

# System Requirements for Hero Innovation Coliseum(H.I.C.)

Bobby Fowler

Hero Innovation Coliseum, or H.I.C. for short, is an arena style video game. A fast paced and competitive environment designed to test the skills and creativity of the players. Equipped with abilities, which they have personally designed and created, players must overcome and defeat each other to prove their worth in the coliseum.

## Table of Contents

1. [Introduction](#)
  - 1.1. [Purpose](#)
  - 1.2. [Scope](#)
  - 1.3. [Overview](#)
  - 1.4. [User Profiles](#)
  - 1.5. [Workflows](#)
2. [Requirements](#)
  - 2.1. [Requirements](#)
  - 2.2. [Stretch Requirements](#)
3. [Product Design Overview](#)
  - 3.1. [Architecture](#)
  - 3.2. [Data at Rest](#)
  - 3.3. [Data on Wire](#)
  - 3.4. [HMI/HCI/GUI](#)
  - 3.5. [Data State](#)
4. [Detailed Schedule](#)
5. [Appendix](#)
  - 5.1. [Sources](#)
  - 5.2. [Citations](#)
  - 5.3. [Resources](#)

## 1. Introduction

## 1.1. Purpose

H.I.C. is a video game that will provide a competitive environment that allows the player to use their creativity to create the character they wish to play.

## 1.2. Scope

H.I.C. is a one versus one arena format. Where players will compete to see who is the best. H.I.C. will be available to play on PC.

## 1.3. User Profiles

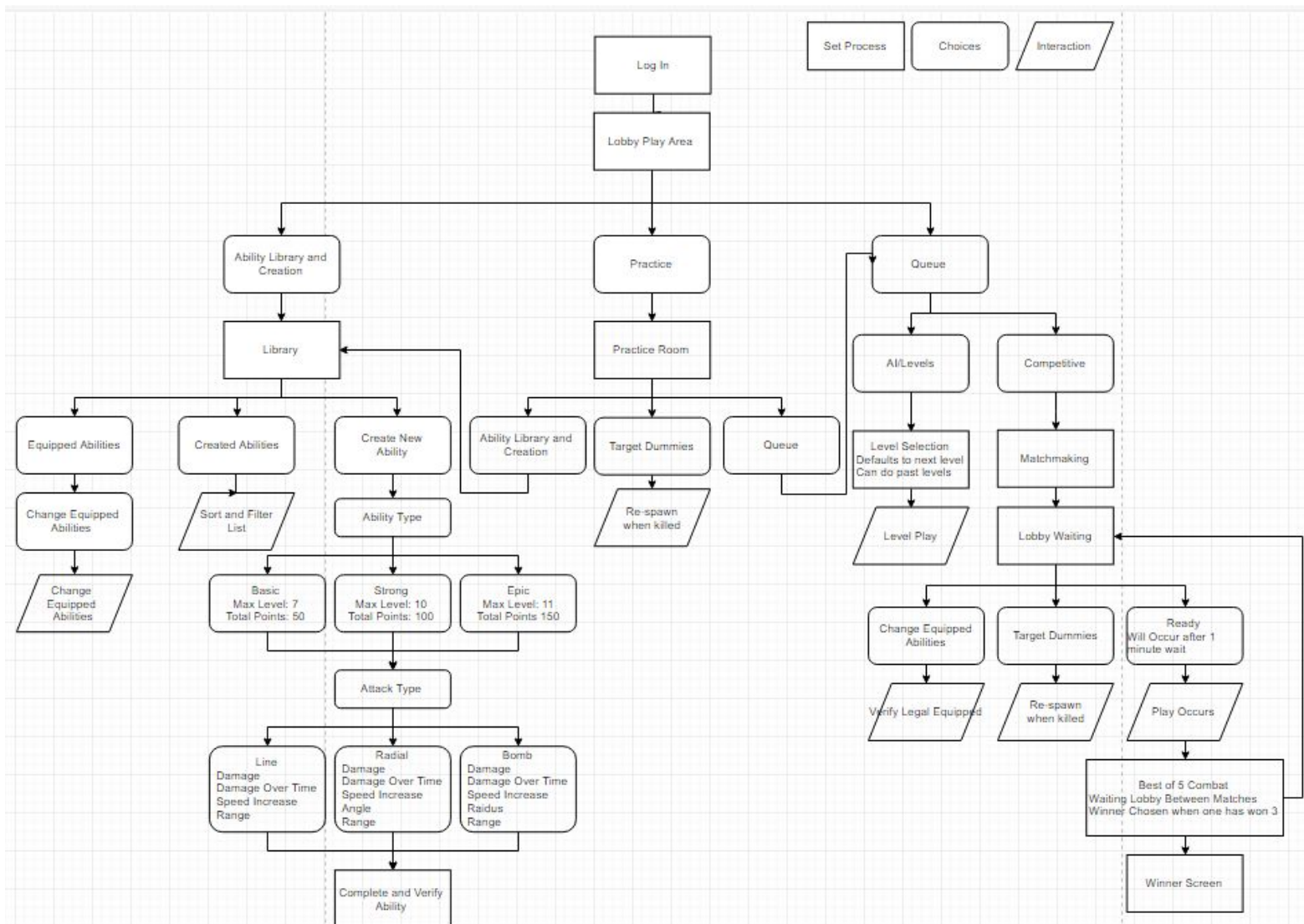
## 1.3.1. Admin

An administrator role that allows the user to enter various testing environments for the game.

## 1.3.2. Player

The common role for players of the game. They will follow the normal workflow [1.4.]

## 1.4. Workflows



## 2. Requirements

## 2.1. Requirements

#	Requirement	Verification
1	There shall be a login screen upon launch of the game	Launch the game and verify a login screen appears.
2	The game shall load the correct user information after login	Launch the game and verify the user's information is the same as last time logged in.
3	The user shall be able to view the library of created abilities.	Verify you can open the player's ability library
4	The user shall be able to change the currently equipped abilities	Verify you can update the equipped abilities of the player
5	The user shall be able to create a new ability	Verify you can create a new ability
6	The game shall verify that an ability is valid when it is created	Attempt to create an invalid ability and verify the game will not allow you to
7	The user shall be able to enter a competitive match	Verify you can enter a competitive match
8	The game shall allow users 1 minute before each game in a match	Verify the user can wait one minute before the game starts
9	Once a player has won best of five games the match shall end	Verify the match ends when three games are won
10	The user shall be able to attack another player by hitting with an ability	Verify an attack registers and deals damage to another player.

## 2.2. Stretch Requirements

#	Requirement	Verification
1	There should be an option to stay logged in between launches of the game	Toggle this option. Close the game and relaunch. Verify that you are taken past the login window and brought into the correct account without input required.
2	The user should be able to enter a training area	Verify you can enter a training area
3	Target dummies should respawn after being destroyed	Destroy a target dummy and verify it respawns

4	The user should be able to access ability library from the training area	Open ability library from the training area
5	The user should be able to add an effect to an ability	Verify an effect is applied to an ability and functions as expected

### 3. Product Design Overview

#### 3.1. Architecture

##### 3.1.1. Unreal Engine

This game engine will use multiple tiers. It handles most of the game logic of the game.

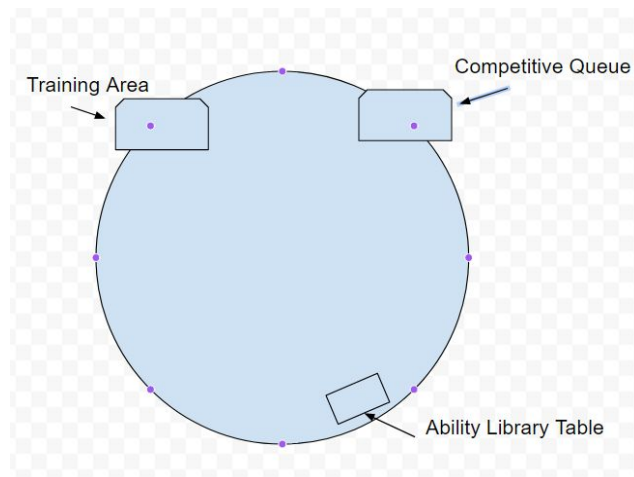
#### 3.2. Data

##### 3.2.1. Unreal Engine

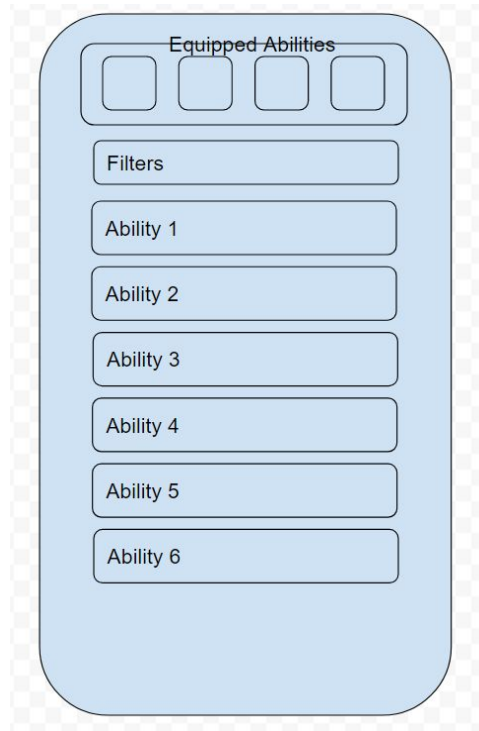
Unreal Engine has built in functionality for handling data at rest, data on the wire, and data state. H.I.C. will tap into this prebuilt functionality to handle data.

#### 3.3. HMI/HCI/GUI

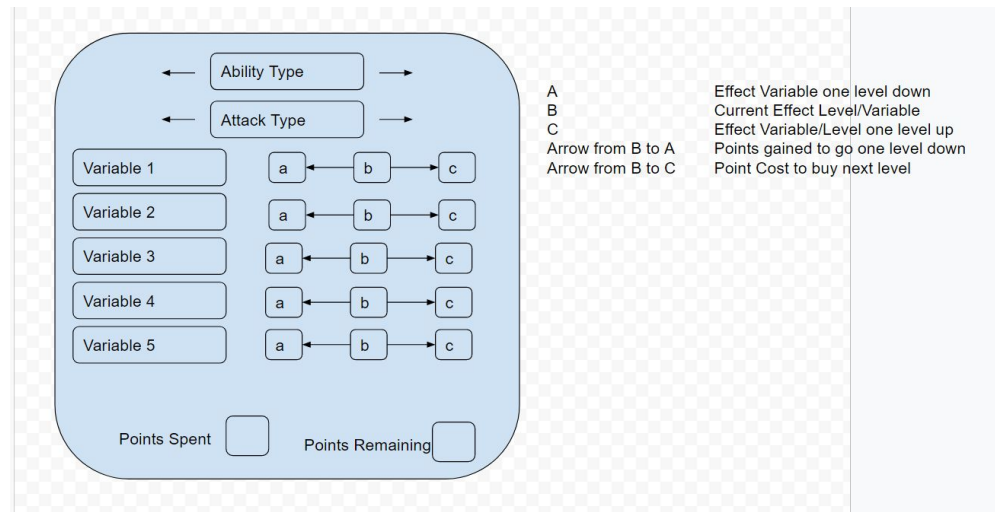
##### 3.3.1. Main Lobby



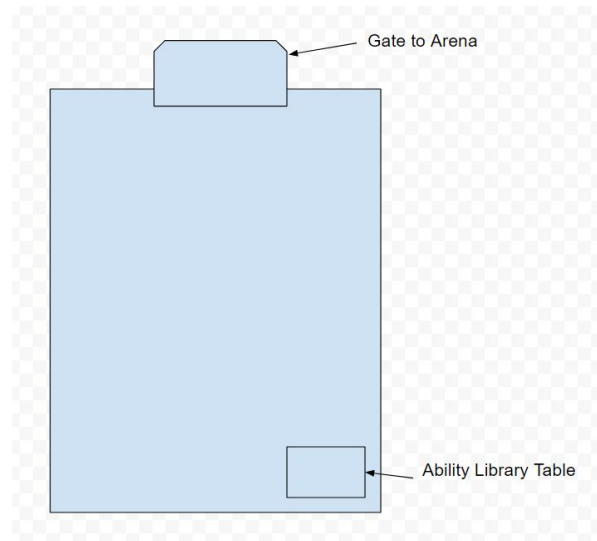
##### 3.3.2. Ability Library



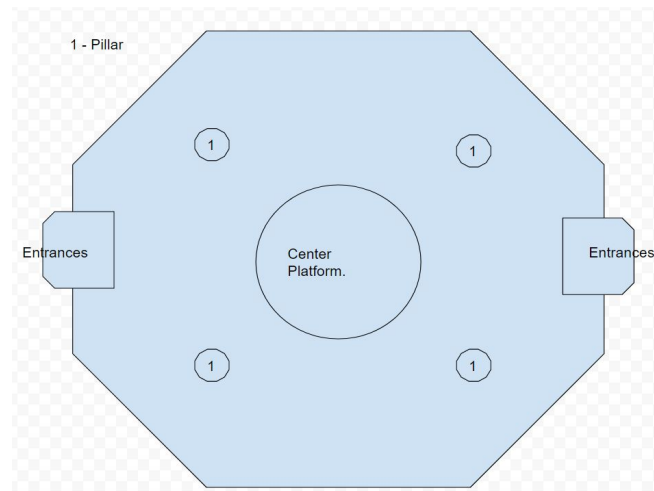
### 3.3.3. Ability Creation



### 3.3.4. Competitive Waiting Area



### 3.3.5. Competitive Arena



## 4. Detailed Schedule

- 4.1. Create the attack templates [2.1.11.]
  - 4.1.1. Hours to complete: 5 Hours
  - 4.1.2. Deadline: June 11th
- 4.2. Create Main Menu Interface [2.1.3., 2.1.7.,2.2.2.]
  - 4.2.1. Hours to complete: 5 Hours
  - 4.2.2. Deadline: June 14th
- 4.3. Build Match Mechanics [2.1.8,2.1.9.,2.1.10.]
  - 4.3.1. Hours to complete: 10 Hours
  - 4.3.2. Deadline: June 21st
- 4.4. Connect Multiplayer
  - 4.4.1. Hours to complete: 5 Hours
  - 4.4.2. Deadline: June 26th

- 4.5. Ability Library Basic Implementation: Create List, Display Equipped, and Change Equipped [2.1.3., 2.1.4.]
  - 4.5.1. Hours to complete: 5 Hours
  - 4.5.2. Deadline: June 29th
- 4.6. Ability Library Be able to create a new ability [2.1.5., 2.1.6.]
  - 4.6.1. Hours to complete: 10 Hours
  - 4.6.2. Deadline: July 3rd
- 4.7. Login Screen Interface [2.1.1., 2.2.1.]
  - 4.7.1. Hours to complete: 3 Hours
  - 4.7.2. Deadline: July 9th
- 4.8. Game saves created abilities and equipped abilities for the account [2.1.2.]
  - 4.8.1. Hours to complete: 2 Hours
  - 4.8.2. Deadline: July 11th
- 4.9. Testing and Debugging
  - 4.9.1. Hours to complete: 15 Hours
  - 4.9.2. Deadline: July 17th
- 4.10. Presentation
  - 4.10.1. Hours to complete: 1 Hour
  - 4.10.2. Deadline: July 19th
- 4.11. Reflection Document
  - 4.11.1. Hours to complete: 5 Hours
  - 4.11.2. Deadline: July 21st
- 5. Demonstration steps for verification
  - 5.1. Requirements
    - 5.1.1. Launch game and verify login screen appears [2.1.1.]
    - 5.1.2. Log-in and verify you can access ability library [2.1.3.]
    - 5.1.3. Inside the ability library verify that the user can create a new ability [2.1.5.]
    - 5.1.4. Verify that the user can not create an invalid ability [2.1.6.]
    - 5.1.5. Verify that you can change the equipped abilities inside the ability library [2.1.4.]
    - 5.1.6. Log out and log back in verify you still have the newly created abilities [2.1.2.]
    - 5.1.7. Enter a competitive match and verify if finds someone to play against [2.1.7.]
    - 5.1.8. Verify that the user has 1 minute to prepare between games [2.1.8.]
    - 5.1.9. Once game starts verify that ability collision is successful and as predicted [2.1.10.]
    - 5.1.10. Verify the match lasts until a player has won three games. [2.1.9.]
  - 5.2. Stretch Requirements
    - 5.2.1. If [2.2.1.] was implemented
      - 5.2.1.1. Toggle the option to stay logged in. Close game. Relaunch and verify it skips past login screen. [2.2.1.]



- 5.2.2. If [2.2.2.] was implemented
  - 5.2.2.1. Verify the user can enter the training area [2.2.2.]
  - 5.2.2.2. Verify you can access your ability library inside the training area. [2.2.4.]
  - 5.2.2.3. Destroy a target and verify it respawns[2.2.3.]
- 5.2.3. If [2.2.5.] was implemented
  - 5.2.3.1. Create an ability with an effect. Use it against a player in a game and verify the effect works. [2.2.5.]

6. Appendix

6.1. Sources

- 6.1.1. Official Unreal Engine Tutorials used and referenced
  - 6.1.1.1. [Introduction to Unreal Engine 4 Ability System](#)  
Used to gain a basic knowledge of how Unreal Engine 4 implements abilities
  - 6.1.1.2. [Unreal Engine 4 Mastery: Create Multiplayer Games with C++](#)  
Used to understand at a more indepth level how the game engine leverages C++ code. As well as, implementing multiplayer.

6.2. Resources

6.2.1. [Unreal Engine 4](#)

Game engine used to implement and visualize the various game features

6.2.2. Art Assets

6.2.2.1. [Environment Assets](#)

6.2.2.2. [Player Assets](#)