



Digital Studio Practice

VR Nursing Awareness Simulator

Gordon Johnson
K1451760

Table of Contents

Introduction.....	2
Tools and Technologies	3
Implementation	4
Conclusion & Future Improvements.....	7
References.....	8
Appendix One – GitHub Commit History	10
Appendix Two – Game Controller Main Components.....	23
Appendix Three – Setting and Getting High Scores.....	26

Introduction

Virtual reality is the first step into a fully immersive environment, with the technology continuing to develop at a tremendous rate. Due to my lack of prior understanding and knowledge of developing for VR, I was aware that this project would involve a steep learning curve, however I was confident I could rise to the challenge.

Providing a final product for a client, adds enormous depth and motivation to the project, even more so when that client is the School of Nursing. Ensuring that the School of Nursing, get a Virtual Reality solution that matches their vision and potentially helps in the training of future nurses, is a key element to this project.

Due to the benefits of regular testing while developing a product, after thorough research, I invested in a Microsoft Mixed Reality Lenovo Explorer HMD Headset. All testing will be conducted on this headset. As this is still a very new product, some unforeseen challenges may present themselves.

Repository

GitHub - https://github.com/LordGee/dsp_vrnursing_beta

Early Development Videos

Learning Interaction - <https://www.youtube.com/watch?v=SjOezdnbt0o>

Mini Game Prototype - <https://www.youtube.com/watch?v=kwoFdxxz-lc>

UI Interaction - https://www.youtube.com/watch?v=Y_I2lmRb9So

Mini Game Improvements - <https://www.youtube.com/watch?v=zLgcvMubpgc>

Player Stats & New Environment - <https://www.youtube.com/watch?v=hRbl5z5LVec>

Final Video

Play through of entire game - <https://youtu.be/gwweUO8yN1k>

Tools and Technologies



Unity 3D

After analysing the brief and discussing various options with the rest of the DSP Team, it was decided that the Unity 3D Game Engine (Unity, 2018), would be the main development tool we would use. This is the engine that most members with development experience, are familiar with. As the coding language for this engine is primarily C#, with minor patching it supports Microsoft Mixed Reality (Microsoft, 2017). The version of Unity 3D used, was originally 2017.2.0p2. In December, the project was updated to version 2017.3.0f3.



Visual Studio

The IDE used for the coding of this project, was Microsoft Visual Studio 2017. As the sole programmer, I am most familiar with this IDE, especially around the debugging and other integrated features that VS provides. There are alternatives that could be used for the development, such as “Mono Develop” or Jet Brains “Rider” but due to the complexities of this project, it was felt that using familiar applications would be beneficial.



Mixed Reality Tool Kit

Although not implemented in the final version of this project, the first iteration began with using this SDK (Microsoft, 2018). This provided an early opportunity to start learning the basics of VR development, whilst awaiting the release of Steam VR for Mixed Reality.



Steam VR

Switching to the Steam VR SDK was important, as it provided multi hardware compatibility, primarily including the HTC Vive, Open VR and now Microsoft Mixed Reality. This SDK provides access to many of the primitive VR functionalities, such as access to camera rigs, tracked controller, model renderers and interactions.



VRTK

The VRTK plugin was chosen for some of its compatibility features, such as the ability to implement multiple SDK's for various devices, such as the Oculus Rift. Once setup, VRTK will operate using the correct Vendor SDK, for the headset in use. VRTK also includes a simulator, this allows members of the group without a VR headset, to test the development of the project.



GitHub

The repository for this project, including a complete history of the development process (also found in Appendix One), can be found on GitHub. Git, is an invaluable development tool, showing each iteration of the project, with the ability to revert to a previous commit if required. It also ensures, that every device / user involved in the development, can always be working on the latest version of the project.

Implementation

Prior to starting the implementation of the project, I analysed the various UX and Game Design documentation, produced during these phases. The most valuable document, was the game flow diagram which provided a summary of events across a timeline, this ensured that events were implemented in the order and structure of the design.

The code structure, is organised into three types of scripts, the manager, objects and task scripts. Some of the manager scripts, implement and handle key aspects of the running game, while a subset of others, handle key information that is not always active.

Major Manager Scripts

Script Name	Purpose / Operation
Event Controller	The Event Controller, is a variation of the Unity messaging system (Unity, 2017). Functions are overloaded to allow for two types of events, one that can accept a float value to be passed through and one that doesn't. This allows functions anywhere within the code, to be listened for and triggered, without the need for finding components. This reduces overheads and improves performance.
Game Controller	The Game Controller, is the starting and end for the entire game. It controls the main game states, as well as variables such as scores, levels and timers. Where possible, all attributes that require an 'Update' function are managed within this class. The three main game states, are 'Brief', 'Playing' and 'Game Over' These are checked every frame, with the relevant functions being executed. This controller, also manages the players statistics, such as the Hydration, Energy and Fade levels, throughout gameplay. The main components of this controller, can be seen in Appendix Two.
Spawn Controller	The Spawn Controller, handles all objects spawn within the game, such as water, food, hazards and task objects. A set of spawn locations have been placed around the scene, this controller takes the locations as a potential position for the next object to be spawned, it also checks whether an object already occupies this location, ensuring that two objects do not randomly spawn on top of one another. Once a suitable, empty location has been found, this controller spawns a relevant object. When an object has been collected, the controller removes it from the scene and frees up the spawn location.
Task Controller	The Task Controller, operates alongside the game controller. It instantiates new tasks into the scene, as and when required and removes the appropriate task once completed.

Minor Manager Scripts

Script Name	Purpose / Operation
Canvas Controller	When needed, the Canvas Controller, updates the player statistics. These can be provided to the player at any given time. The constructor declares the canvas as not active, it is only opened when requested via the event controller. Information in the Canvas, is only updated when the player stats change, so if the player accumulates a point, an event is triggered telling the Canvas it is now time to update. This reduces the overhead of trying to update the canvas stats on every frame.
Constant Controller	The Constant Controller, provides a home for all the constant variables within the application, maintaining starting game values, tag and event names. For the latter two, this is useful as declaring string values can often lead to typo errors. Ultimately, a lot of time can be saved during development by reducing the need to remember these names.
Home Controller	The Home Controller, is primarily responsible for the handling of the high scores in the start scene. When the scene is loaded, the values are obtained from the 'Player Preferences Controller' and populated into the scene.
Player Controller	The Player Controller, handles controller interaction events, defining what happens when a button is pressed, depending on the current state of the game. For example, if the Controller Menu button is pressed, a function is called that triggers the Event Controller to action the listening function in the Canvas Controller script. Originally, this was used for the Pipe Mini Game, however, calls to these functions have been moved to the object specific scripts.
Player Prefs Controller	The Player Preferences Controller, handles the getting and setting of values on the local machine that the game is played on. These variables, are primarily used for maintaining player scores. As the Player Preferences Controller, only handles a small selection of data types, a persistent array of values that could be stored and retrieved was created, see Appendix Three.

Game Task One Implementation

This task involves collecting three bottles of pills, once all three bottles are collected, the game ends. The Task 1 script, remains active throughout the entire game, spawning a new object at certain points during the game, depending on the player progress within other tasks. The task object, needed to be interactable, so when the grip button is pressed it collects the object. Once the first object is collected, the next task is instantiated.

Game Task Two Implementation

Task two, requires a bed to be instantiated that does not conform to the presentation of the other beds in the scene. In this instance, three objects are placed around the bed. One is in the wrong place and should be placed in the correct position, the other two are rubbish that should be placed in the bin. Feedback is presented to the player, using 3D Text just above the bed, indicating how much time is left and how many completed objects have been dealt with.

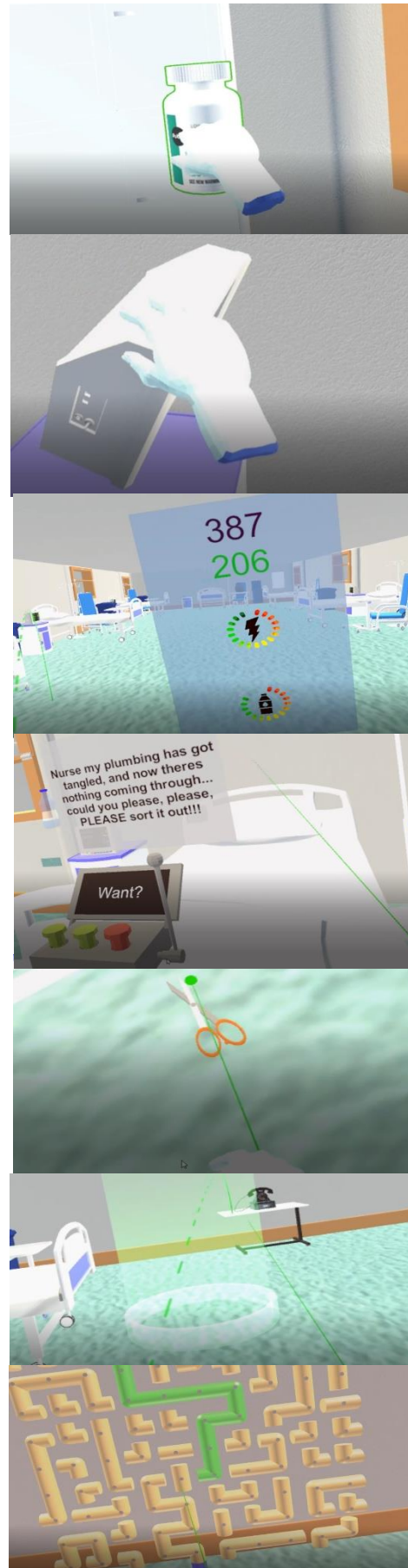
Game Task Three Implementation

Unlike the other two tasks, task three (Pipe Mini Game) has many more moving parts to its operation and is spread across four classes.

The major class here, is the Pipe Manager Class. Which once created, will generate a grid and populate each space within it, with a random pipe piece, until the grid has been completed.

Each piece of pipe has two classes attached. The first, is the Pipe class, this defines what type of pipe piece it is, its position within the grid, colour state and whether it is connected to a positive piece. This class, also handles the rotation of the pipe pieces. Using a coroutine, it adds a slight delay to prevent the objects rotating too quickly, this also makes the mini game more challenging.

The second class, is the Connector class. This handles the collision detection with other pipe pieces and checks the colliding objects connectivity states, reporting back to the Pipe class if successfully connected.



Conclusion & Future Improvements

The team worked well together, there was a clear structure, and everyone had a part to play within the development of this project. Starting with the research that fed into the User Experience decisions, this influenced the design, which provided a solid structure for the implementation. The process passed through a number of iterations, highlighting the value for being flexible to changes, which we experienced first-hand when the overall game structure changed quite significantly. Fortunately, the major changes were during the early stages of implementation.

The investment in the VR Headset, ultimately provided an invaluable asset to the development of this project, allowing me to address VR related issues as soon as they arose. I could not imagine trying to develop a VR project without this hardware.

Future Improvements

- Distraction events, to potentially lead the player away from the task at hand.
- Simulated nurse workload with additional tasks, would provide a systematic way to delegate tasks.
- Animated character to provide the tutorial, currently this is a text canvas which acts as a placeholder / barrier, until the brief has been given.
- The Spawn Controller, could be improved further, to make this more a plugin, to easily allow for additional objects to be implemented quickly.
- Additional checks in the Pipe Mini Game, to prevent impossible games. Also, the pipe rotation would benefit from some animation.

References

Asset References

Visual Assets

Crea3d (2012) *different rubbish*. Available at: <https://www.turbosquid.com/3d-models/free-different-rubbish-3d-model/655562> (Accessed: 10 January 2018)

ProAssets (2016) *Free HDR Sky*. Available at: <https://www.assetstore.unity3d.com/en/#!/content/61217> (Accessed: 18 December 2017)

Rokay3D (2017) *Old Telephone*. Available at: <https://www.assetstore.unity3d.com/en/#!/content/62434> (Accessed: 12 January 2018)

Audio Assets

1pjladd2 (2012) *Eating_an_Apple.wav*. Available at: <https://freesound.org/people/1pjladd2/sounds/143117/> (Accessed: 14 January 2018)

deleted_user_7146007 (2017) *Pills in a Bottle Shaking*. Available at: https://freesound.org/people/deleted_user_7146007/sounds/383873/ (Accessed: 14 January 2018)

f4ngy (2014) *Zap.wav*. Available at: <https://freesound.org/people/f4ngy/sounds/232885/> (Accessed: 14 January 2018)

Gabovitch, I. (2013) *Throwing Away Plastic Trashbag*. Available at: <https://freesound.org/people/qubodup/sounds/192064/> (Accessed: 14 January 2018)

Gladkiy (2016) *Suburban hospital ambience*. Available at: <https://freesound.org/people/gladkiy/sounds/348109/> (Accessed: 02 December 2017)

Hypocore (2012) *buzzer2.wav*. Available at: <https://freesound.org/people/hypocore/sounds/164089/> (Accessed: 02 December 2017)

OtisJames (2014) *thud.wav*. Available at: <https://freesound.org/people/OtisJames/sounds/215162/> (Accessed: 14 January 2018)

swordofkings128 (2017) *Gulping*. Available at: <https://freesound.org/people/swordofkings128/sounds/397611/> (Accessed: 14 January 2018)

Werra (2009) *telephone.mp3*. Available at: <https://freesound.org/people/Werra/sounds/78565/> (Accessed: 12 January 2018)

Implementation References

Microsoft (2017) *Developer readiness status - Immersive headset development*. Available at: https://developer.microsoft.com/en-us/windows/mixed-reality/developer_readiness_status (Accessed: 12 November 2017)

Microsoft (2017) *MixedRealityToolkit-Unity*. Available at: <https://github.com/Microsoft/MixedRealityToolkit-Unity> (Accessed: 12 November 2017)

Unity3D (2018) *Unity User Manual (2017.3)*. Available at: <https://docs.unity3d.com/Manual/index.html> (Accessed: 26 October 2017)

Unity3D (2017) *Events: Creating a simple messaging system*. Available at: <https://unity3d.com/learn/tutorials/topics/scripting/events-creating-simple-messaging-system> (Accessed: 26 November 2017).

Unity Forums (2015) *[Messaging System] Passing parameters with the event*. Available at: <https://forum.unity.com/threads/messaging-system-passing-parameters-with-the-event.331284/> (Accessed: 26 November 2017).

Valve (2017) *SteamVR*. Available at: <https://developer.valvesoftware.com/wiki/SteamVR> (Accessed: 14 November 2017)












VRTK - Virtual Reality Toolkit (2017) *VRTK - Virtual Reality Toolkit*. Available at: <https://vrtoolkit.readme.io/> (Accessed: 23 December 2017)

Appendix One – GitHub Commit History


Commits on Jan 15, 2018

	Clean Up + Fixed Null Ref Exception in build version LordGee committed 15 hours ago
	THATS ALL FOLKS - Development has finished! ... LordGee committed 16 hours ago <ul style="list-style-type: none"> + Added additional audio voices, to make the game more vocal throughout. + Reset Starting instructions to now play from the beginiing + Added additional tutorial instruction beyond the brief + Added voice instructions, pressure vocals to irritate the player and a win clip to complete the task 3 audio set
	High Score Table ... LordGee committed 21 hours ago <ul style="list-style-type: none"> + Added a working high score leaderboard table + Added end game scenarios + Shorter wait for turning pipes + Added display the most recent score attempt
	Player Preferences ... LordGee committed 22 hours ago <ul style="list-style-type: none"> + Added player preferences store and manage the scores and player names + Added load Level 00 on game over
	Fixed further issues with task 3 ... LordGee committed 23 hours ago <ul style="list-style-type: none"> + Fixed issue with teleporting player to task, issue arose by player being teleported out of ward and falling to death + Updated pipe interation with VRTK use interaction, removing null referencee exceptions when testing collider + Score is now calculated to the nearest interger that is greater then the float + Further improvements made to the task acceptor
	Improved Task Acceptor machine interations LordGee committed a day ago
	Fixed issue spawning wrong object ... LordGee committed a day ago <ul style="list-style-type: none"> + Hazard spawner was spawning a rouge water bottle
	Lighting + Button Interaction ... LordGee committed a day ago <ul style="list-style-type: none"> + Addressed lighting issues when loading into the game scene (3x new directional lights added) + Main game now loads from the home screen + Added extra condition to the spawn controller
	Added Play Button ... LordGee committed a day ago <ul style="list-style-type: none"> + Added play button to be selected to start the game + Adjusted Controller scripts to reflect a more appropriate for this scene + Removed all teleportation
	Home Environment Added ... LordGee committed a day ago <ul style="list-style-type: none"> + Added home environment supplied by tommy + Added Camera setup and corrected rotation to match + Added a Home Controller to manage any variables within this scene

Commits on Jan 14, 2018

	Hazard Awareness Task ... LordGee committed 2 days ago + Implemented for the second time the hazard areness task + Objects are zappable via the trigger + Added spawn point for main task objects, now only spawn in the desired location
	Added scripts for trigger interaction LordGee committed 2 days ago
	New Ambient Background Asset Added LordGee committed 2 days ago
	Fixed Revert Error LordGee committed 2 days ago
	Preparing Hazard Objects ... LordGee committed 2 days ago + Resizing and configuring interaction
	Task 2 SFX Implemented LordGee committed 2 days ago
	SFX for Task 3 Implemented ... LordGee committed 2 days ago + Modified how the connection of two pipes were updated
	Collectable SFX working LordGee committed 2 days ago
	SFX assets added ... LordGee committed 2 days ago Apple Eating - ref: https://freesound.org/people/1pjladd2/sounds/143117/ Water Drinking - ref: https://freesound.org/people/swordofkings128/sounds/397611/ Pipe Turn - ref: https://freesound.org/people/f4ngy/sounds/232885/ Pills - ref: https://freesound.org/people/deleted_user_7146007/sounds/383873/ Thud - Ref: https://freesound.org/people/OtisJames/sounds/215162/ Rubbish - Ref: https://freesound.org/people/qubodup/sounds/192064/ Copyright 2013 Iwan Gabovitch, CC-BY3 license.
	Minor update ... LordGee committed 2 days ago + Got rid of the doctor, before he gets arrested for being naked?
	Added doctor asset ... LordGee committed 2 days ago + Won't be used


Commits on Jan 13, 2018

- 

Objects completed out of 3 ...


LordGee committed 3 days ago

 - + Added test mesh to display how many objects have been correctly put away

TODO: May drop this down a bit lower tomrrow, also change the colour
- 


Fully wired up Task 2 ...

LordGee committed 3 days ago


 - + Fixed issue with the failed call back schenario
 - + Added call back option to complete task
 - + Added event trigger for winning the task
 - + Added countdown clock
 - + Improved update functionality
- 

Improved Answering the Telephone ...

LordGee committed 3 days ago


 - + Uses more appropriate collision detection only when neccessary
 - + Fixed issue where phone would not continuously ring
- 

Minor Update

LordGee committed 3 days ago
- 

Further imporvements to the telephone functions ...

LordGee committed 3 days ago


 - + Added checks for isPlaying (Recursive)
 - + Added checks if phone cas already been answered to prevent a second answer
 - + Added ringer so it can be used again later.
 - + Made copy of the bed Task gameobject so it can be destroyed easier mid-game on exit
- 

Who you gonna call? ...

LordGee committed 3 days ago


 - + Phone ringing and instructions communicated

Commits on Jan 12, 2018


- 

Telephone + Audio Assets + Interactivity ...

LordGee committed 4 days ago


 - + Added telephone asset, as placeholder for answering the telephone
REF: <http://u3d.as/t9Q>
 - + Added an audio ring sfx for the phone asset
REF: <https://freesound.org/people/werra/sounds/78565/>
- 

Got Scared, Reverted back

LordGee committed 4 days ago
- 

Pre SteamVR Plugin Update ...

LordGee committed 4 days ago

Going to attempt an update to SteamVR v1.2.3, there may be a conflick between the new version and VRTK. Hense the commit.
- 

Fixed issue of Task 3 loading twice

LordGee committed 4 days ago

Commits on Jan 11, 2018



Scoring and Index Management ...

LordGee committed 5 days ago

- + Ensured consistant approach to updating the game score, now only method is used outside of the game controller.
- + Refasctored the task index functionality, to ensure the correct task is opened at the right point.
- + Fixed task 3 not ending correctly
- + Defined more accuratley when main task objects are spawned.

Commits on Jan 10, 2018



Additional clean elements to task 2 ...

LordGee committed 6 days ago

- + Bottles and Rubbish can now be placed in the bin
- + Three clean up tasks available
- + Task now self instantiates

TODO: imporvements needed in task controller. NEXT JOB



Refactored Test for Win Condition ...

LordGee committed 6 days ago

- + Moved out of an Update check, this will contribute to improving overall performance of the application.




Grab and Place objects ...

LordGee committed 6 days ago

- + Added the ability to grab specific objects
- + when placed on target location object is repositioned to satisfy placement
- + Once repositioned the object is made to be no longer grabbable (is that a word!)
- + Added booling results array to the Task2 script
- + When object placed in correct destination the results array is updated
- + Tests for win conditions is tested every frame, this should refactored to only on each object success, even though this is a small loop it will still be a performance saving no matter how small. TODO: do this refactor sooner rather then later.


Commits on Jan 10, 2018



NullReferenceException : OnTriggerStay [FIXED] ...

LordGee committed 6 days ago


- + Took a while but finally got rid of this error, by using a built in VRTK event, which provided the same result without the need for checking a collision



Rubbish Assets ...

LordGee committed 6 days ago


- + Added rubbish assets
- REF : <https://www.turbosquid.com/3d-models/free-different-rubbish-3d-model/655562>
- + Prepared bin for disposal (Still need to add collider plus script)
- + Fixed issue where objects would randomly spawn at the origin location (e.g. on the floor)



Collectable Fix ...

LordGee committed 6 days ago

- + Fixed issue where collectable items were being grabbed




Simulator + Grab Functions + New Assets ...

LordGee committed 6 days ago


- + Added VRTK Simulator for testing without VR
- + Added and fixed ability to grab objects and teleport around with them
- + Added Medicine Cabinet and Asprin Bottle (Zane Provided)

Commits on Jan 8, 2018




Setting up a novel Task2 layout

LordGee committed 8 days ago



Minor Update

LordGee committed 8 days ago




Correct Timings and Refactored Task 1 ...

LordGee committed 8 days ago

- + Ensures there is a sufficient break between task 1 object spawns.
- + Refactored Task1 scripts
- + If overall game timer is less than 60 seconds then spawn rate ignores the delay
- + Added a 10 second delay to all other objects that are being respawned


Commits on Jan 7, 2018



Minor update ...

LordGee committed 9 days ago


- + Moving to another computer



Asprin collectable ...

LordGee committed 9 days ago

- + Added collider, rigidbody and linked script to make this object collectable




Tutorial extended ...

LordGee committed 9 days ago


- + Added teleportation point to first main object

Commits on Jan 6, 2018


Play Audio Clips while Presenting Instructions ...


LordGee committed 10 days ago

- + Starting instructions are now presented vocally and visually


Voice Overs ...

LordGee committed 10 days ago


- + Added Voice overs supplied by Michaela
- + Added Voice overs supplied by Jess
- + Removed unused game object


Instantiate Main Task + Task Controller ...


LordGee committed 10 days ago

- + Creates starting main task.
- + Provides initial instruction canvas
- + Improvements made to how tasks are managed in the task controller

Commits on Jan 5, 2018


Re-organised Prefabs + Create skeleton of Task One


LordGee committed 11 days ago


Spawn Controller Refactored + Task manager incorporating additional t... ...

LordGee committed 11 days ago

...asks


- + Refactored spawn controller
- + Added additional attributes to incorporate task ones objects that need to be found
- + Added pill bottle. REF : <https://www.turbosquid.com/FullPreview/Index.cfm/ID/409837>


Improved Spawn Management ...

LordGee committed 11 days ago

- + Added additional spawn points
- + Ensures only one object can spawn at any one location at a time

Commits on Jan 4, 2018


Game timer improvements ...

LordGee committed 12 days ago

- + Fixed unused variables
- + Added game timer to the pipe game
- + Changes colour depending on time remaining
- + Now destroys task machine after accepting, ready for reuse
- + Fixed issue with rotating pipes
- + Refactored Bed Buzzer script

Commits on Jan 3, 2018



Food and Drink Consumable ...

LordGee committed 13 days ago

- + Instantiate Food Drink Objects
- + Consuming them via gripping the object replenishes the negative affects
- + Adds points to the score, as incentive to consume.
- + Respawns elsewhere after consumption.



Deminished Vision ...

LordGee committed 13 days ago

- + Based on how hungry / Thirsty the player is the vision is deminished up to a maximum alpha of 0.8f using RGB (0.2f, 0f, 0f) as the overlay.
- + The affect water and food requirement independantly, so if one is replenished the other may still impose an alpha value.
- + Removed player status updates from the game controller



Pipe game now working again ...

LordGee committed 13 days ago

- + Repositioned and resized to fit environment

Commits on Jan 2, 2018



Re-instatiating Pipe Game ...

LordGee committed 14 days ago

- + Current issue with start rotation
- + Resize objects to fit into new environment



Teleport Point Rotation + Task Acceptor Improvements ...

LordGee committed 14 days ago

- + When player teleports to task start location rotation of player is corrected to face the task accepting machine, also within easy reach
- + Added task acceptor functionality back in, interactable objects now provided an outline highlight when hovered over.
- + Added a grip button test to the Player Controller

Commits on Jan 1, 2018



Minor change

LordGee committed 15 days ago



Player Controllers ...

LordGee committed 15 days ago

- + Back in action
- + Identify and action relevant functions for controller inputs



Status canvas working again ...

LordGee committed 15 days ago

- + Due to VRTK Controllers being deactive at begin, the canvas manager could not initilise the canvas correctly. Resolved by making the variable public and specifically defining the game object.
- + Improvements to be made to the way the button pressed is acknowledged, this will be required later anyway, so may as well do it as the next task.



Task Presentation ...

LordGee committed 15 days ago

- + Improved task instruction
- + Added rotation enforcement on telepoint points
- + Implementing Task Acceptor
- + Added bed buzzed back in
- + Removed canvas rotation

Commits on Dec 29, 2017



Improve teleportation usability ...

LordGee committed 18 days ago

- Added snap to destination points for attending tasks
- Added improved trigger for acknowledging new tasks
- Added body physics
- Refactored Task 3 Scripts
- Reviewed order in which events are listened to within a task
- Stopped listening as soon as no longer needed

Commits on Dec 27, 2017



One step forward, Two steps back ...

LordGee committed 20 days ago

After a song and dance, back to where I was 7 days ago



Revert Merge

LordGee committed 20 days ago



Removedd files prep merge

LordGee committed 20 days ago



Merge Revert

LordGee committed 20 days ago

Commits on Dec 21, 2017



OPen and Close Status ...

LordGee committed 26 days ago

+ Attached to controller

Commits on Dec 21, 2017



Hands Assets Added

LordGee committed 26 days ago

Commits on Dec 20, 2017



Canvas + Bug Fixing ...

LordGee committed 27 days ago

+ Status canvas working, needs to be smaller and positioned to controller
+ Fixed issues that were not linked from the move to the new project.

TODO:

- Add triggers to start mission
- Add canvas to controller which opens on menu button press
- Spawn mission acceptor at the correct bed



Unity 2017.3.0f3 Update ...

LordGee committed 27 days ago

+ Upgraded project to new version



Status Canvas Display + Bottle & Water assets added ...

LordGee committed 27 days ago

+ Added status canvas {Score, Game Time, Hydration, Energy and Workload}
+ Added canvas controller to modify the status canvas components
+ Updating variables through the Game Controller class
+ Added Water bottle and Apple to the 3d Assets (Duygu) wired into the food and drink controller for later use.
+ Modified hydration and energy starting levels to represent PNG availblity.

Commits on Dec 19, 2017

**Added 2D Assets** ...

LordGee committed 28 days ago

- + Hydration pngs (Michaela)
- + Energy pngs (Michaela)
- + VR Nursing logo (Duygu)
- + Converted to sprites and resized

TODO: Implement player status menu

**Added bed area assets** ...

LordGee committed 28 days ago

- + Added Bin
 - + Added monitor
 - + Added IV Stand
 - + Added chair
 - + Added food table
- + Also added partition walls to make the rectangle
+ Adding ceiling and spotlight for the room lighting

TODO:

- Zane to complete cabinet model due to missing panel
- Add status canvas
- Rewire code managers

Commits on Dec 18, 2017

**Added Beds** ...

LordGee committed 29 days ago

- + Imported 6x beds, positioned and resized

**Constructing room** ...


LordGee committed 29 days ago


- + Added room assets (Tommy)
- + Added skybox


REFERENCE

Skybox: <http://u3d.as/sdN>

Commits on Dec 17, 2017


- 


Setting up managers ...
 LordGee committed on 17 Dec 2017
 + Added managers {game, event, needs}
 + Camera Rig
 + Created floor
- 

Importing Previous Scripts ...
 LordGee committed on 17 Dec 2017
 + Imported important files from previously tested project including:
 + Game Managers
 + Task Manager + Task 3
 + All scripts created in test scene
- 

Imported SteamVR ...
 LordGee committed on 17 Dec 2017
 + Added fresh copy of SteamVR

Commits on Dec 15, 2017

- 

New Project Created ...
 LordGee committed on 15 Dec 2017
 + Fresh Start
- 

Initial commit
 LordGee committed on 15 Dec 2017

Verified

Commits on Dec 15, 2017

- 

New Asset Test
 LordGee committed on 15 Dec 2017

Commits on Dec 11, 2017

- 

Added the task acceptance machine
 LordGee committed on 11 Dec 2017

Commits on Dec 5, 2017

- 

Fixed Previous Issues
 LordGee committed on 5 Dec 2017
- 

Dodgy Push ...
 LordGee committed on 5 Dec 2017
 Crash hopefully fixed by next iteration
- 

Added an end game effect ...
 LordGee committed on 5 Dec 2017
 All none used objects fall from the wall

Commits on Dec 2, 2017



Complete loop ...

LordGee committed on 2 Dec 2017

One task logic down



Building UI Interaction

LordGee committed on 2 Dec 2017



Audio assets + Bed light flashing + buzzer ...

LordGee committed on 2 Dec 2017

Audio assets Added

Ambience : <https://freesound.org/people/gladkiy/sounds/348109/>

Buzzer : <https://freesound.org/people/hypocore/sounds/164089/>

Commits on Dec 1, 2017



Additional Assets

LordGee committed on 1 Dec 2017

Commits on Nov 30, 2017



Pipe Mini Game

LordGee committed on 30 Nov 2017

Commits on Nov 29, 2017



Food / Drink Spawner

LordGee committed on 29 Nov 2017

Commits on Nov 28, 2017



Minor changes to new assets

LordGee committed on 28 Nov 2017

Commits on Nov 22, 2017



New assets added ...

LordGee committed on 22 Nov 2017

+ Man

+ Filing Cabinet

+ Bed side table

Commits on Nov 21, 2017



Scriptable Objects

LordGee committed on 21 Nov 2017



Pointers and Teleporting

LordGee committed on 21 Nov 2017

Commits on Nov 14, 2017



SteamVR + Game Controller ...

LordGee committed on 14 Nov 2017

- + Added the SteamVR Plugin into the project and testing how it operates
- + Added Game Controller script to manage the player and game states

Commits on Nov 12, 2017



MRTK Testing

LordGee committed on 12 Nov 2017

Commits on Nov 7, 2017



base interaction ...

Jiexiang Shen committed on 7 Nov 2017

Achieved the base function we discussion

Commits on Nov 6, 2017



Playground scene for testing

LordGee committed on 6 Nov 2017

Commits on Oct 26, 2017



Initial Commit

LordGee committed on 26 Oct 2017



Initial commit

LordGee committed on 26 Oct 2017

Appendix Two – Game Controller Main Components

```
private void Start()
{
    currentGameState = ConstantController.GAME_STATE.Brief;
    hydrationTimer = hungerTimer = 0f;
    CanvasController.gameHydration = hydrationLevel =
        ConstantController.HYDRATION_MAX;
    CanvasController.gameEnergy = hungerLevel = ConstantController.HUNGER_MAX;
    EventController.TriggerEvent(ConstantController.EV_SPAWN_FOOD);
    EventController.TriggerEvent(ConstantController.EV_SPAWN_WATER);
    EventController.TriggerEvent(ConstantController.EV_SPAWN_HAZARD);
    gameScore = 0f;
    winCondition = true;
    gameTimer = timerInterval = ConstantController.GAME_TIME;
    UpdateGameScore(gameScore);
    UpdateGameTimer();
    VRTK_BasicTeleport.BlinkColourColor = Color.clear;
}

private void Update()
{
    gameTimer -= Time.deltaTime;
    if ( Mathf.Floor(timerInterval) - Mathf.Floor(gameTimer) >= 1f ) {
        timerInterval = gameTimer;
        UpdateGameTimer();
    }
    switch ( currentGameState ) {
        case ConstantController.GAME_STATE.Brief:
            UpdateBrief();
            break;
        case ConstantController.GAME_STATE.Playing:
            UpdatePlaying();
            break;
        case ConstantController.GAME_STATE.GameOver:
            UpdateGameOver();
            break;
        default:
            Debug.LogError("No active game state!");
            break;
    }
}
```



```
private void UpdatePlaying()
{
    if ( Time.timeSinceLevelLoad < ConstantController.GAME_TIME ) {
        if ( hydrationLevel > 0 ) {
            hydrationTimer += Time.deltaTime;
            if ( hydrationTimer > ConstantController.HYDRATION_DECREASE_TIME ) {
                hydrationLevel -= 1f;
                AdjustFadeColor();
                CanvasController.gameHydration = hydrationLevel;
                hydrationTimer = 0f;
            }
        } else {
            winCondition = false;
            GetComponent<AudioSource>().clip = endGameStates[0];
            GetComponent<AudioSource>().Play();
            currentGameState = ConstantController.GAME_STATE.GameOver;
        }
        if ( hungerLevel > 0 ) {
            hungerTimer += Time.deltaTime;
            if ( hungerTimer > ConstantController.HUNGER_DECREASE_TIME ) {
                hungerLevel -= 1f;
                AdjustFadeColor();
                CanvasController.gameEnergy = hungerLevel;
                hungerTimer = 0f;
            }
        } else {
            winCondition = false;
            GetComponent<AudioSource>().clip = endGameStates[1];
            GetComponent<AudioSource>().Play();
            currentGameState = ConstantController.GAME_STATE.GameOver;
        }
    } else {
        winCondition = false;
        currentGameState = ConstantController.GAME_STATE.GameOver;
    }
}

public void SpawnHazard()
{
    StartCoroutine(DelayNewSpawn(ConstantController.EV_SPAWN_HAZARD, 45f));
}

private IEnumerator DelayNewSpawn(string _spawn, float _delay)
{
    yield return new WaitForSeconds(_delay);
    EventController.TriggerEvent(_spawn);
}
```

```

private void AdjustFadeColor()
{
    float increment = maxAlpha / (ConstantController.HUNGER_MAX +
        ConstantController.HYDRATION_MAX);
    alpha = 0f;
    alpha += (ConstantController.HUNGER_MAX - hungerLevel) * increment;
    alpha += (ConstantController.HYDRATION_MAX - hydrationLevel) * increment;
    if ( alpha > maxAlpha ) {
        alpha = maxAlpha;
    }
    VRTK_BasicTeleport.BlinkColourColor = new Color(0.2f, 0f, 0f, alpha);
    VRTK_SDK_Bridge.HeadsetFade(VRTK_BasicTeleport.BlinkColourColor, 0.1f);
}

public void UpdateGameScore(float _score)
{
    gameScore += Mathf.Ceil(_score);
    CanvasController.gameScore = gameScore;
}

public void UpdateGameTimer()
{
    CanvasController.gameTimer = gameTimer;
    EventController.TriggerEvent(ConstantController.EV_UPDATE_STATUS_CANVAS);
}

public float GetGameTimer()
{
    return gameTimer;
}

private void UpdateGameOver()
{
    if (!once) {
        if ( winCondition ) {
            GetComponent().clip = endGameStates[2];
            GetComponent().Play();
            gameScore += Mathf.Ceil(gameTimer);
        }
        once = !once;
        if (GetComponent().isPlaying)
            finishTimer = GetComponent().clip.length;
        else
            finishTimer = 1f;
        StartCoroutine(LoadMainMenu(finishTimer));
    }
}

private IEnumerator LoadMainMenu(float _length)
{
    yield return new WaitForSeconds(_length);
    FindObjectOfType<PlayerPrefsController>().SetPlayerScore(gameScore);
    SceneManager.LoadScene(LEVEL_00);
}

```

Appendix Three – Setting and Getting High Scores

Player Preferences Controller

```

public void SetPlayerScore(float _value)
{
    PlayerPrefs.SetFloat(CURRENT_SCORE, _value);
    AddNewScoreToHighScore(_value);
}

private void AddNewScoreToHighScore(float _testScore)
{
    bool isAvailableSpace = false;
    for ( int i = 0; i < NO_OF_SCORES; i++ ) {
        if ( !PlayerPrefs.HasKey(HIGH_SCORE_ARRAY + (i + 1)) ) {
            isAvailableSpace = true;
            PlayerPrefs.SetFloat(HIGH_SCORE_ARRAY + (i + 1), _testScore);
            break;
        }
    }
    if (!isAvailableSpace) {
        int lowestIndex = -1;
        float lowestScore = 0;
        for ( int i = 0; i < NO_OF_SCORES; i++ ) {
            float thisTest = PlayerPrefs.GetFloat(HIGH_SCORE_ARRAY + (i + 1));
            if ( thisTest < _testScore ) {
                if (thisTest < lowestScore || lowestScore == 0f) {
                    lowestScore = thisTest;
                    lowestIndex = (i + 1);
                }
            }
        }
        if (lowestIndex != -1) {
            PlayerPrefs.SetFloat(HIGH_SCORE_ARRAY + lowestIndex, _testScore);
        }
    }
}

public List<float> GetHighScores()
{
    List<float> scores = new List<float>();
    for (int i = 0; i < NO_OF_SCORES; i++) {
        if (PlayerPrefs.HasKey(HIGH_SCORE_ARRAY + (i + 1))) {
            scores.Add(PlayerPrefs.GetFloat(HIGH_SCORE_ARRAY + (i + 1)));
        }
    }
    scores.Sort();
    scores.Reverse();
    return scores;
}

```

Home Controller

```
private void DisplayHighScores()
{
    GameObject text = GameObject.Find("HighScoresText");
    List<float> scores = FindObjectOfType<PlayerPrefsController>().GetHighScores();
    for (int i = 0; i < scores.Count; i++) {
        text.GetComponent<TextMesh>().text += "\n" + (i + 1) + ". " + scores[i];
    }
}
```