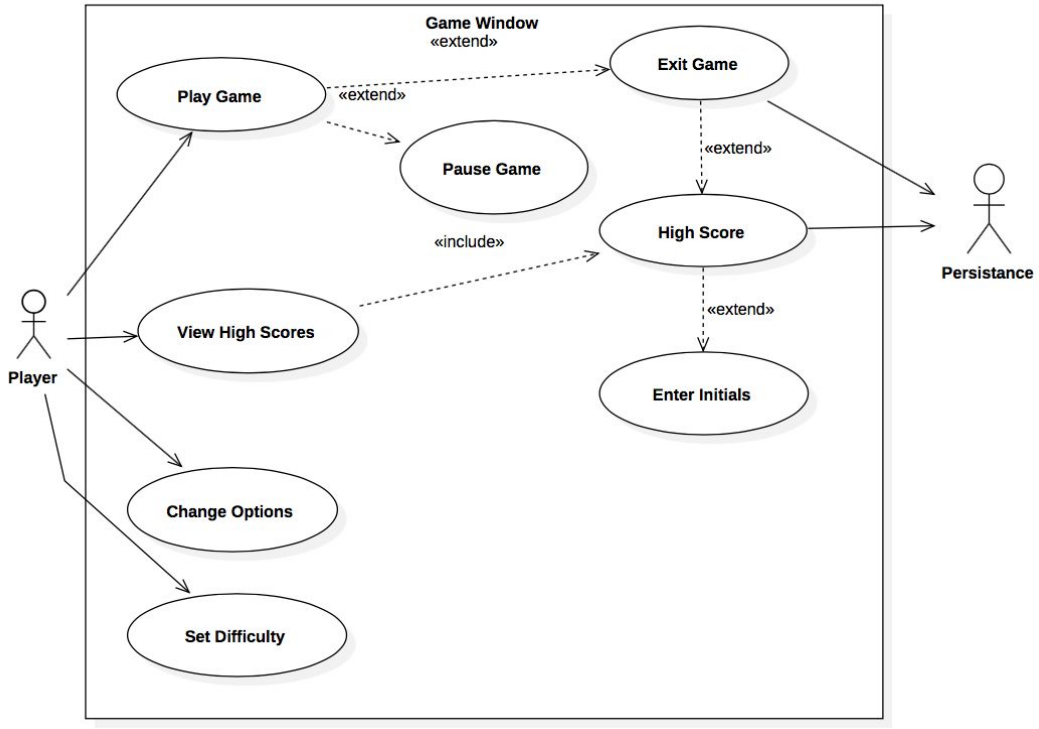
Summary: This will be a side scrolling style game in which the player is a developer doing battle with typical computer adversaries such as bugs, memory leaks and hackers. The developer can gain new attacks and abilities through caffeine power ups found throughout the level. The player will be able to attack enemies that move around the level and attempt to attack and kill the player. Points are awarded for damaging and killing enemies, and high scores will be tracked between game sessions. The player will also be able to save his or her game and load the previous play session rather than starting a new game.

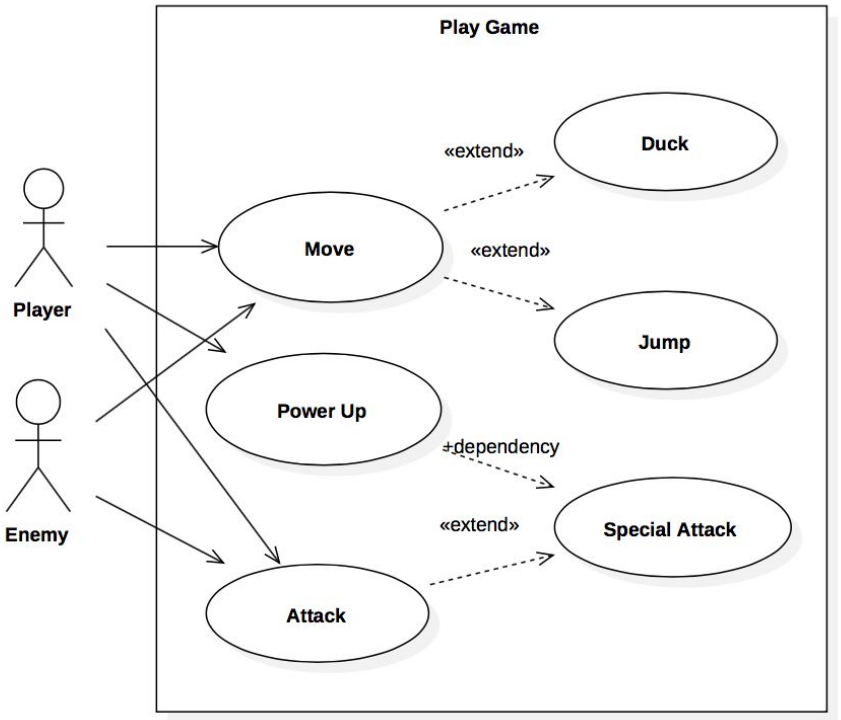
Requirements:

User Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
UR-001	The game state should be able to be saved under current account	Gameplay	Player	Medium
UR-002	High scores should be saved with associated initials	Persistence	Player	Low
UR-003	The user should be able to manuever the character with arrow keys	Game play	Player	Critical
UR-004	The user should be able to change the difficulty of the game	Settings	Player	Medium
UR-005	The user should be able to pause the game	Gameplay	Player	Medium
UR-006	The admin should have access to each account created, as well as being able to manage all high scores	Settings	Admin	Medium
UR-007	The user should be able to pick up power ups	Gameplay	Player	High

UR-008	The user should be able to attack enemies	Gameplay	Player	Critical
UR-009	The user should be able to start the game from the menu	Gameplay	Player	Critical
UR-010	Create Account	Gameplay	Player	High
Functional Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
FR-001	The difficulty should have variable levels	Settings	Player	Medium
FR-002	The player should be able to move around level	Game play	Player	Critical
FR-003	The enemy's health should decrease when in attacked by player	Game play	System	Critical
FR-004	The enemy should disappear when defeated	Gameplay	System	Critical
FR-005	The enemy will be able to attack the player	Gameplay	Enemy	Critical
FR-006	The enemy will be able to move	Gameplay	Enemy	High
FR-007	The player should die and game ends when health is depleted	Gameplay	System	Critical
Non-functional Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
NR-001	The game window should be moveable	Settings	Player	Low
NR-002	The system should interact with mysql when transferring account data	Interface	Admin	Medium
NR-003	The game should run smoothly at every game state	Gameplay	Player	High

Use Cases:


Game Window
«extend»
Play Game
«extend»
Exit Game
Pause Game
«extend»
High Score
«include»
Persistance
Player
View High Scores
«extend»
Enter Initials
Change Options
Set Difficulty


Play Game
«extend»
Duck
Player
Move
«extend»
Jump
Power Up
+dependency
Enemy
«extend»
Special Attack
Attack

Use Case ID:	UC-001	
Use Case Name:	Save Game	
Description:	When the player pauses the game and presses "Save Game" the game state should be saved under their account name	
Actors	Player	
Pre-conditions	The player wants to save the game	
Post-conditions	The game state is saved	
Frequency of Use:	Every time the player plays the game	
Flow of events:	Actor Actions:	System Response:

	1. The player pauses the game	The game pauses.
	2. The player presses "Save Game" button	The game state is saved under that account name.
Variations:	None	
Exceptions:	None	
Developer Notes:	None	

Use Case ID:	UC-002	
Use Case Name:	Saving High Scores	
Description:	When a players game ends, a high score should be saved and associated to the account that achieved said high score	
Actors	Player	
Pre-conditions	The game must first end for this case to be active	
Post-conditions	A new high score is saved to the players account and database	
Frequency of Use:	Once per game	
Flow of events:	Actor Actions	System Response
	1. The player dies in game	Game Ends
		High Score is saved to database
Variations:	N/A	
Exceptions:	Score is not higher than previous high score	
Notes and Issues:	Each account should only have one high score	

Use Case ID:	UC-003	
Use Case Name:	Character Movement	
Description:	When the user presses on any of the four arrow keys: up, down, left or right, then the player should move in that direction	

Actors	**Player**	
Pre-conditions	**The play presses an arrow key**	
Post-conditions	**The player moves in chosen direction**	
Frequency of Use:	**Every time the user plays the game**	
Flow of events:	**Actor Actions:**	**System Response:**
	1. The user presses an arrow key	**The player is moved in the direction that matches the arrow key that was pressed.**
	2. When the user releases the arrow key	**The player stops moving in that direction**
Variations:	**None**	
Exceptions:	**If the player is trying to move off the screen, then the player won't move in that direction**	
Developer Notes:	**None**	

Use Case ID:	**UC-004**	
Use Case Name:	**Changing Difficulty**	
Description:	**The player should be able to access a menu where he can change the difficulty of the enemies**	
Actors	**Player**	
Pre-conditions	**The player accessess the start menu**	
Post-conditions	**The difficulty is changed to whatever the player selected**	
Frequency of Use:	**Possibly every game**	
Flow of events:	**Actor Actions**	**System Response**
	1. User accessess the start menu	**Start menu is launched**
	2. User changes the difficulty	**Difficulty is changed**
		Each enemy is updated with new difficulty
Variations:	**The user could select multiple difficulties**	
Exceptions:	**N/A**	
Developer Notes:	**Will decide if the game will restart when difficulty is changed**	

Use Case ID:	**UC-005**	
Use Case Name:	**Pause Game**	
Description:	**The game state is paused and is able to exit to main menu, save game, or resume game**	
Actors	**Player**	
Pre-conditions	**N/A**	
Post-conditions	**The game state is paused**	
Frequency of Use:	**Often**	
Flow of events:	**Actor Actions:**	**System Response:**
	1. The user presses the "Esc" key	**The game pauses and a pause menu appears**
Variations:	**None**	
Exceptions:	**None**	
Developer Notes:	**None**	
Use Case ID:	**UC-006**	
Use Case Name:	**Admin Access**	
Description:	**The admin should have access to account settings, as well as being able to update and delete any high scores**	
Actors	**Admin**	
Pre-conditions	**There must be accounts and high scores the admin can access**	
Post-conditions	**Any changes the admin makes should take effect.**	
Frequency of Use:	**Very Rarely**	
Flow of events:	**Actor Actions**	**System Response**
	1. Admin accesses accounts/high scores	**All accounts/high scores are shown**
	2. Admin makes any changes necessary	**Account/High Scores are updated**
Variations:	**N/A**	
Exceptions:	**If no accounts or high scores are present, then the admin will have nothing to access**	

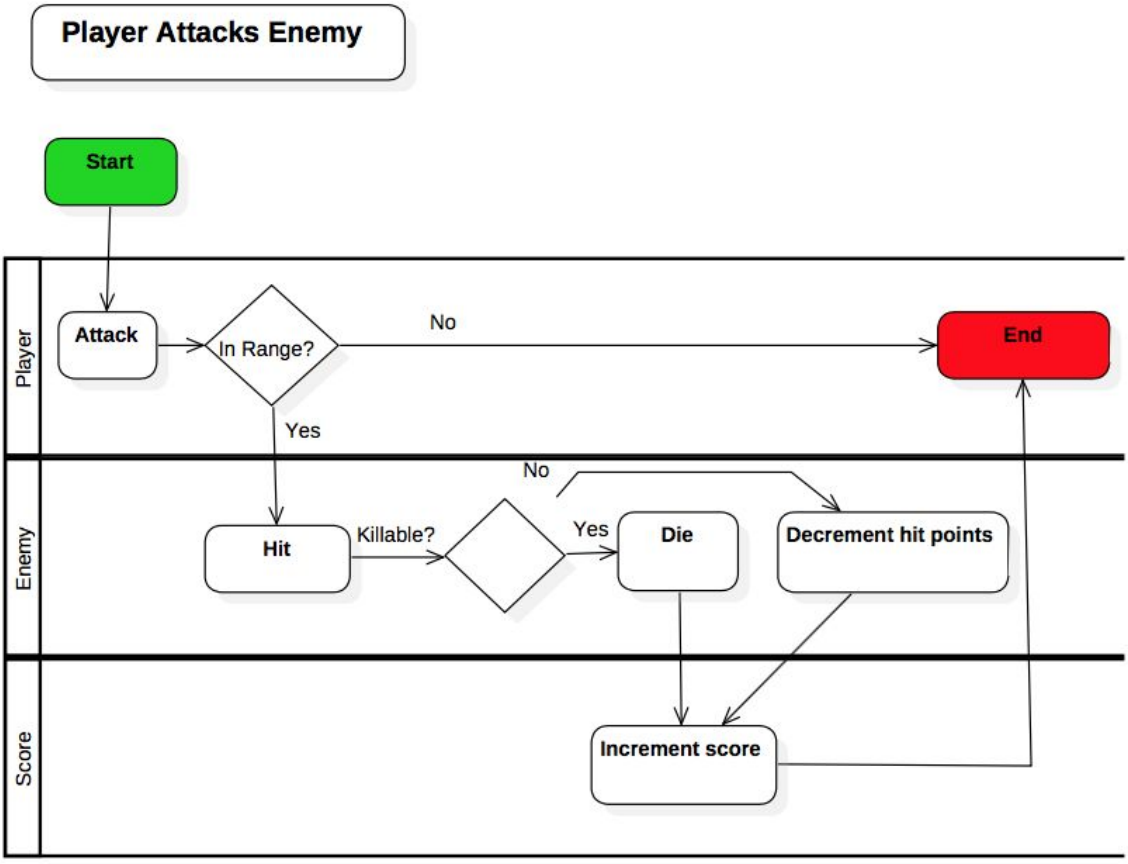
Developer Notes:	**Will be parts that the admin can and cannot access**	
Use Case ID:	**UC-007**	
Use Case Name:	**Pick up power up**	
Description:	**The player picks up a power up and gains special powers**	
Actors	**Player**	
Pre-conditions	**There is a power up to be picked up**	
Post-conditions	**The power up disappears after being picked up**	
Frequency of Use:	**Every game**	
Flow of events:	**Actor Actions:**	**System Response:**
	1. The player moves over a power up	**The player gains special abilities for a specific amount of time**
Variations:	**There are different power ups to be picked up**	
Exceptions:	**The player can only have one power up at a time**	
Developer Notes:	**None**	
Use Case ID:	**UC-008**	
Use Case Name:	**Attack Enemies**	
Description:	**The player should be able to attack any enemy he encounters**	
Actors	**Player**	
Pre-conditions	**An enemy must be present**	
Post-conditions	**The enemy's health will decrease by the amount of damage taken**	
Frequency of Use:	**Every game**	
Flow of events:	**Actor Actions**	**System Response**
	1. Player attacks an enemy	**Register hit**
		Decrease enemy health by amt of dmg

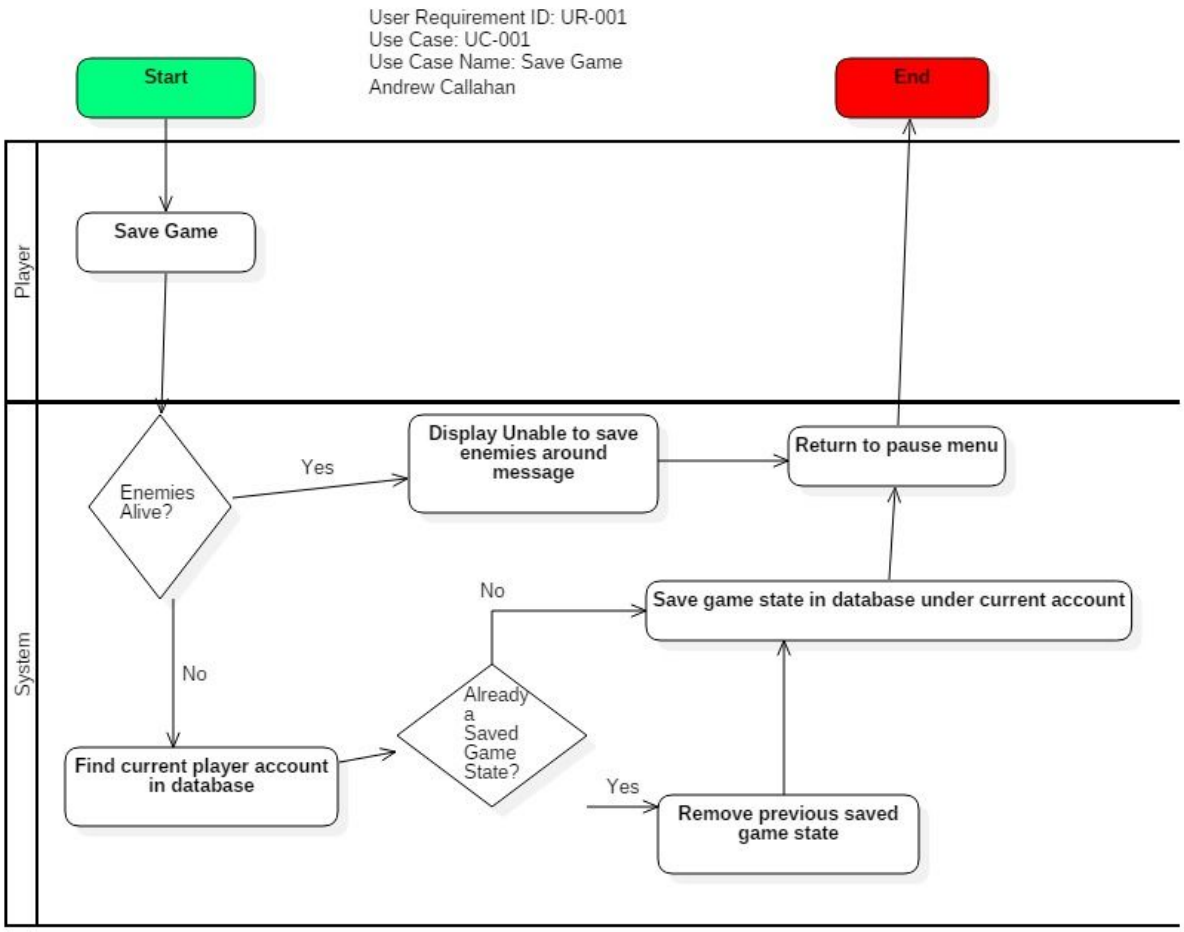
Variations:	**Possibly multiple enemies who take different amounts of dmg**	
Exceptions:	**If no enemies are present the user should not be able to attack**	
Developer Notes:	**Will decide the type of player attacks in the future**	
Use Case ID:	**UC-009**	
Use Case Name:	**Start Game**	
Description:	**The game should start when the "Start Game" Button is pressed**	
Actors	**Player**	
Pre-conditions	**The player is at the main menu**	
Post-conditions	**The game is started**	
Frequency of Use:	**Every time the player wants to play the game**	
Flow of events:	**Actor Actions:**	**System Response:**
	1. The player clicks on the "Start Game" Button	**The system goes to the game and exits the main menu**
Variations:	**None**	
Exceptions:	**None**	
Developer Notes:	**None**	
Use Case ID:	**UC-010**	
Use Case Name:	**Take Damage**	
Description:	**The enemy should attack player when within a certain distance**	
Actors	**Player**	
Pre-conditions	**Player within range of enemy**	
Post-conditions	**Players health will decrease, or player will lose power up if applied**	
Frequency of Use:	**Every game**	
Flow of events:	**Actor Actions**	**System Response**
	1. Player attacked by enemy	**Register hit**

	2. Player health decreases	**Decrease Player health by damage dealt**
Variations:	**Enemies with different types of attacks/damage values**	
Exceptions:	**None**	
Developer Notes:	**Attack types will be decided later, and if they will be different than player's attack types**	

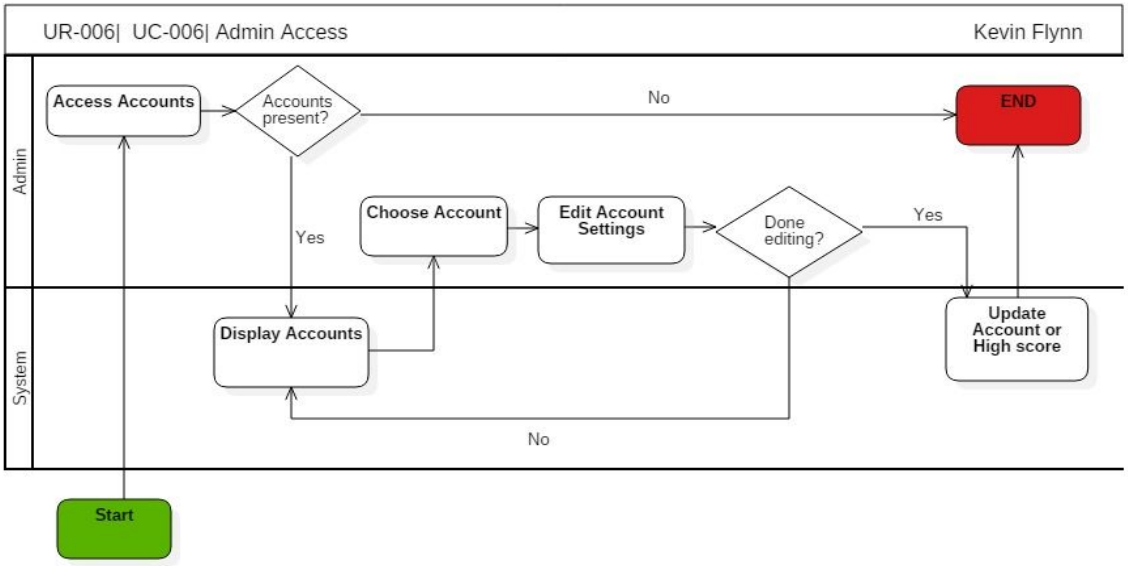
Activity Diagrams:

Hayden Reich
Use Case ID #: UC-008
Requirement ID #: UR-008
Player Attacks Enemy

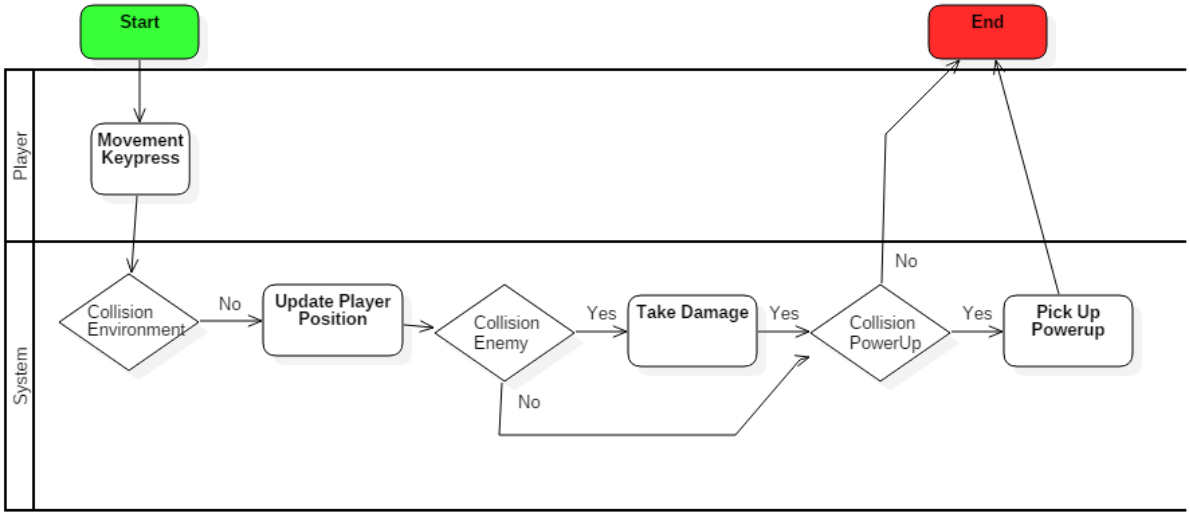



User Requirement ID: UR-001
Use Case: UC-001
Use Case Name: Save Game
Andrew Callahan
Start
End
Player
Save Game
System
Enemies Alive?
Yes
Display Unable to save enemies around message
Return to pause menu
No
Save game state in database under current account
No
Already a Saved Game State?
Find current player account in database
Yes
Remove previous saved game state

Admin Account Access



User Requirement ID: UR-003
Use Case: UC-003
Use Case Name: Character Movement
Jonathan Vu

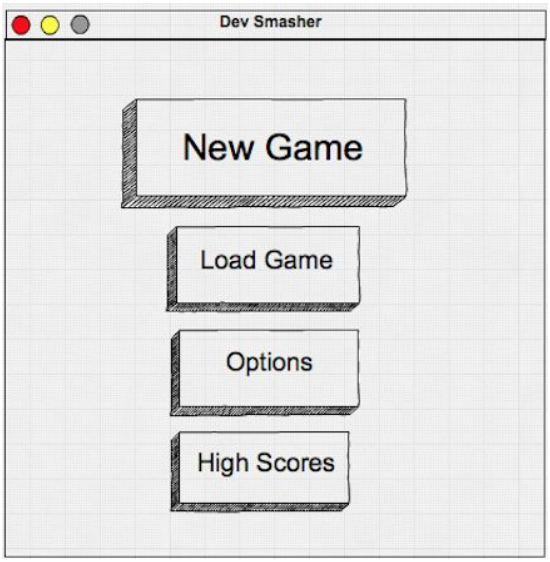


Data Storage:

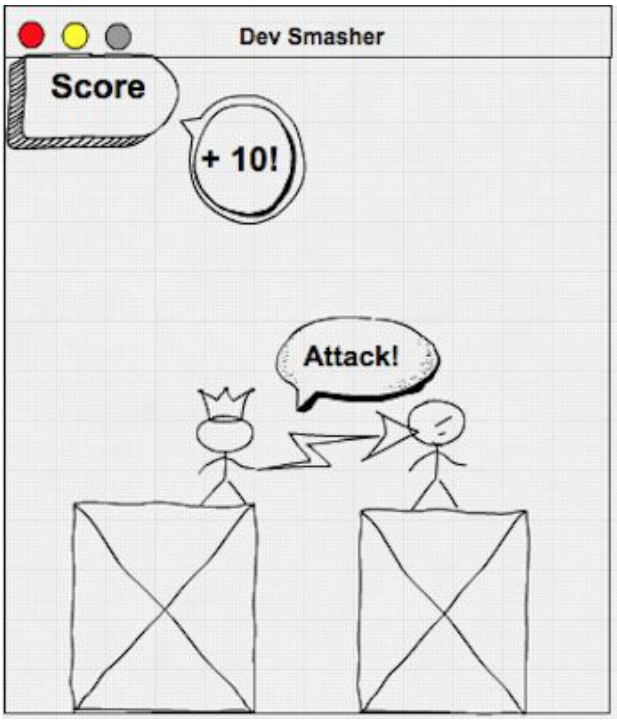
There will be a mySQL relational database that will store all of the accounts, along with their high score, and their current saved state within the game. When new accounts are made they will be added to the database. For checking high scores the information will be grabbed from the database.

UI Wireframes:

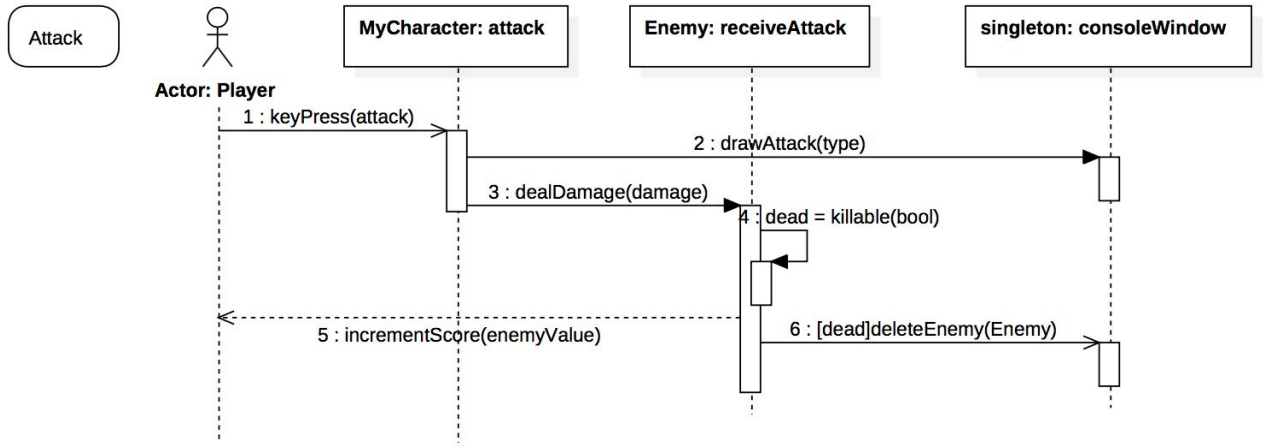
Main Menu:



Game Play - Attack:

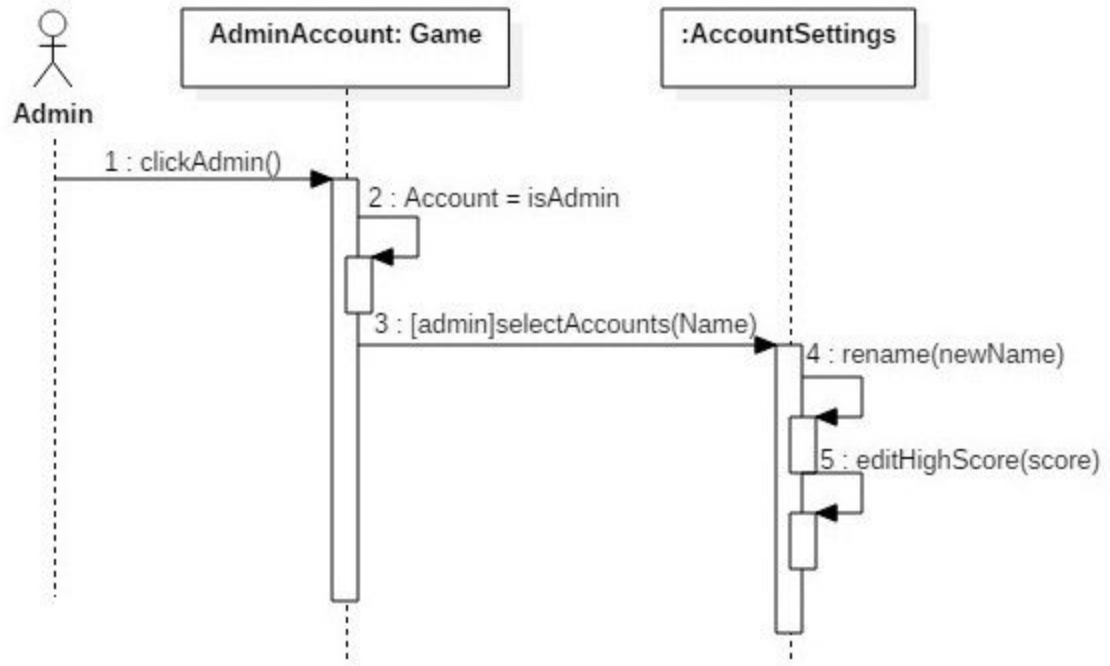


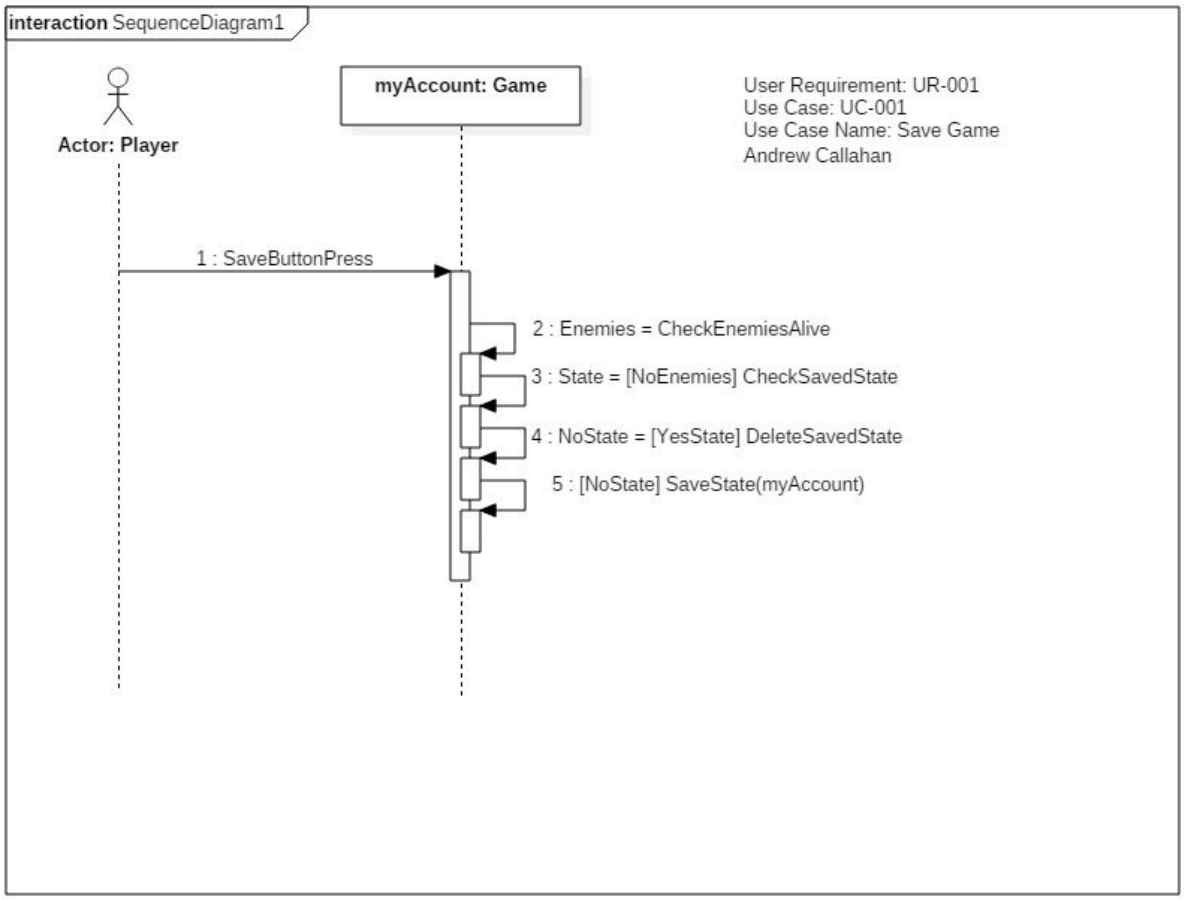
Sequence Diagrams:
Hayden Reich
Use Case ID #: UC-008
Requirement ID #: UR-008
Player Attacks Enemy

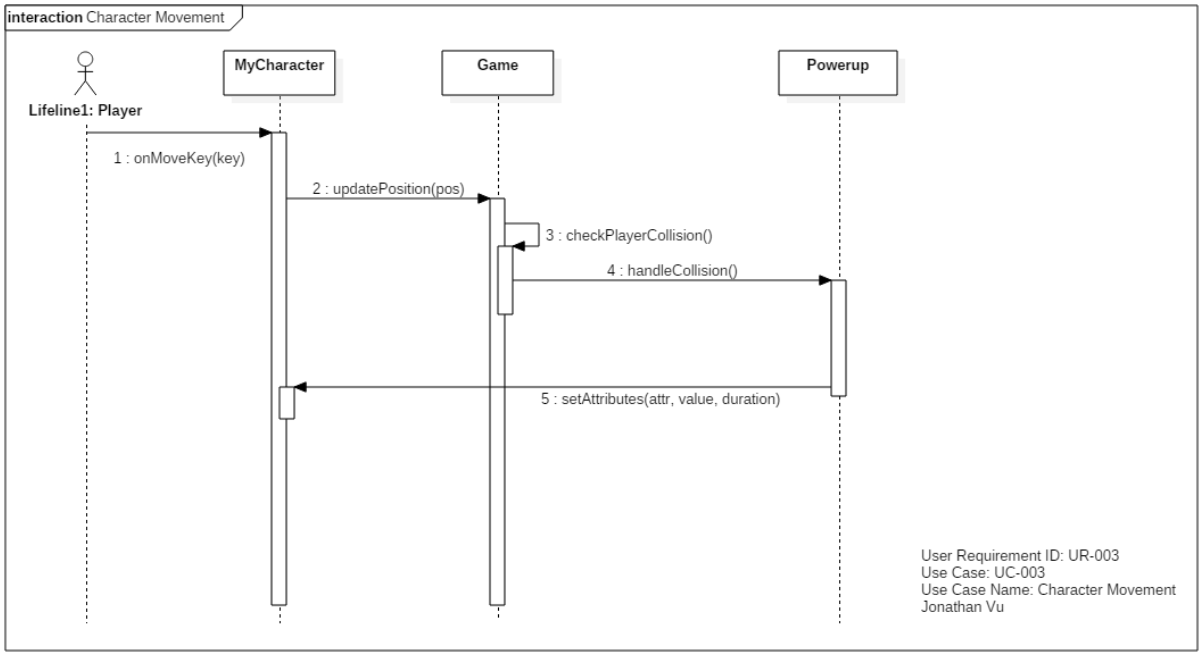
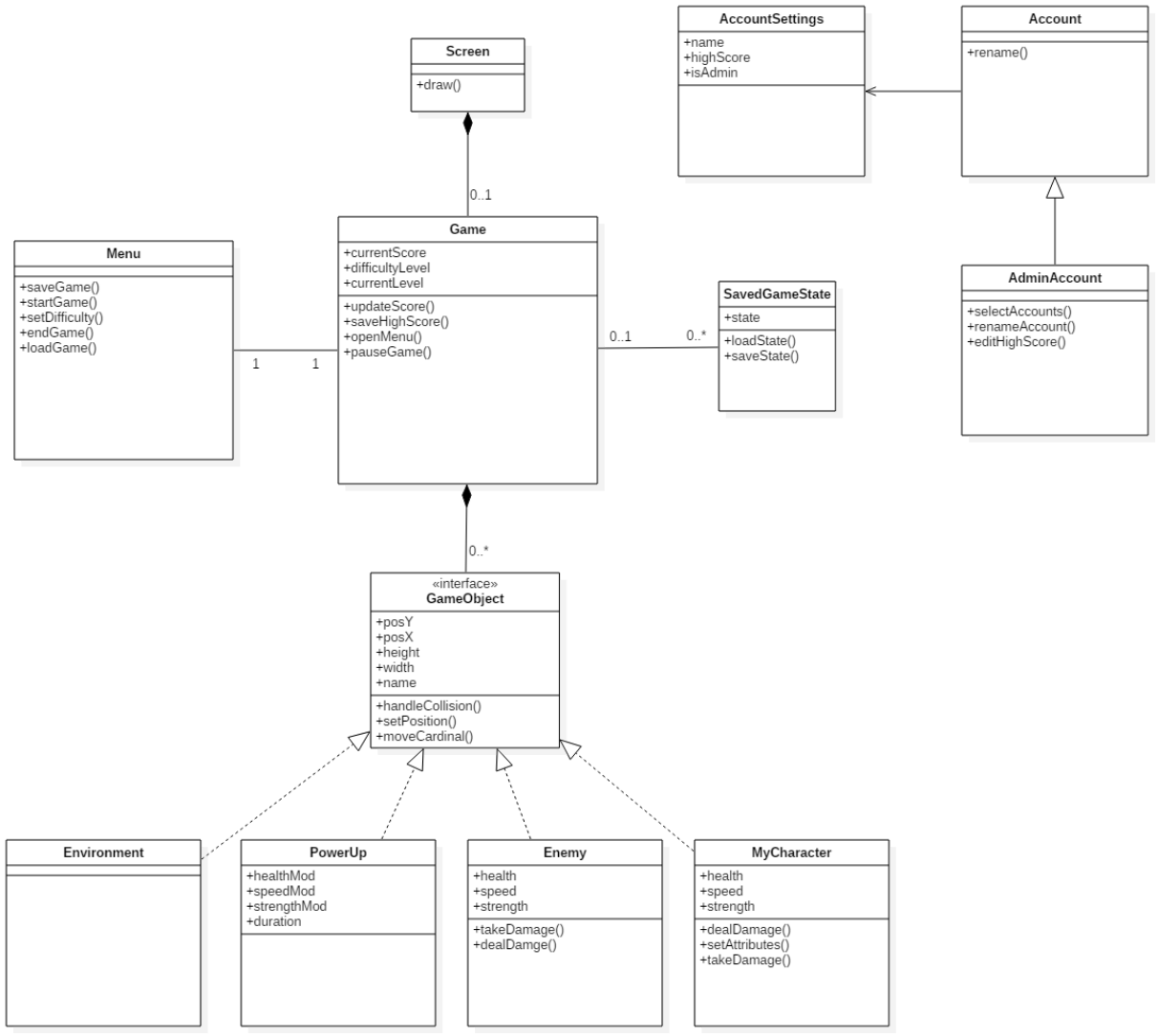


Kevin Flynn
Use Case ID #: UC-006
Requirement ID #: UR-006

Admin Access






interaction Character Movement
MyCharacter
Game
Powerup
Lifeline1: Player
1 : onMoveKey(key)
2 : updatePosition(pos)
3 : checkPlayerCollision()
4 : handleCollision()
5 : setAttributes(attr, value, duration)
User Requirement ID: UR-003
Use Case: UC-003
Use Case Name: Character Movement
Jonathan Vu

Class Diagram: