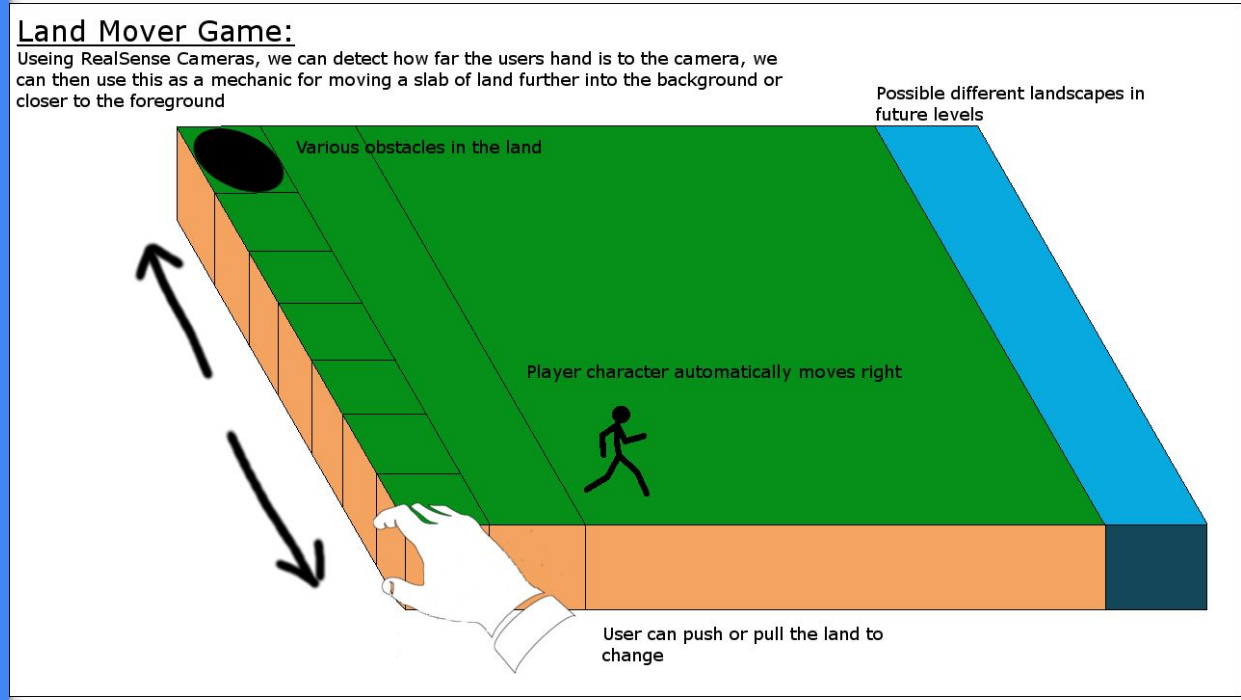
LandSlider:
A Depth Detection Game

Team 5

Concept

Concept

- Maneuver rows of mixed terrain tiles to assist an NPC to the end of the level
- Maneuvering rows would be controlled by Intel RealSense camera detecting hand gestures
- Tiles consist of terrain that will either impede or hurt the NPC while he progresses



Requirements

Purpose of Product

- To demonstrate and experiment how users might use gestures and depth to control object in a virtual space
- Exploring the problem of how users interact in the game's environment assets at different depths from the players perspective

Scope of Product

- Create a game that allows for the user to use hand and depth tracking to manipulate tiles that are at different depths from the player's perspective.
- Tiles will have different types of land terrain such as volcanoes, spikes, snowmen, forests, or rivers
- The player will move tiles arranged in rows using gesture detected by the RealSense camera
- At the end of the game a data file will be created detailing what transpired during the game

External Interface Requirements

- User Interface
  - RealSense enabled menu system complemented by traditional mouse enabled system
- Hardware Interface
  - Intel RealSense F200 camera, tracking range of 7 - 31 inches
    - Features: 3D scanning, gesture detection, object, hand and depth tracking
- Software Interface
  - Unity Game Engine interface with Intel RealSense SDK

Functional Requirements

- Menus
  - Main
  - Pause
  - Options
- Scenes
  - Game
  - Calibrations
- Data Log File

Performance/Quality Requirements

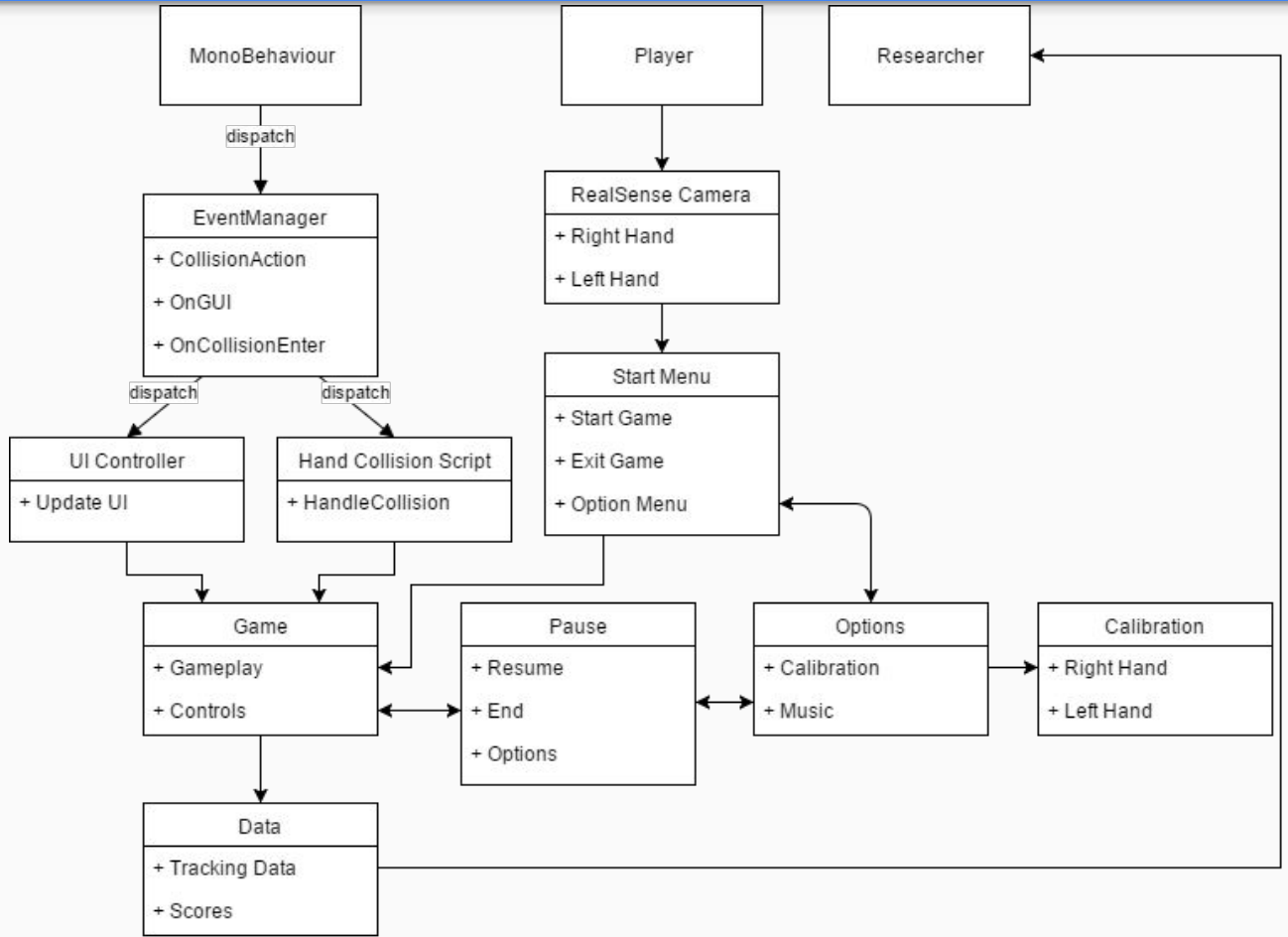
- The depth and gesture detection should be quick and responsive to not distract from gameplay experiences, minimal disconnects
- Paramount that the gameplay itself doesn’t result in crashes or errors
  - If errors or crashes occur they should be limited to camera detection issues that can be corrected by the calibration screen
- Tutorial level will be provided to help learn the basic controls of the game
- If the camera has stopped tracking the user should proceed to the calibration screen

Design

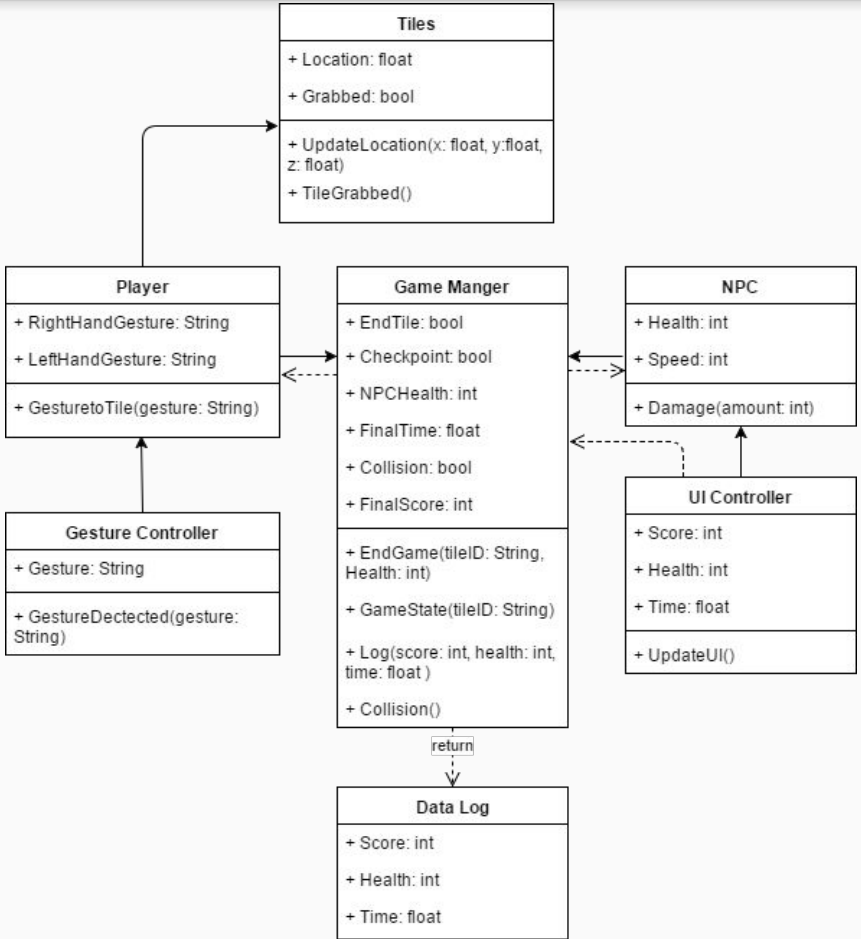
Design Drivers

- Using Unity version 5.0.0 was influenced by the the Intel RealSense SDK system requirements for Unity integration
- Unity was chosen over Unreal Engine because some team members had previous experience with the software
    - Unity supports C# and Javascript, C# was chosen for similar reasons to why we chose Unity
- RealSense was chosen over similar devices because it provide higher fidelity in detecting gestures and hands

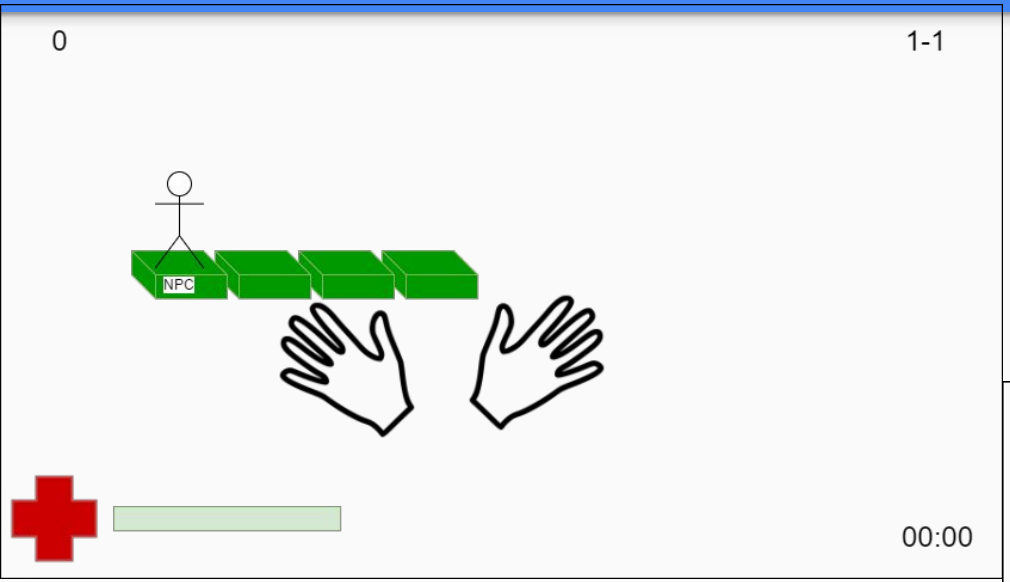
High Level System Architecture

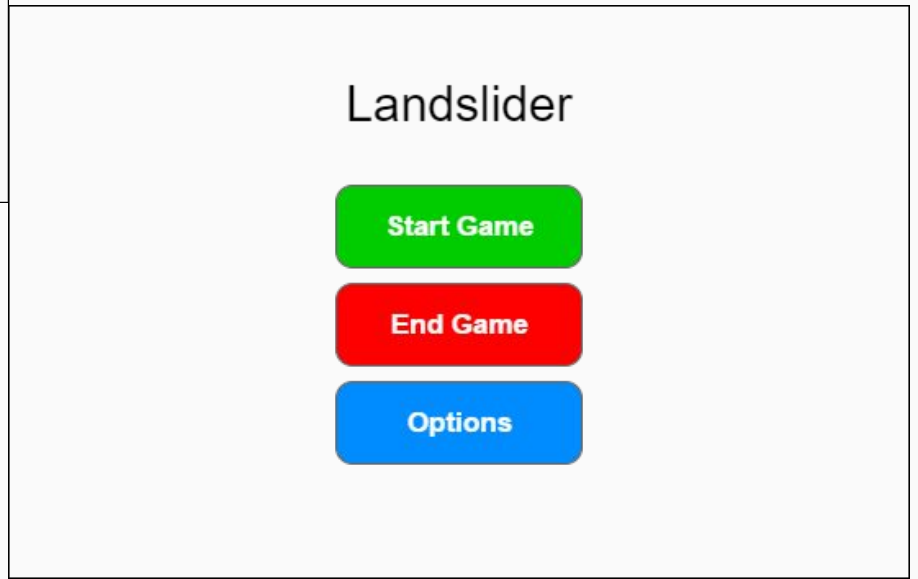


Low Level System Architecture

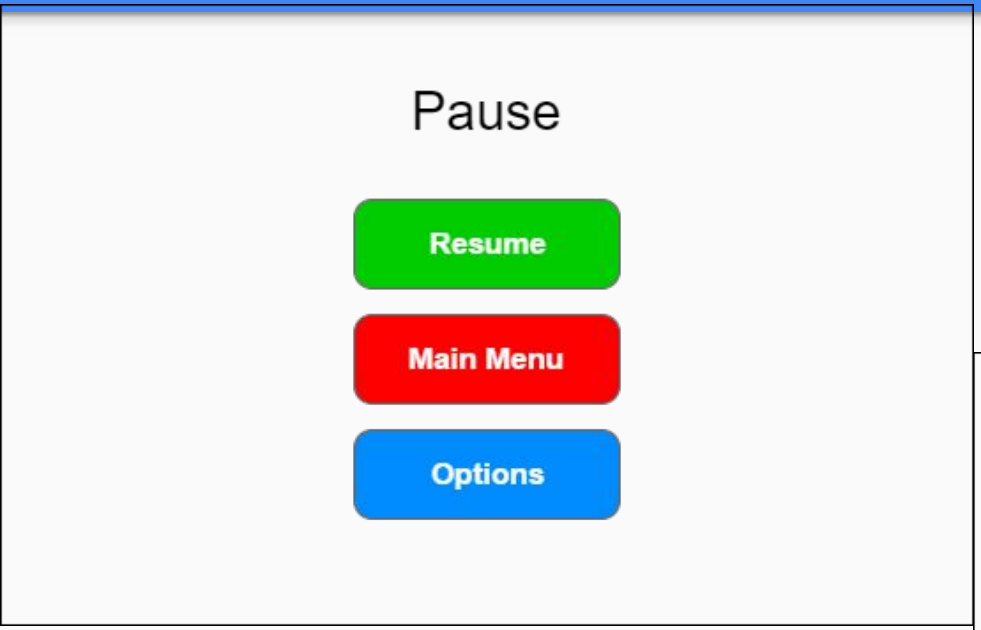


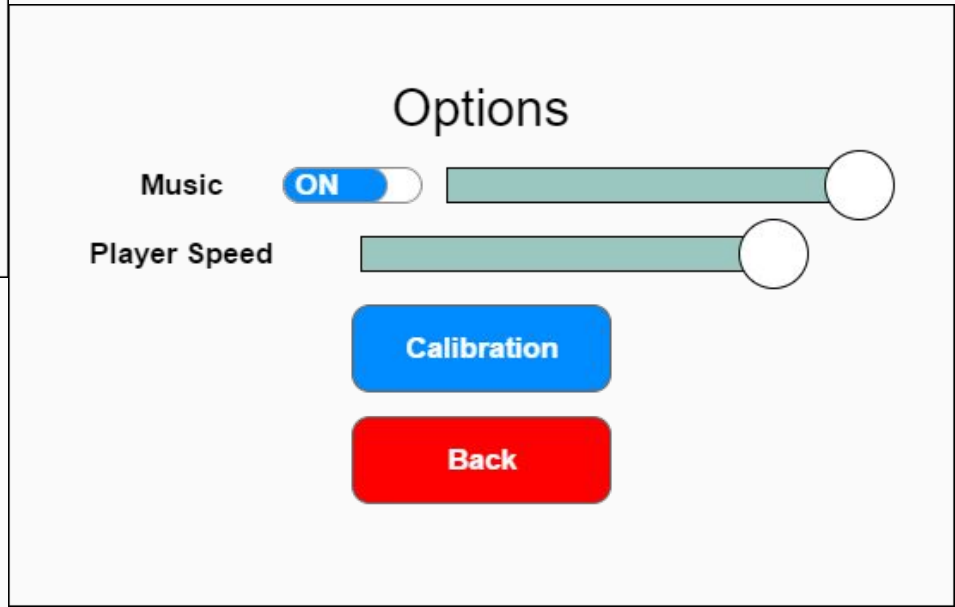
User Interface Design





User Interface Design





Test

Major Testing Issues

- The most important and difficult aspect of testing our system is making sure the user is continuously in correct orientation to the camera
- A wide range of recognition means that the camera can give the impression that the gestures and detection is identical at all ranges
- Lastly it is important that the RealSense enabled menus or game objects are not affected by any issues that could arise from the camera

Component Testing Strategy

- Bottom up testing approach will allow individual functionalities to be tested ensuring that each component works thus limiting the scope of errors
- After individual components are tested they will be integrated into the game and tested with the other components

Integrated System Testing Strategy

- Testers will be acting as end users, these testers will be from within the team and from outside
- Testing the full system will be similar to playing the full game, this means doing the following
  - Navigating Main Menu
  - Playing the level
  - Navigating sub levels
- Testers will ensure that all RealSense capabilities are working and are interacting with the game correctly

Testing Results

- It was discovered that the playability of the game was severely reduced with the inclusion of two hands, it has transitioned to one hand only
- The timer and score components worked when separated but when integrated into the both encountered errors and bugs
- The feedback about RealSense menus were that they seemed at first confusing but that was attributed to the type of interactions
- The learning curve of the game was obvious during testing which ensured that we included a tutorial level

Demo