**Software Requirements and Design Document**

**For**

**Group 25**

Version 3.0

**Authors**:

Nicholas Tidwell,

Keith Van Dyke

Bryce Hart

# Overview

The system will be a 2D platform game with sprite graphics, audio, and the ability to control using a computer keyboard. The game will be part of the Platform Fighting genre where characters are animated sprites and various weapons and world objects can be utilized. The platform will be an arena that is unique and randomized upon each play. Players survive until their health reaches zero or they leave the bounds of the arena. In either case the other player wins. General settings for the game may be changed by the user. The game is started from the Main Menu. At the end of each battle, the choice is given to either play again or go back to the Main Menu.

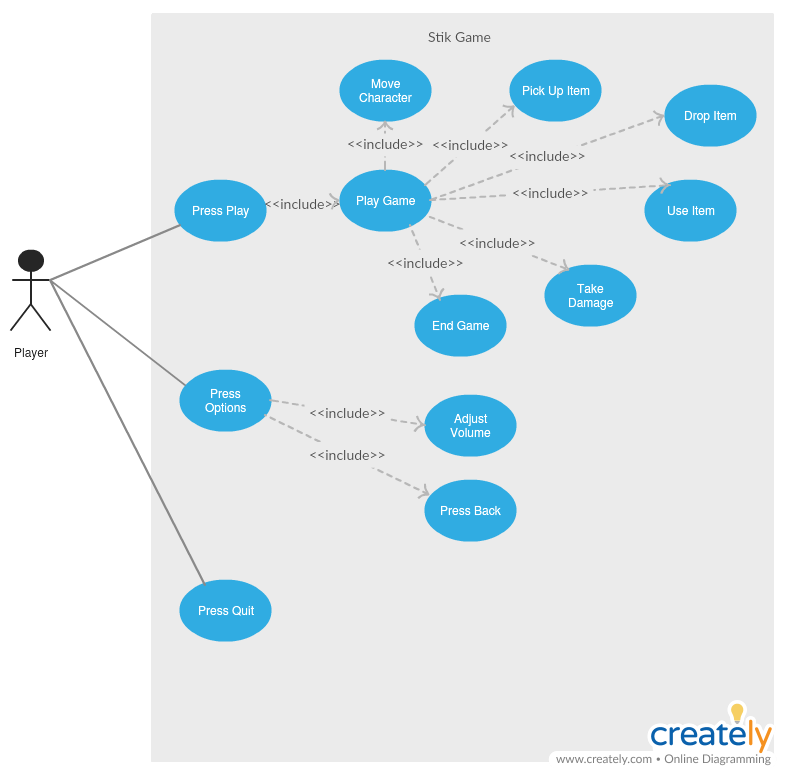
# Functional Requirements

1. *Medium - Game shall contain graphics, audio, and keyboard controls.*
2. *Low - Levels shall be unique and randomized.*
3. *Low - Game shall allow for consecutive plays.*
4. *Medium - Game shall contain settings.*
5. *Medium - Game shall show a HUD*
6. *Medium - Game shall include hazards that affect the player besides other players.*
7. *High - Game shall allow for players to move, jump, use weapons, and equip items.*

# Non-functional Requirements

* *Game shall be scripted in C#.*
* *Game shall be developed in Unity2D.*
* *System shall have minimal user interface that does not contain any information that is not necessary to the game itself*
* *UI shall be controlled by Keyboard (Mouse to aim if remote multiplayer is implemented)*
* *System shall run at minimum of 60fps;*
* *System shall require minimum processing power - (IE no more power than a mobile CPU)*
* *System graphics shall run at 60fps without the game skipping over any of the frames, causing the graphics to look “jumpy”.*

# Use Case Diagram

**

***Press Play***

*· Entry Condition: User Clicks on either Play button from Main Menu*

*· Normal Flow: Corresponding Play button is activated*

*· Exit: Corresponding scene is loaded*

***Press Options***

*· Entry Condition: User Clicks on the Options button from Main Menu*

*· Normal Flow: User can adjust volume. User settings are updated and saved.*

*· Exit: User presses the Back button*

***Play Game***

*· Entry Condition: Either play button is activated*

*· Normal Flow: Scene is changed to Original or Random game mode. User plays the game.*

*· Exit: none*

***Move Character***

*· Entry Condition: User presses WASD keys (Player 1) or arrow keys (Player 2)*

*· Normal Flow: Character moves in the direction the keys are pressed and jumps when W or up key is pressed*

*· Exit: User releases movement keys*

***Pick Up Item***

*· Entry Condition: User presses F key (Player 1) or J key (Player 2)*

*· Normal Flow: Item is added to next available inventory slot*

*· Exit: User releases pick up key*

***Drop Item***

*· Entry Condition: User presses E key (Player 1) or K key (Player 2)*

*· Normal Flow: Current item is dropped*

*· Exit: User releases drop key*

***Use Item***

*· Entry Condition: User presses G key (Player 1) or L key (Player 2) to shoot and up movement key for jetpack. Character must have a jetpack or gun equipped*

*· Normal Flow: Gun shoots or jetpack turns on*

*· Exit: User releases shoot key or up movement key*

***Take Damage***

*· Entry Condition: User hit box collides with a bullet*

*· Normal Flow: Health is decreased by 10*

*· Exit: none*

***End Game***

*· Entry Condition: Either characters health reaches zero*

*· Normal Flow: User clicks restart button*

*· Exit: New game begins*

***Adjust Volume***

*· Entry Condition: User drags volume slider to desired position*

*· Normal Flow: Unity sets Master Volume mixer to value of slider*

*· Exit: Game music gets new volume*

***Press Back***

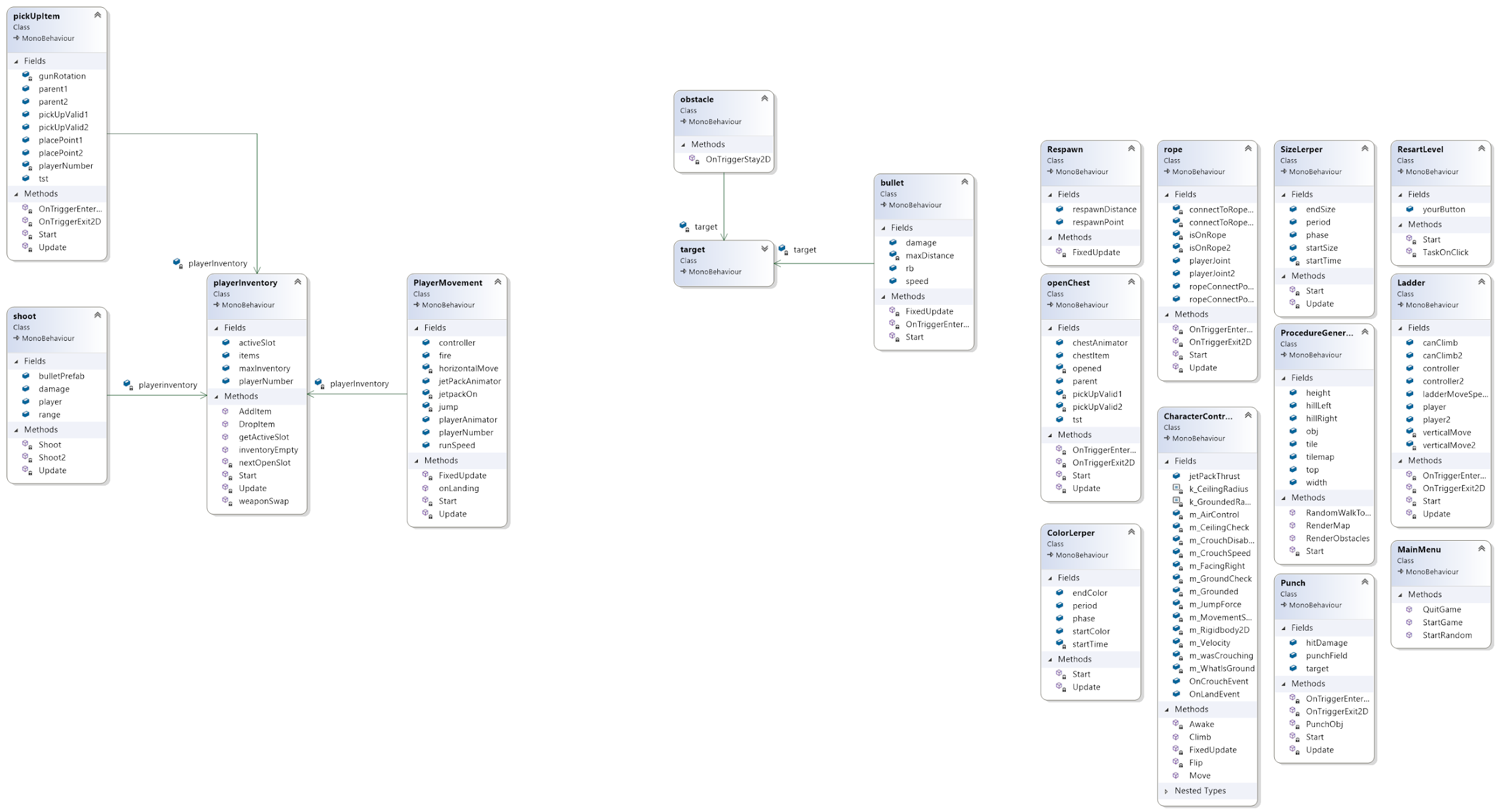
*· Entry Condition: User clicks back button*

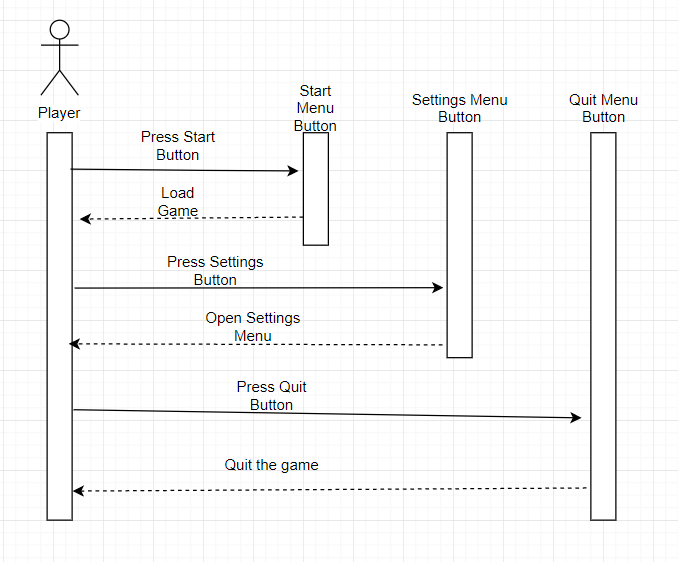
*· Normal Flow: Back button is activated released*

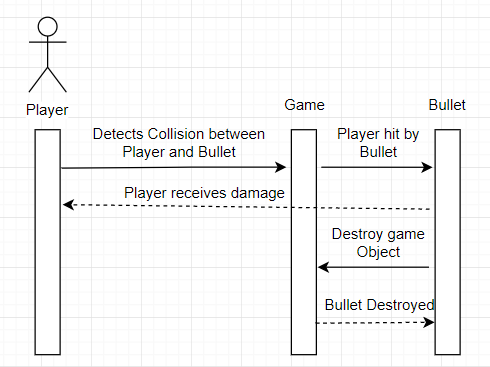
*· Exit: Returns to original main menu*

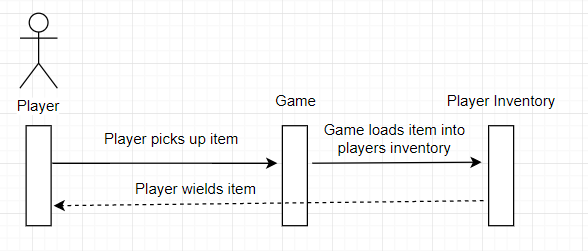
# Class Diagram and/or Sequence Diagrams

All of our scripts derive from MonoBehavior which is “The base class from which every Unity script derives. It offers some life cycle functions that are easier for you to develop your app and game.”

**







# Operating Environment

*Currently Developed in:*

* *Unity 2019.2.3f1*
* *Unity 2019.2.8f1*

*Tested on:*

* *Toshiba Satellite Radius P55W - Windows 10*
* *Windows Surface 4 - Windows 10 Pro*
* *MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)*
  + *macOS High Sierra Version 10.13.6*

# Assumptions and Dependencies

*Our game should be playable on any Desktop or Laptop and doesn’t rely on any dependencies.*