Project Handover Document

Escape Room Games – The Island
Deakin Cloud Ventures

Trimester 1, 2021

1. Project Information

1.1. Client/Product Owner

Andrew Ferguson
Director - Escape Room Games Pty Ltd
info@escaperoomgames.com.au

1.2. Academic Mentor/Supervisor

Yong Xiang
Professor of Network Engineering
yong.xiang@deakin.edu.au

1.3. Project Team

Escape Room Games – The Island
Deakin Cloud Ventures

Student ID	Student Name	Role
217265358	James Papas	Team Leader, Project Management, Design, Development
213389305	Levi Peters	Team Leader, Project Management, Programming, Development
218015247	Sean Corcoran	Programming and Development
219364312	Marion Perera	Design
214358992	Anthony Antoniou	Programming and Development
219195844	Jack Oberstar	Design
216091896	Parisa Nafis	Programming and Development
219326261	Andrew Samothrakitis	Programming and Development
219397329	Arsha Anoop	Design
220415726	Naadhish Yahya	Design
219559949	Jonathan Vieri	Coding Programming / Audio Production

2. Project Overview

Throughout the years the world has grown to know and love tabletop gaming. Now that technology is continuing to evolve, boardgames have been refined to now incorporate digital overlays within a real-world environment. Now instead of placing down tangible boards and pieces, developers have introduced digital User Interfaces, digital boards and digital assets to create an even more immersive gameplay experience. These digital overlays are called Augmented Reality. Augmented Reality is so incredible and hyped that 3.5 billion people are expected to use AR mobile applications by 2022 (Digi-Capital, 2020). That is approximately 44% of the world's population. Our client is looking to develop an

Augmented reality game called ‘The Island’, which aims to create a truly immersive escape room experience situated around a digital island. This island can be placed down so the user can navigate around to collect resource and build a raft. The raft can then be used to get off the island.

Aims of the project:

- Our main goal is to deliver an augmented reality boardgame called “The Island” for the Android platform that meets the expectations of our product owner.
- Have a fully functional game that is available for download on the Google Play store. Game to have a refined User Interface, Performance optimisation and minimal bugs.
- The game will centre around a virtual deserted island where players must collect resources to build a raft and escape the island. The player must defend themselves against rats, snakes, and birds whilst also collecting resources to eat, drink, and to build a raft to get off the island, all within 15 minutes.
- To maintain a tropical island theme, complete with wooden, palm and sand themes.

High-level Deliverables:

- Re-designed The Island Logo
- Created and linked Wiki Page to Game with ‘How to Play’ Video
- Added immersive noises and soundtracks to the game.
- Added Weather Systems to the game
- Added new 3d animal objects and rigged animations to game.
- Added attacker indicator system.
- Create new death and escape scene cinematics.
- Place Island and marker system.
- Added Credits to game.
- The Final Island Game (.APK)

3. User Manual

3.1. Getting started

3.1.1. Development Environment

Applications Required

Windows

Source Control: GIT installed plus SourceTree or GitHub Desktop

IDE: Visual Studio Community or VS Code

Unity: Unity Hub and then download Unity version 2020.1.1f

Mac

Source Control: GIT installed plus SourceTree or GitHub Desktop

IDE: Visual Studio or JetBrains Rider (free with a student account)

Unity: Unity Hub and then download Unity version 2020.1.1f

Installing Unity Hub and Unity 2020.1.f

<Demo Video Link>

Getting started with the repository

Link to the team's repository is here

<Demo Video Link>

How to open and run the project in Unity

<Demo Video Link>

How to build and run the project on device

<Demo Video Link> <How to build and run on device PDF>

How to open the game's scripts (source code)

<Demo Video Link>

Overall File Structure of the project

<Demo Video Link>

User Journey

<Demo Video Link>

3.1.2. Design Environment

Windows

Source Control: GIT installed plus SourceTree or GitHub Desktop

Unity: Unity Hub and then download Unity version 2020.1.1f

3D Modelling: Blender <Download Link>

Sound design: FL Studio

2D image/icon design: Adobe Photoshop CC

Mac

Source Control: GIT installed plus SourceTree or GitHub Desktop

Unity: Unity Hub and then download Unity version 2020.1.1f

3D Modelling: Blender <Download Link>

Sound design: FL Studio

2D image/icon design: Adobe Photoshop CC

Installing Unity Hub and Unity 2020.1.f

<Demo Video Link>

Getting started with the repository

Link to the team's repository is here

<Demo Video Link>

How to open and run the project in Unity

<Demo Video Link>

How to build and run the project on device

<Demo Video Link> <How to build and run on device PDF>

4. Completed Deliverables

Name	Description	Trello card	Creator(s)	Location	File name
Death scene cinematic	Death scene cinematic that plays when the user has died in-game.	https://trello.com/c/T0B8zsUB	Jack,	Teams – Handover and showcase/T3 2020	DeathScene.mp4
Escape scene cinematic	Escape scene cinematic that plays when the user has completed the game objectives and won.	https://trello.com/c/T0B8zsUB	Jack	Teams – Handover and showcase/T1 2021	EscapeScene.mp4
Menu cinematic	Rotating island that is played behind the main menu.	https://trello.com/c/T0B8zsUB	Jack	Team's Bitbucket Repository	IslandRotation.mp4
Weather Systems	Rainy Weather and Thunderstorms	https://trello.com/c/jlfpy3vQ	Marion, Naadhish	Team's Bitbucket Repository	Rain Animation Thunder.blend
Non-scannable Icons	Used to indicate an inactive image trigger when two of the same image triggers are scanned at the same time.	https://trello.com/c/cKWO0Ptm	Levi, Marion, Naadhish	Team's Bitbucket Repository	NoScanV1.png NonScannable_3D_icon1.blend Upside Down Bucket.blend

3D assets	Created and animated 3d attackers into the game, snake, new rat, and bird.	https://trello.com/c/F5KB9Pi9 https://trello.com/c/PpU04icZ	James	Team's Bitbucket Repository	Playscene.unity Rat.blend Bird.blend Snake.blend
UI in Main menu screen	I animated the UI in the main menu screen to bounce on tap	https://trello.com/c/4tF90X7H	James	Team's Bitbucket Repository	Playscene.unity
Attacker Indicator	Create attacker Indicators to point towards attacker proximity	https://trello.com/c/mzMXDSlu	James	Team's Bitbucket Repository	Snake.png Rat.png Bird.png
Attacker Prompts	Redid the previous attacker prompts that display when an attacker enters the game.	https://trello.com/c/bSKBFK4A	James	Team's Bitbucket Repository	RatAttackWarning.png SnakeAttackWarning.png BirdAttackWarning.png Playscene.unity
Skip Button for cinematics	Configured and designed skip button for cinematics to quickly return to game main menu.	https://trello.com/c/USuLMsIP	James	Team's Bitbucket Repository	Skip.png EscapeScene.unity DeathScene.unity
Playscene Settings Overlay	Added in Settings overlay on game setup screen and pause panel.	https://trello.com/c/zFNSsrav	James, Sean	Team's Bitbucket Repository	Playscene.unity
Graphic for pre game marker	Created graphic for pre game marker to be placed prior island is placed	https://trello.com/c/6BtAOJWz	James	Team's Bitbucket Repository	Playscene.unity NewMapPlaceholder.png
Designed Credit Panel	Created and implemented credit panel into game	https://trello.com/c/5QglJkmX	James, Sean	Team's Bitbucket Repository	Playscene.unity

Credits Script	Script that implements the features need to control the credits panel.	https://trello.com/c/5QglJkmX/337-add-credits-to-game-scene	Sean	Team's Bitbucket Repository	CreditsScript.cs
Get Camera Position Script, Tracking of active attackers on screen.	Script used to track the position of the player's camera in relation to the attacking animals.	https://trello.com/c/bPY9TKJC/281-add-3d-attacks-with-pointer-where-they-are-coming-from	Sean	Team's Bitbucket Repository	GetCameraPos.cs
Development Testing Controls	Created keyboard shortcuts to trigger different events to speed up testing of the game during development.	https://trello.com/c/FFdQHRVe/271-create-dev-testing-controls	Parisa, Andrew, Sean, Levi	Team's Bitbucket Repository	GameSceneScript.cs IslandPlacement.cs AutoQuality.cs
Damage Indicators	Hit makers are shown on screen at the location of an attacker when the player taps on attacker.	https://trello.com/c/mzi1mx63/276-visual-feedback-when-you-hit-an-attacker	Sean	Team's Bitbucket Repository	DestroyObject.cs
Tracking of active scanning trigger objects	Upon scanning a trigger object to add resources player is shown on screen scanning indicator. Indicator is removed when finished or the	https://trello.com/c/xpxEDyX8/282-progress-bar-to-show-scanning	Andrew, Sean	Team's Bitbucket Repository	GamesSceneScript.cs

	resource moves out of view.				
Adjustment of Ground Plane Icon.	Icon that shows upon Vuforia's ground detection system was enlarged for better readability.	https://trello.com/c/ThGOmkH1/273-placement-icon-bigger-if-possible	Parisa	Team's Bitbucket Repository	PlaneIndicator.prefab PlayScene.unity
Placement Marker System	Map Marker is placed upon plane detection. Key marker names follow AR camera as player places markers around map. Upon Island placement Map Marker is replaced by and Island model.	https://trello.com/c/6BtAOJWz/223-this-is-important-high-priority-pre-game-card-markers-on-island-functionality	Parisa	Team's Bitbucket Repository	MapMarkerQuad.prefab GamesSceneScript.cs BillboardToCamera.cs Game_Marker.mat IslandPlacement.cs Game_Marker.png btnPlaceMarker.prefab PlayScene.unity
Quit state disables other inputs.	Disables specific button inputs when application is quitting.	https://trello.com/c/sC2nVMfw/320-disable-touch-input-when-quitting	Parisa	Team's Bitbucket Repository	GameSceneScript.cs ButtonAnimation.cs Menu.cs PlayScene.unity Main_Animation.unity
Auto Scale Quality	When the player hits Auto Quality setting mode the application adjusts the game quality if performance issues are detected.	https://trello.com/c/JeRCcmrJ/270-auto-scale-quality	Parisa	Team's Bitbucket Repository	Menu.cs SettingsMenu.cs AuotQuality.cs FPSScript.cs NotificationLerp.cs PlayScene.unity

	Auto Quality script then makes the appropriate visual adjustments and notification animates to notify user of quality setting change.				Main_Animation.unity NotificationBackground.png
Adjusting Terrain Quality to New Settings Menu	Switches terrain quality whenever a setting related to it is changed in any of the new Settings panels in the application.	https://trello.com/c/SGzdbxsO/343-adapt-terrain-placement-to-new-setting-configuration	Parisa	Team's Bitbucket Repository	TerrainSwitch.cs AutoQuality.cs
Universal Transfer of Settings Menu Adjustment.	Whenever the player changes a setting in any of the new settings menu overlay it will save that value and adjust other settings menu overlay values to it once they are open in the application.	https://trello.com/c/uaIE22lA/335-add-setting-configs-to-buttons-in-the-playscene-settings-overlay https://trello.com/c/6PKwedsT/341-setup-event-triggerscomponents-and-game-objects-for-new-in-game-	Parisa	Team's Bitbucket Repository	IslandPlacement.cs Playscene.unity Main_animation.unity SettingsValues.prefab SettingsMenu.cs Menu.cs SettingsTransfer.cs

		settings-menu			
Place Island Button Reacts to Ground Detection.	Place Island Button now pulses and glows when the application detects the ground at the start. This is to indicate that the button is now interactable and the user can place the map marker. Glow is disabled when ground ceases to be detected.	https://trello.com/c/k2Yc1rAy/322-place-island-button-to-do-something-when-island-ready-to-be-placed	Parisa	Team's Bitbucket Repository	HighlightButton.cs BorderFlash.anim Playscene.unity
Build button user experience enhancements.	Build button is greyed out until the users has all the resource bars filled to the right capacity. The button is then interactable and has colour until the player taps it and resources are depleted once more.	https://trello.com/c/YFaNBNVm/279-grey-out-the-build-button-until-ready	Parisa	Team's Bitbucket Repository	BuildButton.png GameSceneScript.cs Playscene.unity
Script + Game Asset Name Clean-up	Cleaned up scripts in project for readability and smoother functionality.	https://trello.com/c/0btKvv3j/303-check-rescue-card-code-ensure-it-works-with-health-code-updates	Parisa	Team's Bitbucket Repositor	GameSceneScript.cs Playscene.unity SettingsMenu.cs Menu.cs

		https://trello.com/c/6PKwedsT/341-setup-event-triggerscomponents-and-game-objects-for-new-in-game-settings-menu				
In-Game Sound Effects	All the in-game sound effects for interactions and actions inside the game.	https://trello.com/c/gHOgncYC/313-create-sound-effects-and-music	Jonathan	Teams – Handover and Showcase – T1 2021 – Sound Files	BeachWave.mp3 BirdChirp1.wav BirdChirp2.wav BirdDying1.wav BirdDying2.wav ButtonClick1.wav ButtonClick2.wav ButtonClick3.wav ButtonDeny1.wav ChickenSound1.wav CricketNoise.wav DayAmbiance.mp3 Hit1.wav Hit2.wav Hit3.wav Hit4.wav	NightAmbiance.mp3 RainingSound1.wav RatDying1.wav RatDying2.wav Snake1.wav Snake2.wav Snakedying1.wav Snakedying2.wav Splat1.wav Thundersound1.wav Thundersound2.wav Thundersound3.wav
How to Play video	Video on the instruction of how to play the game and how the game works.	https://trello.com/c/5ifguR6w/134-create-	Jonathan - Andrew	Teams – Handover and Showcase – T1 2021 - Videos	The Island – How to play.mp4	

		how-to-play-video			
Day and Night System	In game day and night overlay and status.		Anthony James	Team's BitBucket Repository	Playscene.unity Nightanimation.anim Dayanimation.anim Day Overlay.png Night Overlay.png
How to Play Wiki	A step by step wiki to show users how to play the game	https://trello.com/c/5ifguR6w/134-create-how-to-play-video	Andrew	https://how-to-play-escape-room-game-the-island.fandom.com/wiki/How_to_Play:_Escape_Room_Game_-_The_Island_Wiki	
Health Level of Animals	I had to change the attackers health level	https://trello.com/c/GfLlynx9/287-attackers-have-different-health-amounts-rat-bird	Andrew	Team's BitBucket Repository	GameScene.cs DestroyObject.cs
Add Snake Object	I had to add the snake object to the game.	https://trello.com/c/PpU04icZ/324-import-animals-and-animations-to-unity-replacing-originals-	Andrew	Team's BitBucket Repository	GameScene.cs

		add-snake-into-play-scene			

5. Roadmap

Future Features

- Multiplayer Implementation
- Code and enable two multiplayer modes.
- VS Mode: Have 2 or more players in the same space (or over internet in different locations) place an island and sync the game timer, start the game and each race for the 15 minutes to get off the island first, or to outlive the other player.
- Cooperative Mode: Have 2 or more players in the same space playing the one game, using the same island and sharing Wood resources to build the raft as a team and escape together! Each player has individual Health, Water and Food values, so each player must still survive by collecting resources, but all Wood stockpiles between players.

- Difficulty Settings
- Add difficulty settings to the game.
- This can include rate of resource drain, frequency of animal attacks, number of resources needed to build the raft, amount of time allocated to the game.
- The game can be setup to have a default “normal” difficulty tweaked by the developers as the core/base experience. Easy and Hard difficulties can be implemented to allow the user a better chance to beat the game or give themselves that extra level of challenge.
- Can be implemented as either a difficulty selection, or as individual tweaks for more advanced player options.

- Weather Systems and Cave
- Code in and implement a weather system. The screen overlay for rain and lightning has been developed graphically.
- When it rains, the players Water resource might go up, but there could be a chance of the player being struck by lightning, removing large amounts of health. It could be worth the risk to the player to be in the rain or to seek shelter.
- The Cave: when the rainy weather sets in, the player could interact with the cave (probably in the same vein as the rescue cards) to seek shelter from the rain. There were also thoughts of having the cave swap between active and inactive to randomly contain resources and could also provide protection from animal attacks for a given amount of time.
- Using the cave would give the player a small time of relaxation as they can’t be attacked/hurt and maybe even get a small increase of resources. However, it could also be set so that no other resources can be collected while using the cave, as realistically the player would not be out foraging items while hiding.

- Refined Spawning Mechanics
- Currently the animals all spawn based on the location of the user's phone, and so always spawn in a "predictable" location and the rat/snake spawn up in the air. For realism, they should be spawned at the ground level, and setting up random spawning would make it more fun for the player.

In Development

- Scale Island Size
- After placing the island, currently it only has 1 size. Given that players could have different sized play spaces, work was started to allow the user to scale the size of the island.
- There were a few issues with scaling the island and keeping textures and meshes working properly. The island and meshes were exported and changed to better accommodate this. However, by the end of Iteration 2, there was still some issues getting these changed assets to work properly in the project.
- Furter work needs to be completed to get this fully working, though looking at the Trello, it appears most issues were fixed, but the developers were attempting to also get water added to the island which was causing problems.

- Water added to Island
- There was an attempt to add water to the island, which was possible, but once Scaling of the island was starting to be implemented, it was causing problems.
- The first part of water implementation was adding water in blender and importing in, however this could not be figured out, every time it was ported out to Unity, the water wouldn't come with it properly.
- One of our developers found some water additions that exist within Unity, there was not enough time at the end of Iteration 2 for this to be investigated fully. This appears to be the best bet going forward to get the water added in.

6. Open Issues

There are a few small bugs and issues with the application. However, the stakeholder has said they are happy with where it is to be published.

1. Audio for the settings button is not initially set to the volume slider and the effects the sound slider and checkbox has is delayed.
2. The terrain island is missing sand textures on the corners of each island terrain asset.
3. Rotate button does not work with the island art asset in the unity project as they are terrain assets (cannot rotate or scale terrain assets). If scaling is to be implemented then the islands will need to be fbx, obj or any other 3D mesh format not a terrain asset.
4. If the project was to go through another squad cycle, the blend files/folders should be placed outside the unity project repository as it adds unnecessary load time to the unity project.
5. Mobile shader optimisation if performances issues crop up in the future.

7. Lessons Learned

Modular Programming and Commenting Practices-
The Game Scene script was the largest script in the project and the hardest to navigate and sift through.
Having smaller scripts means the project benefits from the positives modular programming. It means the component functionality of Unity can be utilised easily and code navigation is easier. It also reduces the risk of merge conflicts occurring from multiple developers working on the same script.
Other issues were that some scripts would have specific formulas/variables attached to them that weren't provided more context in the script comments. This made it harder to assess a lines specific importance or how to adapt it to changing project specifications.
Prefabs-
A lot of merge conflicts happened in the primary game scene that could have been avoided /easily amended if most of the scene assets were prefabs stored in the asset's directory. This way component instances could easily be restored while manually set event changes could still be retained in scene hierarchies.
Art pipeline transfer-
Another hurdle was transferring art assets from Blender into a compatible format for Unity to read. Some examples of this can be seen in the initial rain asset, the newly rendered islands and water.
Having a more streamlined art asset pipeline would have been good. Also utilising the Unity asset store would have saved some time. Having a look at Unity's shader pipeline would have helped implement water features into the application.
Another major issue was that the Island art models from the previous trimesters were terrain assets. This means that any changes to the terrain in world space is inherently restrictive and a couple of features for this and previous iterations did not work. Using game mesh assets instead of terrain assets would have been better suited to the nature of this application.
Merging with Unity-
There were multiple merge issues between branches during this trimester. Setting up Unity YAML Merge to work with the project alleviated a significant amount of merge issues. Instructions to set it up can be found here:
https://docs.unity3d.com/Manual/SmartMerge.html

8. Product Development Life Cycle

Within the duration of the trimester, each member had worked proactively in meeting the required processes and practices set out by the project and unit requirements. These processes involved the continuous tracking and assigning of tasks using third party management tools. One of the key team management tools used in this respect, included Trello. Trello boards are organisation tools like Kanban boards that ideal for sprint centred development. This tool was additionally used as a KPI (key performance indicator) tool, so that peers can identify a member's commitment to project deliverables.

Another tool that was used to track the progress and commitment of individuals, included the use of Bitbucket. Bitbucket is a source code repository hosting service which allowed the real-time syncing of documents between users. This was a tool that was effective against conflicts of work between members, and an essential tool to commit new saved work to the project. The commits pushed to the repository was then used to identify what tasks had been recently updated or in progress of being updated.

The processes and practises conducted systematically included, creating tasks in Trello specified by the product owner, assigning story points for prioritisation purposes, assigning team members appropriately to the tasks, committing new work to bitbucket. This process was followed along with ongoing team communication in weekly meetings and measured between 4 iterations. The nature of the sprints gave time to reflect and adapt development techniques to feature and time constraints of the product life cycle. This process was completed up to the end of the trimester.

Weekly meetings were held through a private Microsoft Teams channel and stakeholder meetings took place in a private discord channel. Additionally, it was proper protocol to notify anyone of absence, repository commits and merge conflicts through the Microsoft Teams private squad chat channel.

8.1. New Tasks

With new tasks that were introduced into the project we started by first conducting a team meeting with the product owner and In this meeting each team member would show work completed for review by panel team. Once every team member has showcased their work the product owner would then review and suggest what he wants to be implemented or removed from the project. In this meeting, team members had the freedom to express their own ideas that would help in improving the product. After the meeting, the team leaders would create Trello cards and assign story points to corresponding tasks according to the importance of the task. After these tasks were created and given points, We would have a discussion and divide the tasks between everyone and divide them according to everyone's strengths, team members also had the ability to take ownerships/join task if it intrigued them. This process was completed each time a new iteration was reached, and new tasks were created between each sprint and product owner meeting.

8.2. Definition of Done

Prior and during each iteration, ideas on the project maintenance and new features were discussed and assigned to members. Keeping track of task progress was essential to reach project outcomes before the end of each iteration. To achieve a streamline process of completing tasks, team members participated in the below activities:

1. **Task Completion (Trello):** Comment tasks completion and move task card to ‘Review’ panel.
2. **Communication (Microsoft Teams):** Communicate task changes to team.
3. **Commit Task (BitBucket):** Commit changes to branch.
4. **Task Review (Trello):** Team members review task.

5. **Task Completion (Trello):** Move task card to ‘Completed’ panel.

This process not only managed the flow of the project tasks, but also included all team members to be involved with the new changes being made, as well as avoiding conflicts when working on the same tasks or committing to the BitBucket branch.

8.3. Task Review

Reviewing tasks is an essential process that needs to be actioned to uphold the quality and integrity of the project. Throughout all tasks the review process was actioned. Details of the process commenced when the Trello card for a task has been moved to the ‘Review’ panel and had been communicated to the team. The task will then be committed to BitBucket where team members can access and test changes. If the quality of the task has not reached expectations, then the task will be pushed back to ‘In Progress’ to continue development. If the task has been confirmed as complete, then the task will be moved to the ‘Completed’ panel and is ready to be committed in the main branch.
This review process established consistency of quality developments to the project.

8.4. Testing

Throughout development of a new feature, we are constantly testing out the code / functions implemented using unity to emulate the game running on a mobile device.

Once the newly added feature works in the unity game engine, we will then go ahead and install the game onto a mobile device to test that there are no issues that the device emulator might have picked up. This is done either by
i) building the application to a file, then it is moved to the mobile device’s filesystem where it can be accessed and installed for testing.
ii) directly plugging the mobile device into your local computer and hitting the build +run settings via Unity.

Testing on a variety of android devices were done to find more bespoke problems or issues.

If bugs are found, Trello cards are made to amend those bugs and are usually fixed and tested once more. The application would often have bugs or errors after a big merge of different repository branches.

Once features are tested tasks related to them in Trello are shifted from the Review lane to the allocated Iteration lane.

8.5. Branching Strategy

8.5.1. Branching

At the end of the iteration our default main branch is the “2021_T1_handover” branch, as it was packaged up for handover to the next team.

Currently the strategy is that the team member’s work in groups split by their specialisation respectively into development and design teams. To which they create their own branches to work on from current most recent handover branch.

During our time with the project these branches were named: 2021_T1_Design and 2021_T1_Development.

Generally, developers team up with other developers and designers with designers however this is not always set in stone.

Once a team is happy with the features they have built and committed in their branch it’s discussed in one the weekly team meetings to which there is a branch merge into the team’s branch that needs the updated feature. For example, the design team updates the menu button design, the design branch is merged into the development branch.

At the end of an iteration both design and development branches are merged into a master branch.

The process of completing a merge will be discussed shortly first let’s look at the best practices working with GIT within the groups themselves.

8.5.2. Working with GIT

important

- Before you start any work, fetch and pull any changes.
- Communicate with the other members of your group, that you are not working on the same thing at the same time, and they don’t have any outstanding changes to commit push before you start.
- Once you have made your required changes and ready to commit, check with the group that no one else has anything to commit.
- After getting the go ahead, commit and push your changes, make sure to tell the group that there are changes to pull.
- Once you’re finished a session make sure that all outstanding changes have been committed and pushed and everyone knows to pull said changes next time they start.

The easiest way to do the above is to communicate the times you will be working on the project ahead of time or meet up virtually (Discord) to work together if working on it at the same time.

8.5.3. Merge Process

Note: Merges are generally pretty seamless, and conflicts easily dealt with provided that the two different teams have not worked on the same Unity scene file.

Unity scene files are created by the Unity used to represent a scene layout made up of the different scene components.

These files are saved in YAML format which by their nature are difficult to merge there are tools built into Unity to help with merging these files however (more info here)

Sometimes however these files just don't merge leaving to manually copying the different components over, therefor, when possible, teams should work in and with prefab objects to avoid the changes made to these scene files.

Merge Mid iteration into another team's branch.

- Communicate to the teams at a merge is about to occur so they don't commit anything during the merge (therefor so new commits/push during merge cause merge conflicts plus those new commits may not be carried over)
- Create a test branch of the branch to be merged into
- Merge the from branch into the test branch
- Look for merge conflicts (make note of the amount, impact and if they can be resolved)
- Check merge conflicts can be solved easily, and the project works as expected
- Checkout the real branch that the test branch was created from and repeat the above testing, after resolving the merge issues if they exist
- Communicate the merge has been completed to the team

In the event of unsolvable merge conflicts in the Unity YAML files try the following

- Take a local copy of project from the merge from branch

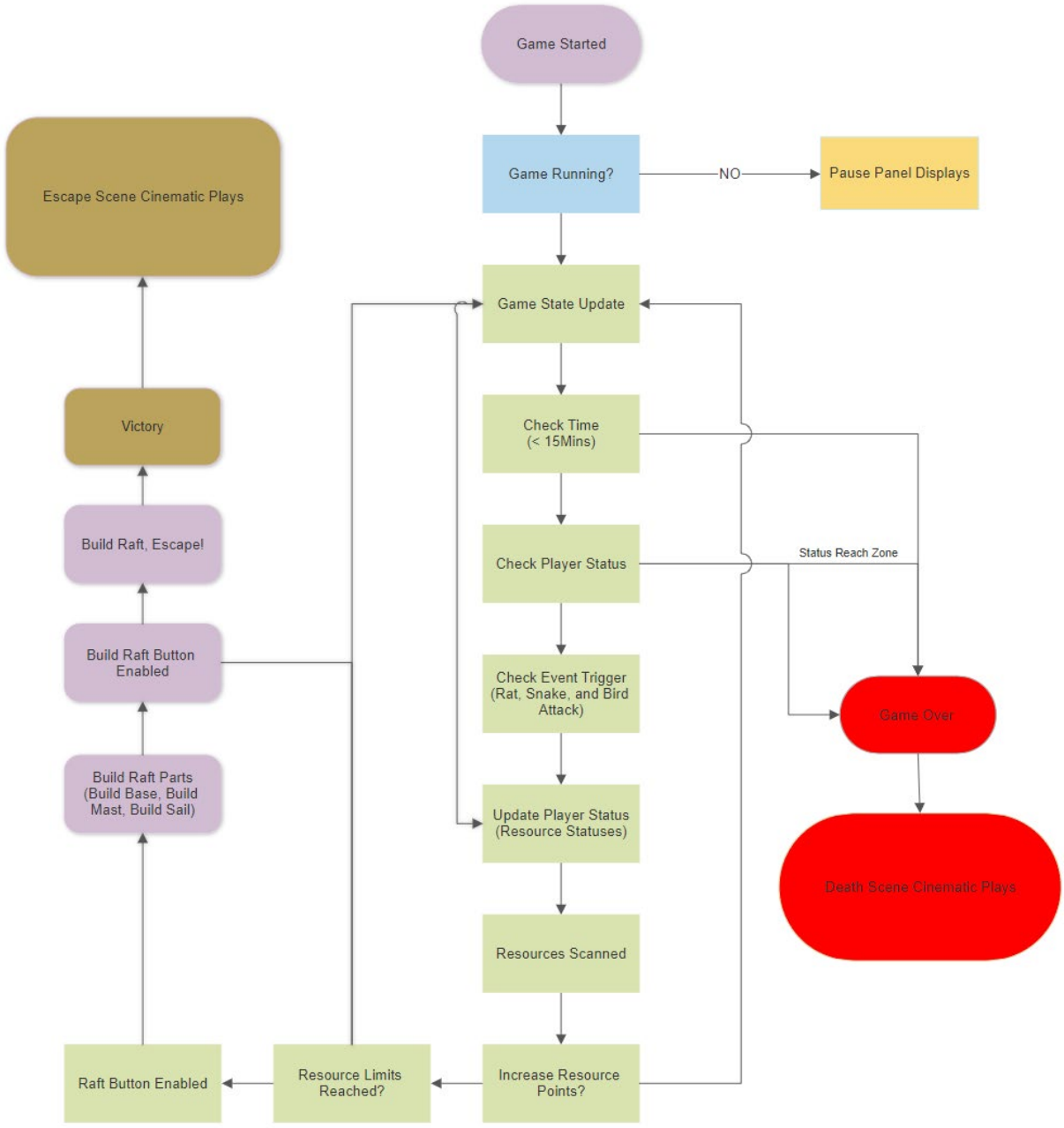
- Open it as sperate project within a Unity
- Checkout the merge to branch
- Merge the files as per above minus the YAML files causing the merge conflicts and their meta files
- With both Unity projects open manually copy paste and reconfigure the components from the local Unity project of the merge from branch to the current real Unity project
- Test the project
- Once complete commit and push the changes to the branch
- Checkout the from branch and merge the new changes as well so both branches are up to date
- Delete your local separate Unity project

End of iteration merge

Follow the same steps as above following merging to the end iteration branch, then merge back into both design and development branches.

9. Product Architecture

9.1. UML Diagram



9.2. Tech Stack

Software Used in Development:

- **Unity**: Cross platform game engine and development environment that has been used to develop the virtual reality mobile game. This platform has been chosen as the client preferred the use of this development platform and its integration with Vuforia, the augmented reality development tool kit.
- **Vuforia**: As the client had some experience with developing augmented reality applications with Vuforia prior to this project he requested that it was used when

developing ‘The Island’ as it was free and provided the level of precision and feature set required.

- **Microsoft Teams**: Communication, meeting scheduling platform and administration file sharing platform.
- **Discord**: Communication and collaboration platform. Used for collaborative working sessions, screen sharing and voice communications during iteration sprints. This was also used as our primary communication platform with the client.
- **Trello**: Project management platform used for task allocation, progression tracking and task review.
- **BitBucket**: Online code repository powered by Git.
- **Blender**: Used for 3D modelling, animation, lighting, rigging and image sequence editing allowing the cinematics and animations to be developed.
- **Adobe Photoshop CC**: Image generation and manipulation software used to create UI elements, target images, logos, and other imagery.
- **Adobe Premiere Pro**: Used to compress cinematics and alter frame rates. Also used for video editing.
- **Fruity Loops FL Studio**: Audio production software. Used to create sound effects and soundtracks.

10. Source Code

Link to the team’s Bitbucket repository is as follows:

https://bitbucket-students.deakin.edu.au/users/scorcoran/repos/escape-room-games---the-island/browse

For the video demo of getting set up with our repository see here <Video Demo>

The most up to date branch currently is the 2021_T1_Handover branch at the point of handover.

However, it is advised to create new branches post-handover from the end of iteration branch.

The main scripts (source code) can be found here within the project files as per the below path:

~/Escape Room Games - The Island/Assets/Scripts/

11. Login Credentials

Vuforia Developer Portal - https://developer.vuforia.com/

- The Vuforia developer portal is used to create and download Vuforia databases that contain the recognition data for multiple objects including trigger images, trigger objects, and Vumarks.

- The developer portal is also used to manage the Vuforia license and generate an API key that can either be used for cloud recognition or to use the databases downloaded.

- **Username**: info@appdevelopmentgeelong.com.au

- **Password**: EscapeTheIsland3

- **API Key**:
AYW5Wpf/////AAABmcHyxDzJ4kPTnIff2VgnzEmJlZTjZWetORXOlCDC9sqgdpuA1+ng7NzBhAJwW0Sm1Y1xvTXRaW+8ebCRxoIByc8b0dsOGIlkR029OqiUkgN7K+e2R+jRcOE5LNfIhGW+4+sjdM58w1i9HRl3uudVYG9hILdmTiH+wLIrh5wSq08dFYShCuOb0Bzj9ZVAK2uWVDq/tH34ZWliYRGCTyLJ5E0QJZXstsoLj8wvFlW5oRKy6l8JV+UNvbgHkJM7edI6FwZ3hiiKhUOAeJa92Zy78bsp7M3R5tuBjSb8Cuqb7ee971JjuGHoEG+/G6EU7PRs4Qg0wjKjIyBmtU+fi9UNx8hdKlM+CEghK5Z9saL9U4IV

Trello board / Roadmap - https://trello.com/en

- Trello is used as our project management platform.

- Individual access is granted to the client and to all Deakin emails for members of the project who request access

Bitbucket (Deakin hosted) - https://bitbucket-students.deakin.edu.au/login

- Source control platform used for collaborative development of the project.

- The repository is public and can be found here: https://bitbucket-students.deakin.edu.au/users/scorcoran/repos/escape-room-games---the-island/browse

- Individual accounts are granted write or admin access which are used to commit to the project's development

12. Appendices

Assets:

- **Blender Files: Link**

- **Game APK: Link**

- **Sound Files: Link**

Platforms:

- **Trello Board: Link**
- **Trello Roadmap: Link**

Videos:

- **How to play video: Link**

- **Showcase Video: Link**

***Source Folders can be found here: Link**