Software Major Project: Check 2

By: Tongyu Hu (436016921)
Teacher: Mr Shirlaw
Class: Software Design Year 12
Due date: Week 1 Term 2

Functions and Modules

Modules

Modules are software components that are independently developed to make up a total program. Modules usually contain one or more routines such as functions or procedures to serve their own unique purpose.

Below are a few modules in which are to be used in Zenith:

Name	Description
PlayerControlMovement	Game module that allows for player to provide key inputs on their keyboard to control the movement of the player within the system
PlayerNameInput	A system module in which allows for the player to input their username to connect to the server. It allows for the username to be saved on a server that will be loaded each time that they use the program
LoadNextScene	A system module that displays the next scene when activated that can be attached to a button for the initial menu for players to navigate through the program
PauseMenu	A game module that allows for user to pause the game while in progress to allow them to access options to restart, resume, quit and go to settings
HealthBar	Game module is connected to a slidebar that initially is set to 100 (full health) and as the player interacts with certain objects, the health decreases by increments of ten.
MovementNetwork	This game module allows for the players to view other players movements on their own device. Essentially allowing for the syncing of movements across a remote server in which players connect to.
PlayerJump	Allows the user to provide the key input of space in order to control the jump of the player object. It initially will check if the player is grounded through a physics.raycast system
SettingsMenu	This module operates the resolutions of the game in which it is to operate in, allowing users to select their preference based on their individual computers and also control whether the game is in fullscreen or not

Functions

Functions are a block of organised and preferably reusable code that should perform a single process or related action that exists within a module. Together, several functions are the basis of the processes in which a module executes.

Below are a few Functions in which will be used within Zenith:

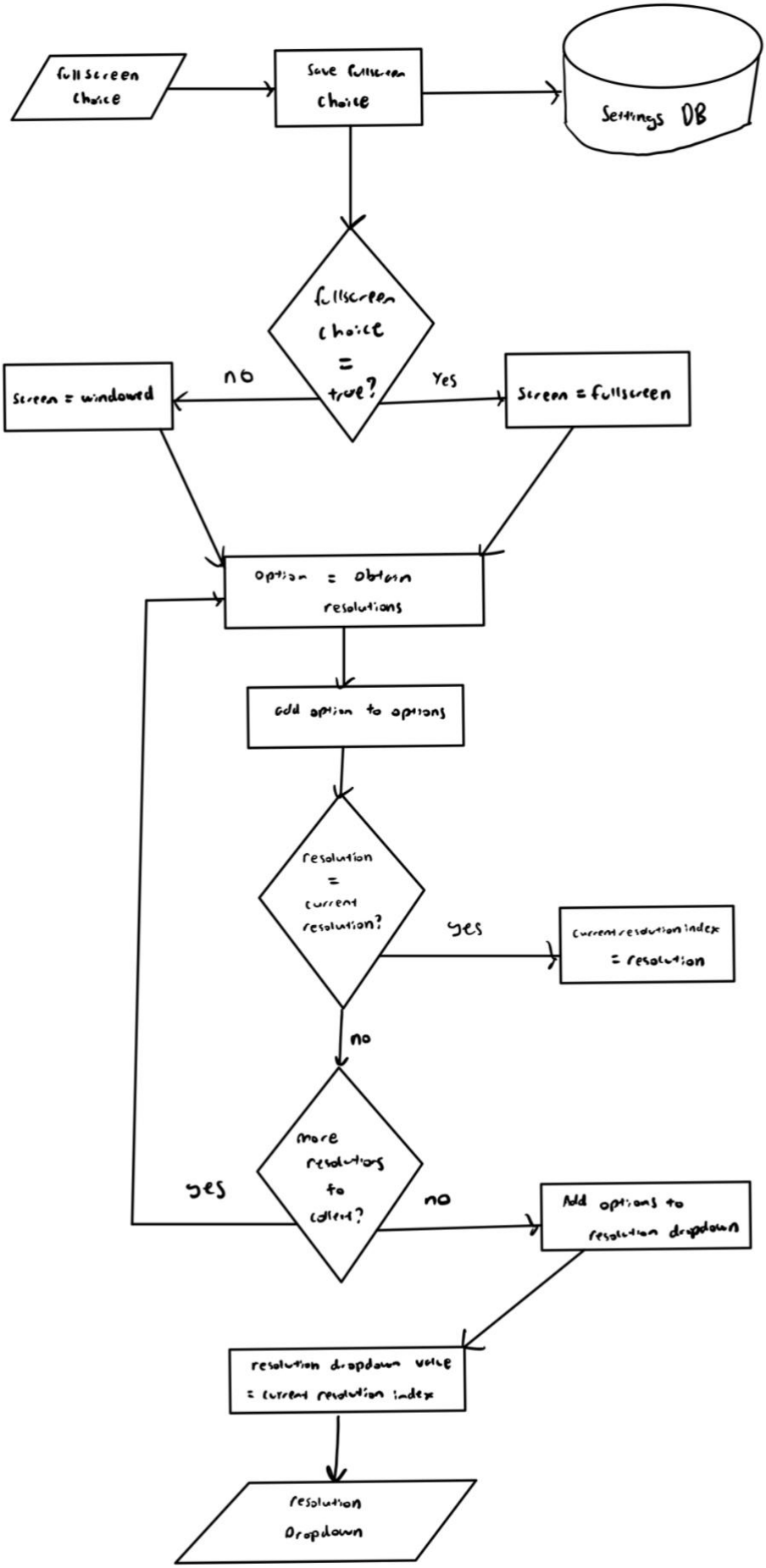
Name	Within Module	Description
GroundCheck	PlayerJump	This function is used to check whether the player object is touching the ground or not, then setting a Boolean variable to either true or false
Update	PlayerJump	This function is called repeatedly as the game progresses to allow for the player to jump if the GoundCheck returns true in addition with the key input of space
FindOpponent	MainMenu	This function allows for the player to connect to the remote server for multiplayer, while also setting different screen elements to be inactive and active during this process to communicate information to user
OnPlayerEnteredRoom	MainMenu	The function is called upon as the player joins the server and is within a waiting room. It checks whether both players are connected and loads the game
SetPlayerName	PlayerNameInput	Allows player to save their username that they use on a remote server so it will automatically display next time
SetUpInputField	PlayerNameInput	Checks if user already has an existing username and if not allows them to set it by calling the SetPlayerName function
TakeDamage	PlayerHealth	Allows for the health of the player to decrease
OnCollisionEnter	PlayerHealth	Checks if player has collided with a specified object and if so, calls the TakeDamage function to decrease the player health

Pseudocode

The pseudocode displayed below is representative of the ResolutionMenu module which is mainly allows for an array of compatible screen resolutions of a specific user's computer to be created and a dropdown menu for them to select their desired preference.

```
BEGIN SettingsMenu()

        Let resolutionDropdown = TMP_Dropdown
        Let resolutions = Array of available screen resolutions
        Let resolutionslength = length of resolutions array
        Get currentResolution
        Let options = List of strings
        Let currentResolutionIndex = 0
        Get fullscreenChoice
        Let i = 1

        IF fullscreenChoice = True THEN
                Screen.fullscreen = isFullscreen
        ELSE
                Screen.fullscreen = notFullscreen
        ENDIF

        WHILE i < resolutionslength

                option = resolutions[i].width + "x" + resolutions[i].height
                append option into options

                IF resolutions[i].width = Screen.currentResolution.width and
                                resolutions[i].height = Screen.currentResolution.height THEN
                        currentResolutionIndex = i
                ELSE
                        do nothing
                ENDIF

                i = i + 1

        ENDWHILE

        Add options into resolutionDropdown
        resolutionDropdown.value = currentResolutionIndex

END SettingsMenu()
```

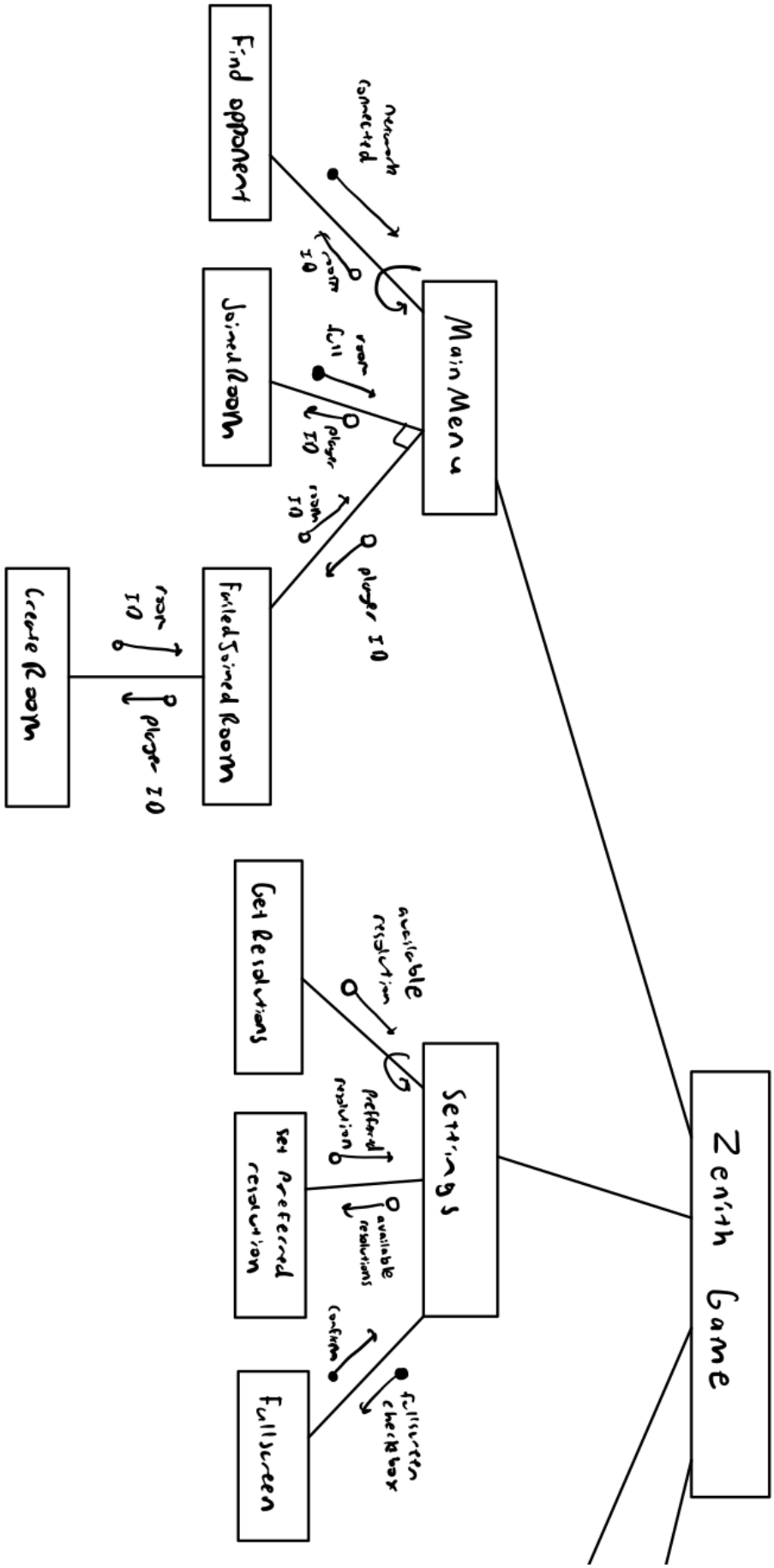
System Flowchart

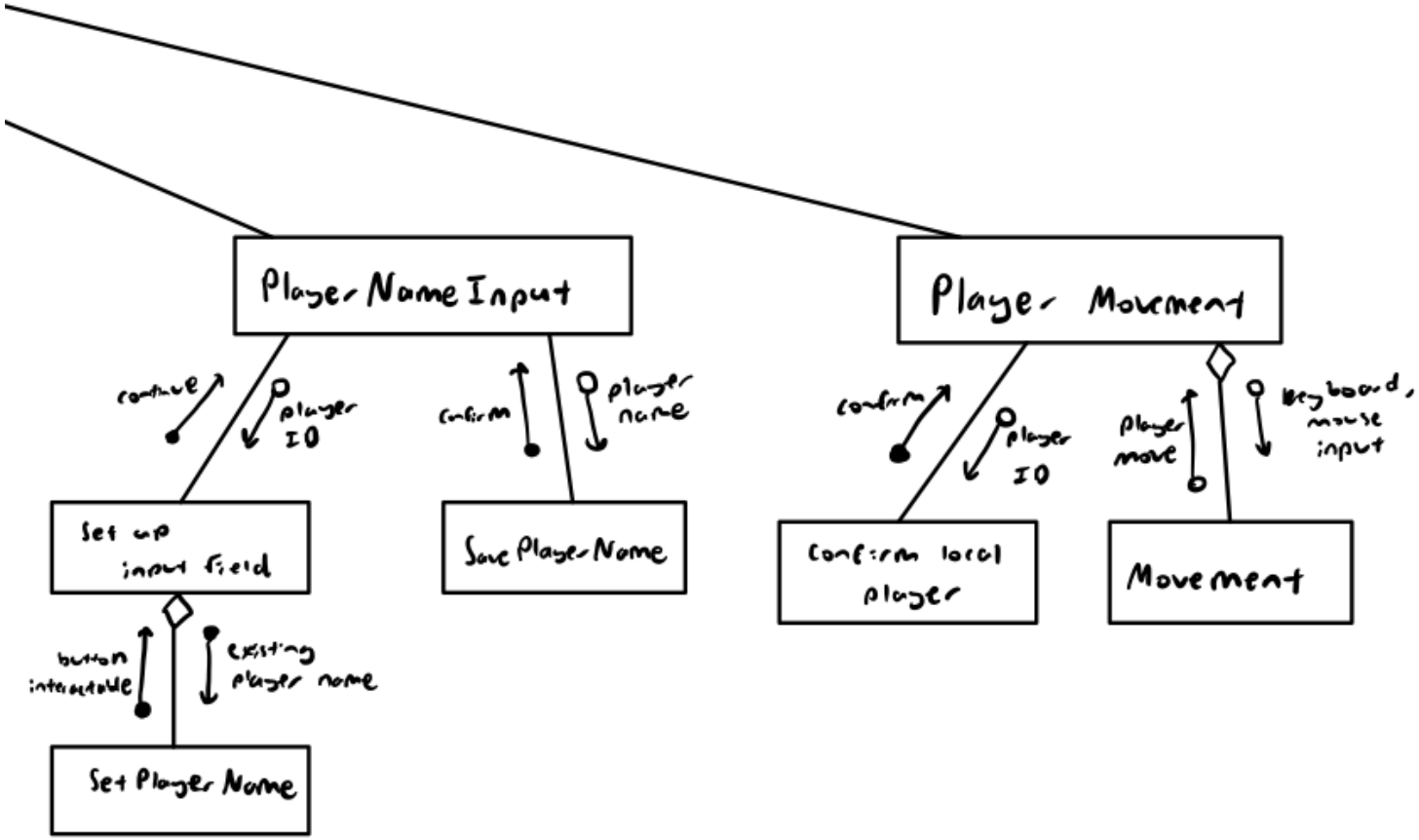


IPO Chart

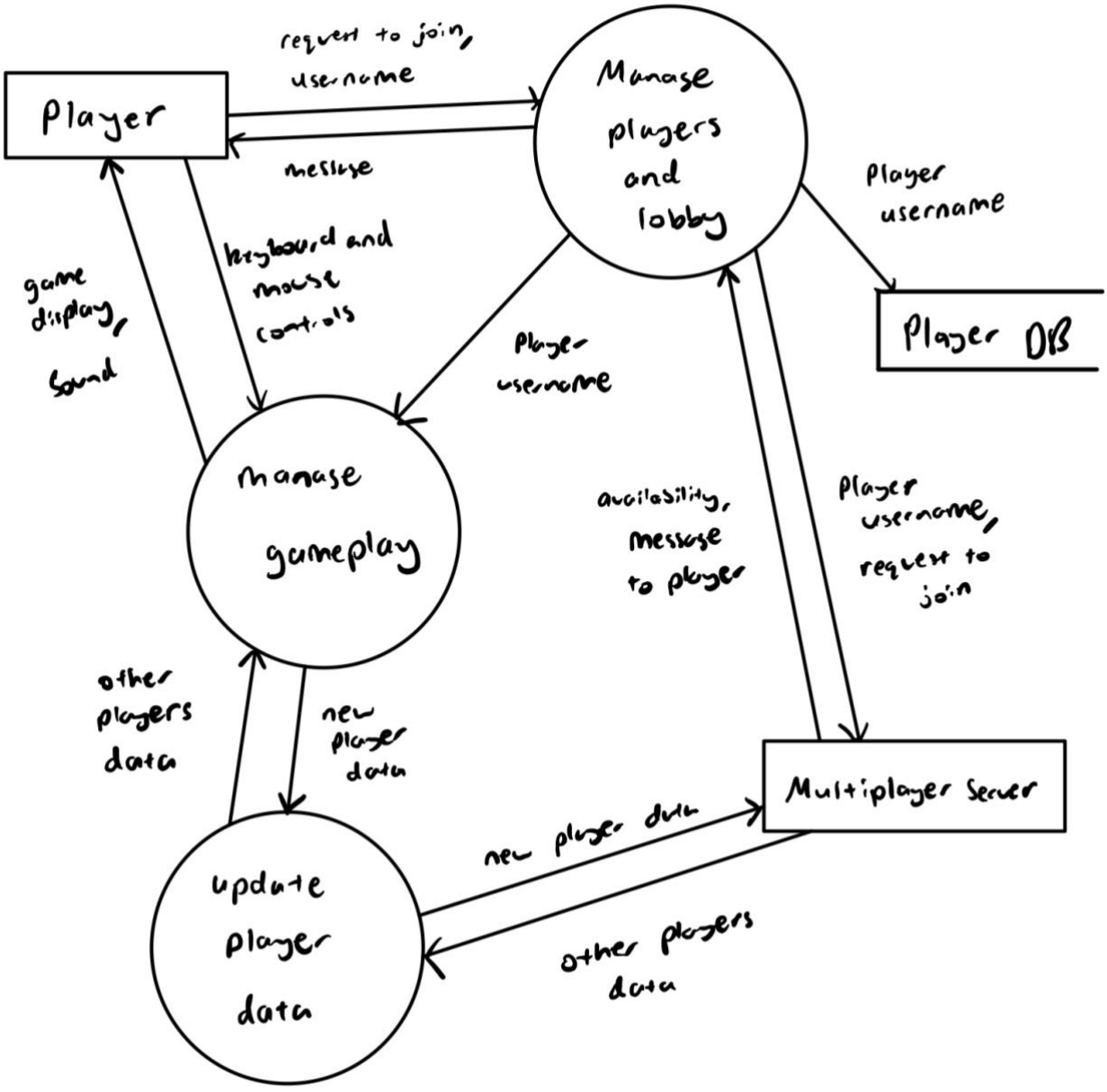
Input	Processing	Output
Initial game launch (Click to launch application from desktop/applications)	1. Check screen resolution and make a selection automatically 2. Load background music 3. Load all images, buttons and text 4. Run Zenith	Play background music Display Zenith game menu
Mouse click/selection of different buttons	1. Run specific function in which the buttons pressed is attached to 2. Load next scene or panel that corresponds to the button 3. Load new background music if necessary, depending on scene	Play background music Display new scene or panel
Escape/W/A/S/D/Space button/Mouse input	1. Check which button is pressed and execute accordingly 2. Check movement of mouse 3. Move player and camera according to movement of mouse and click of button 4. Update player location and camera location 5. Update background music	Play background music Display new scene or panel if ESC button pressed Display movement of player within the game and the rotating free-look camera

Structure Chart




Player Name Input
Player Movement
confirm
player IO
confirm
player name
confirm
player IO
player move
keyboard, mouse input
Set up input field
Save Player Name
Confirm local player
Movement
button interactable
existing player name
Set Player Name

Data Flow Diagram



Files and Data Structures

At these stages within the development of zenith, limited files have been utilised for the creation of the game. They include background music and image files that are added for the aesthetics and ergonomics of the game itself. In later stages, more files may be added in accordance with the needs of the game.

Name	Type	Description
MainMenu.jpeg	JPEG image file 	A JPEG file used for the background of the game Zenith within the main menu
HealthBar.png	PNG image file 	A PNG file used to display and improve the aesthetics of the health bar during gameplay
GameMusic.mp3	MP3 audio file 	An MP3 file attached to the AudioMixer of Unity to play background music in the main menu
MainMenuMusic.mp3	MP3 audio file 	An MP3 file that is also attached to AudioMixer of unity to play background music within the actual gameplay for the player
AnimationCube.mat	MAT material file 	A MAT file that is a material within the game used for animations of objects

Data Dictionary

Identifier (Data Item)	Data Type	Format	Number of bytes for storage	Size for display	Description	Example	Validation
resolutions	Array(string)		20	20	Resolutions available within specific computer	2560x1920	
options	List(string)		20	20	Contains a list of all available resolutions in string	2560x1920	
qualityIndex	Integer	N	1	5	Ranging from 1-7 to select quality of graphics	1	
volume	Floating Point	NNN.N	4	5	Ranging from 000.0 to 100.0 for volume	089.3	Must be between 000.0 and 100.0
isFullscreen	Boolean	X	1	1	Indicates game to be fullscreen or windowed	True	
movementSpeed	Integer	X	1	5	Ranges from 0-9 to change character movement speed	5	Valid between 0-9
strSceneName	String		255	255	Name of specific scene in game	MainMenu	
strTag	String		255	255	Name of tag attached to game object	Player	
distToGround	Floating Point	N.N	4	5	Distance of player	0.5	

					object from ground		
isGrounded	Boolean	X	1	1	Value to check if player is touching the ground	False	
MainMenuImage1	File		1500000	1000000	JPEG file for background of game	MainMenuImage1.jpeg/Images/Assets	Must be valid file and end with .jpeg
GameMusic	File		3500000		MP3 file for background music of game	GameMusic.mp3/Music/Assets	Must be valid file and end with .mp3
Vector3	UnityEngine.CoreModule	x, y, z	1000	1000	3D vector that has x,y and z components	3,5,10	Valid between -1000 to 1000 for all x,y,z
findOpponentPanel	Unity GameObject		20000	20000	A unity gameobject that represents a panel in the game		
MaxPayersPerRoom	Integer	N	1	5	Specifies maximum number of players in room	2	Valid between 2-4
transTarget	String		255	255	Specifies name of target in which camera is to follow	Player	
AnimationCube	file		50000000	50000000	MAT material file used within animations of objects in the game	AnimationCube.mat/Materials/Assets	Must be valid file ending with .mat

Platform/OS Considerations

The supported platforms and operating systems that Zenith will run on will either limit or expand its player base depending upon a larger or smaller number of supported devices and operating systems.

Platforms specify the range of devices that the game will run on, ranging from mobile phones to the gaming consoles, and even running in a web browser. A wider range will ensure a larger and more accessible game, thereby resulting in a higher player base yet will elongate the total development process due to the modifications that must be made for each platform.

The operating system in which software will run upon dictates the limitations that will be placed on diverse users, depending upon the hardware they use which is usually preinstalled with the companies preferred OS (e.g. Surface laptops and windows). As such, allowing the game to run on a large number of platforms means configuring it to work on each operating system efficiently and smoothly. Caution must also be made to not interfere with the OS itself and other applications running in the background, so that an ethical game can be made.

Since Zenith will be a multiplayer game, it should support a wider range of operating systems it will run on and subsequently, its release on differing platforms. Initially, the program will be available on both Mac OS, Windows and Linux, supporting the vast majority of all NSB students and normal users. This will be done through the Unity builds menu, as selections can be made in order for the program to be built for several operating systems. In consideration to mobile gaming and Zenith running on IOS and Android, it will be placed for later revisions and updates of the game as in the present moment, there is a constrained amount of time. In the end the main goal is to create a game which can run on laptops and desktops, which all NSB's should have access to and therefore, allow for the main objective (providing entertainment to establish relationships within NSB) of Zenith to be realised.

From the above considerations, the initial release will be relatively feasible and to a certain extent, easy to be accomplished due to the ease in which Unity allows for versions of the game to be built for different platforms and operating systems. As such, the game will be easily accessible to most students within NSB.