**Project Analysis Document – Joseph Henry**

**Program:**

**Program Idea**

Network Jumper is a side scrolling game in which the player has to jump onto platforms for as far as possible without falling into gaps or before the screen has scrolled past the player. The platforms will be placed on the screen with procedurally generated dimensions. This game is designed for a younger audience.

**Why the task is suited to computational solutions:**

The game will take in user inputs to change the position of a character, it will create a connection between two clients over a network, generate the map procedurally.

**Stakeholders:**

**Stakeholders for the program**

Joseph Hurr

James Lowe

Ethan Cox

These people will test my game for me at each prototyping stage and give feedback on alterations I could make to improve the function of my game.

**Stakeholder Meeting Minutes**

Date – 14.10.19

Attendees – Joseph Henry, Joseph Hurr, James Lowe, Ethan Cox

Agenda – Explain concept projects to rest of the group, then discuss ideas for possible improvements to our projects based on the concepts given.

Joseph Hurr – Rather than doing a completely procedural map, use a text file of pre-set map chunks and randomly chose one to stick onto the end of the previous chunk.

James Lowe – Use a peer to peer network connection rather than using a client-server network connection to implement the multiplayer.

Ethan Cox -

**Existing Similar Solutions:**

**Cup head**

Cup head is a side-scrolling, run and gun game developed for a range of platforms such as the xbox and pc. The game has the option to play in coop with another player and the player/s have to complete a large variety of set missions before they can complete the game.

**Flappy bird**

Flappy bird is a side-scroller mobile game where the player controls a bird attempting to fly in-between two green pipes without touching them.

The game scrolls sideways at continuous speed and has a gravity function. The map is procedurally generates gaps for the bird to be navigated through.

**Jetpack joyride**

Jetpack Joyride is another mobile side-scroller in which the player controls a man who is continuously moving to the right in an attempt to cause as much havoc in this continuous hallway as possible.

The game scrolls sideways at increasing speed and has a gravity function. It also has a procedurally generated course for the player to navigate through, with objects such as lasers, which force the player to change position or lose the game.

**Super meat boy**

In the game, the player controls Meat Boy, a red, cube-shaped character, as he attempts to rescue his girlfriend, Bandage Girl, from the game's antagonist Dr. Fetus.

The game scrolls with the player and has a gravity function and the ability to slow down a players fall by sticking to the wall. Super meat boy also has set map layouts.

**Final thoughts from research**

I like the idea of a base of a continuous scrolling speed but if the player is too far ahead then the scroll should be centred on the leader to prevent players hitting the edge of the screen. I also believe a basic gravity feature would fulfil the purpose needed for my game to become functional. I would like to implement a similar course generation as seen in jetpack joyride and flappy bird. A different assortment of obstacles each time the game is started. Personally, I do not think the set map layout is a good implementation to add to my game, as the game would lose its replay ability thus making it more unenjoyable for the players to play over time.

**The Game:**

**Essential Features**

* The game should allow two clients to connect to a server, this is a key feature as it will provide a way to implement a multiplayer feature.
* Multiplayer should be well synced across platforms to prevent any kind of bugs generated from players being out of sync, for instance one player going off the screen on one client but on the other it is perfectly on the screen.
* The map should be procedurally generated so that it is infinite and makes the experience for the player different in each game. (Makes the game more replay able).
* Characters should move in accordance to their client’s key inputs for instance d to move right and space bar to jump.
* There should also be a gravity feature so that when a player moves off a platform it doesn’t float but falls either to the nearest platform below it or off of the map.

**Limitations**

One limitation of my game will be that the smoothness of the game being displayed to the clients will be greatly dependant on the connection speed between the clients and the server.

Can’t be played across long distances as it only works via a LAN network connection.

The game will only hold two connections so there is a limit of two players on the game.

The game will only work on a windows or mac PC so is not portable compared to if it was able to be played on a phone for example. Also due to this it limits the size of the player base as not everyone has a PC to be able to run it on.

**Prototypes**

Prototype 1: Create a basic connection between two clients with the connection being stable and the screens being in sync from the connection. The player’s character will be generated with basic movement controls. All player movements will be displayed on both connections no matter if both clients are hosted from the same pc or from two separate ones.

Prototype 2: The second prototype will change the players movement controls to be more suited to the game for example they won’t be able to move down but instead will fall via gravity, they also won’t be able to float upwards but instead will have a jump feature. There should be a spawn area for players to move around in and player characters should collide with the environment around them but not including the other clients character.

Prototype 3: The map should still generate the same spawn area, however there will also

**Success Criteria:**

Prototype 1 –

* The program will allow for 2 players to connect to each other through sockets
* The multiplayer will have all clients in sync with a stable connection
* Controls will be WASD. These controls will affect the position of the client’s character.
* Movements are synchronised on both screens.

Prototype 2 –

* The spawn area should be generated for players to spawn on and move around.
* Collision physics implemented to the player and the environment.
* Gravity implemented into the game to make sure that when a player goes off of a platform they fall rather than floating.

Prototype 3 –

* The map will procedurally generate itself and be displayed the same on both clients.
* The screen will scroll at a set rate to show players more of the map.
* The game will only begin once both clients are connected to the server.

**Requirements for game to be playable:**

**Hardware Requirements**

* A LAN connection to the other client (via a router or switch)

**Software Requirements**

* Python with the pygame module installed