

Version 1.0

August 20, 2015

Presented by: jUstin morritt

London, ON

Contents

[Dude Wheres My IQ? 2](#_Toc430185655)

[Theme / Setting / genre 2](#_Toc430185656)

[Core GamePlay Mechanics (Brief) 2](#_Toc430185657)

[Controls 2](#_Toc430185658)

[Monetization Model 2](#_Toc430185659)

[Monetization type 2](#_Toc430185660)

[Project Scope 2](#_Toc430185661)

[Game Time Scale 2](#_Toc430185662)

[Team Size 2](#_Toc430185663)

[Licences/ Hardware/ Other Costs 2](#_Toc430185664)

[Influences 2](#_Toc430185665)

[Television 2](#_Toc430185666)

[Games 3](#_Toc430185667)

[Target Market & target PlatForms 3](#_Toc430185668)

[The Elevator Pitch 3](#_Toc430185669)

[Target Contact Demographics 3](#_Toc430185670)

[Platforms 3](#_Toc430185671)

[What sets this project apart 3](#_Toc430185672)

[Core gameplay MECHANICS (detailed) 3](#_Toc430185673)

[Mechanic #1 – Acceleration / Deceleration / Jumping 3](#_Toc430185674)

[Mechanic #2 – Interaction With 3d Objects 3](#_Toc430185675)

[Mechanic #3 – Pickups 3](#_Toc430185676)

[Mechanic #4- Compile System 4](#_Toc430185677)

[Story and gameplay 4](#_Toc430185678)

[Story (BRIEF) 4](#_Toc430185679)

[Story (Detailed) 4](#_Toc430185680)

[Gameplay 4](#_Toc430185681)

[Assets 4](#_Toc430185682)

[2D Textures 4](#_Toc430185683)

[Art Samples 4](#_Toc430185684)

[SOUND (Ambient) 4](#_Toc430185685)

[SOUND (PLAYER/COLLISION 5](#_Toc430185686)

[CODE 5](#_Toc430185687)

[ANIMATION (Environment) 5](#_Toc430185688)

[ANIMATION (Character) 5](#_Toc430185689)

[Menu and Game Layouts 5](#_Toc430185690)

# Dude Wheres My IQ?

## Theme / Setting / genre

“Dude Where’s My IQ?” is a …

## Core GamePlay Mechanics (Brief)

- Acceleration / Deceleration with movement, jumping

- Interaction with 3D objects

- Compile System (acts as a timer)

### Controls

The player will use the mouse to aim around and use the (E) button to engage with items.

Movement of the player will consist of the traditional WSAD keys.

W (forward), S (back), A (strafe left), D (strafe right). The last key used will be the (I) key to use inventory.

## Monetization Model

### Monetization type

This will be a free to play game.

## Project Scope

### Game Time Scale

I will be getting a little under 4 weeks total Coding Preparation. This includes everything from collecting all of the assets to programming it all into a functional game.

### Team Size

Team will consist of me only.

### Licences/ Hardware/ Other Costs

All assets will be using 100% open art. There will be no costs except for our time. As far as Hardware goes, this game should run on any windows computer.

## Influences

### Television

Survivor Series

### Games

This Game Implements some well-known methods found in many survival games including “Rust” and “Dead Island”.

## Target Market & target PlatForms

### The Elevator Pitch

The life of a tiny robot cleaning bugs of a programmer is now you! You must help the programmer compile his program before a total motherboard meltdown occurs.

### Target Contact Demographics

Any age group will find this game fun and challenging, this should be a good game to burn a little time on and study some great mechanics that were put into place.

### Platforms

This game will be solely released on the PC.

## What sets this project apart

This game will put the player on the edge of their seat while they rush around trying to compile the program. The cartoony 3D look will keep the players eyes peeled as they search the motherboard of randomness. There will be core components of the motherboard placed for the nerds out there that will recognize them.

## Core gameplay MECHANICS (detailed)

### Mechanic #1 – Acceleration / Deceleration / Jumping

Players will have a realistic speed and feel. There will be Gravity in the game that will feel realistic and be to par with most modern first person games. There we be an implemented jump that will force the player off the ground for a jump. A player will Accelerate to a steady jog speed and then decelerate to 0 after the W or S key is let go.

### Mechanic #2 – Interaction With 3d Objects

Players must eliminate bugs by interacting with them with the E key.

### Mechanic #3 – Pickups

There will be speed boosts randomly that will increase the players speed thus giving a bonus to helping you complete the compilation process.

### Mechanic #4- Compile System

There will be a Compilation bar on the right side of the screen that will start out empty at the start of a game. There will also be another bar that has do with how many bugs are in the program at the time. If the bugs reach the max amount, you will have a full motherboard meltdown and lose the game.

## Story and gameplay

### Story (BRIEF)

- Trapped on a motherboard!

- Eliminate those bugs before a nasty motherboard meltdown occurs!

- Help a programmer compile a program!

### Story (Detailed)

You are a mini robot that navigates through the motherboard of a programmer that is trying to compile a program. You only exist during the compilation process so it is your goal to eliminate all the bugs to let the program fully compile.

### Gameplay

Navigate around the Motherboard by climbing and jumping around finding all the bugs. Help the Programmer get through the full compilation process by keeping the total bugs in the program to a minimum. If you let too many bugs through then there will be a motherboard meltdown and you will lose the game.

## Assets

### 2D Textures

| Resource | Role | Source |
| --- | --- | --- |
| Buttons | Interaction, Navigation | Created by Photoshop CC |
| Backgrounds/Menus | Visual appeal / Emersion | Created by Photoshop CC |
| Sprites | Character Animation/ Effects | Created by Photoshop CC |
| Billboards | Transistors , Environment | Created by Photoshop CC |

### Art Samples

These are in progress.

### SOUND (Ambient)

There will be eerie ambient style music going on in the background to keep a player on edge.

### SOUND (PLAYER/COLLISION

There will be sounds when a player engages with specific objects. These will consist of killing a bug, getting a speed bonus… and many more.

### CODE

Code will consist of a lot of Boilerplate code that stems from the Introduction to 3D programming. I will be creating many different classes for Screens as well as for Buttons. There will be a player class and an inventory class. A “State Machine” will be created to handle all of the different states the game can be in. Depending on what state the game is in certain draw calls will be called as well as certain updates will be updated. I will be customizing as many shader calls as I can as well as implementing a geometry shader for billboards.

### ANIMATION (Environment)

Not much in terms of environment animation. The buttons will have a grow and shrink effect when you hover over them. Animation will be mostly in the form of items moving around in the game. Depending on the time I get I will implement as many sprite animations as I can.

### ANIMATION (Character)

#### Player

The Player will not inherently have any animations you can see because it is a First person game. There will however be a nice silhouette of a shadow of your player on the ground. There will be some reflective surfaces that you will be able to see yourself in.

### Menu and Game Layouts

**Main menu** will consist of 4 buttons. The play button which will take the player straight to the game. A music toggle button, which will turn the music on or off. A sound toggle button which will turn sound fx on or off. **Pause screen** will have 2 buttons. One will allow the player to restart. The other will allow the player to quit. Pause screen will also have the word “PAUSED” in the middle of the screen. **Game Screen** will have 2 bars on it, they will indicate the amount of bugs in the game as well as the total compilation status.