
The Island Game Group Assessment 3: Stage 1: Project Report Submission

OUA Building IT Systems (CPT111)

SP1, 2023

by

Sasha Bekier - s3335379,
Maddison Jackson - s3964407,
John Burkett - s3964404,
Sunggil Kang – s3992011,
James Gregory - s3992252.

May 2023

Table of Contents

1. Project Background	3
1.2. Motivation	3
1.3. The team	3
1.4. Project aim	6
1.5. Project goals	6
1.6 Project Scope	6
2. Project Progress	7
2.1. Description	7
2.2. Outcomes to date	8
2.3. Scope creep	14
2.4. Progress	16
2.5. Testing	16
2.6. Tools and Technologies	48
3. Challenges and Learning	51
3.1. Group challenges	51
3.2. How the challenges were addressed	52
3.3. Learning from the challenges	53
3.4. Changes	53
3.5. Project plan refinements	53
3.6. Timeline refinements	54
3.7. Risks and unexpected events	54
4. Marketing Pitch	55
5. Skills and Jobs	56
Scrum Master - 2D Game Development	56
Senior Unity Developer	57
2D Game Artist	58
Marketing Manager	59

1. Project Background

1.2. Motivation

Our team's goal is to develop a 2D Island Game (Turn-Based Combat Fantasy Game) called "Lost Isles" that caters to the interests of the adult gaming community on Steam and Itch.io. As eager gamers ourselves, we have a passion for fantasy and combat-based games and therefore as a group, we decided to create something unique that stands out from the overabundance of 2D games already available. Our game will feature an engaging storyline, original and different artwork, immersive different environments, various character classes, and innovative gameplay mechanics, among other elements, to appeal to adult gamers looking for something fresh and exciting.

Besides our love for gaming and the desire to create something unique, there are other reasons for choosing this project. One is the opportunity to enhance our skills and knowledge in game development, which we can utilise in our future careers. Additionally, completing this project will help us to build our portfolios and demonstrate our abilities to potential employers. As Unity is one of the most widely-used game creation platforms globally, we opted to use it for our game's development, allowing us to utilise its extensive features and support. Overall, our team is excited to take off on this project and bring our ideas to life.

1.3. The team

John Burkett

Student Number: s3964404

Student Email Address: s3964404@student.rmit.edu.au

Role in the project:

I've taken a software engineer/ developer role in the project.

Primarily design and technical writing with regards to the initial stages of the project, writing detailed documentation on the MVF's, as well as test plans and flow chart contributions with the goal of setting the right requirements that are achievable, but still a challenge.

I've taken a programming role towards the implementation phase of the project, contributing to code and implementation aspects of some key MVF's as well as quality assurance with code reviews for GitHub pull requests.

Personal Background and skills relevant to the project:

I've had a support and junior developer role with a software company that sells a CRM and accounting product. This has helped me with the design and technical writing phases of the project and some coding, however game development is a different ball game than I'm used to, although some skills have been transferable, there was still much to learn and skills to improve during the entire development cycle of this project.

Maddison Jackson

Student Number: s3964407

Student Email Address: s3964407@student.rmit.edu.au

Role in the project:

I have taken both a developer, game design, and UX designer and strategist role within the project.

Primarily I have worked towards the design of our puzzle features, UX of the game (e.g., main menu, settings, party and command system screens and more), and I have also worked towards the development of our puzzle system within our project. I have also contributed to report writing and writing documentation on MVF 3: The Puzzle System, as well as flow charts for this MVF.

Personal Background and Skills relevant to the project:

I have experience/background knowledge with UX design after completing a UX course and providing UX designs for specific applications. Not only this, I also have some knowledge in game design as somebody who personally plays games regularly and understands the positives and negatives of some game design features, and I believe this helps a lot during the process of game development. Lastly, report writing is something I have done for years, and I believe this is something I do very well in, and can therefore form professional reports for our project, highlighting and discussing the requirements for our reports.

Sasha Bekier

Student Number: s3335379

Student Email Address: s3335379@student.rmit.edu.au

Role in the project:

I have taken on the role of Team Leader and Lead Developer within our project. My primary leadership focuses have been on trying to maintain team cohesion, boost motivation and provide a level of oversight. I did not anticipate how much this would prove to be a teaching role.

As Lead Developer my focus has been on providing the ground work to allow the

other developers to get into their aspects of the project, getting the basics of our game laid out so the rest of the team had a world to work in.

Personal Background and Skills relevant to the project:

In previous incarnations I have been an adult educator, a website developer and a disability team leader and my role in this project has leaned heavily on that history. Apart from John I have the most coding experience in the team and that has allowed me to guide the dev process and provide valuable feedback to other team members, helping to improve their code and providing hurdle help when they have become stuck at various points in the development process.

James Gregory

Student Number: s3992252

Student Email Address: s3992252@student.rmit.edu.au

Role in the project:

My role was to design and (with the assistance of sasha) program the stats system, and design the initial concept of the class system before it was simplified. I also worked on the display of said stats and provided my ideas regarding other features such as combat, story, aesthetics, etc.

Personal Background and Skills relevant to the project:

I don't have much experience making games, but I did have knowledge regarding programming in python from an earlier course which made transitioning to programming in unity much easier. Additionally, I've played quite a number of rpgs before and lifted ideas that I liked from those and suggested/implemented them

Sunggil Kang

Student Number: s3992011

Student Email Address: s3992011@student.rmit.edu.au

Role in the project:

I have taken on the designer/developer role for this project and was mainly in charge of the combat system for our project. I have been contributing to core decisions that were made for the project and was also brainstorming and coming up with ideas and proposals for the project.

My role as a developer was to attempt to program the combat system (despite being a total beginner) that I had a major part in coming up with and learnt alot about Unity and how to properly use Github.

Personal Background and Skills relevant to the project:

I actually come from a background of biology and health which is a 180 degree pivot to computer science so I had minimal skills relevant to the programming part of the project. However, as a game enthusiast, I had many ideas on how to make our game more enjoyable, unique and make more sense as a game regarding its genre and goals.

1.4. Project aim

Our project seeks to design and develop a 2D adventure game called "Lost Isles," which will offer players an engaging and immersive experience while exhibiting our team's advanced game development, design, and programming skills. Our goal is to produce a game that not only provides entertainment but also presents a challenge to players, offering rewarding experiences that will entice them to continue playing.

1.5. Project goals

Below are the goals our group has achieved in our project to meet our project aim:

- Developed a captivating storyline with twists and turns that keep players engaged and motivated to progress and keep playing.
- Designed visually marvellous and detailed environments that immerse players in the game world, providing an original and memorable experience.
- Programmed challenging and strategic combat and game mechanics that require players to think critically and use their skills to overcome obstacles and succeed in the game.
- Implemented a user-friendly interface that is natural and easy to navigate, ensuring players can focus on the game's content and mechanics without being impeded by confusing controls and/or menus.
- Conducted extensive playtesting and gathered feedback from each group member to identify areas of the game that needed improvement and to fine-tune the difficulty and balance of the game.
- Optimised the game's performance for PC to ensure a smooth and consistent experience for our audience, adult PC gamers.

1.6 Project Scope

The objective of our project is to develop a 2D fantasy-based PC adventure-combat (turn-based) game called 'Lost Isles' that offers a distinctive and captivating

gameplay experience. Our team will utilise Unity as the game engine and limit its compatibility to PC platforms to the adult PC gaming community. We plan to employ Stable Diffusion and GitHub as project management tools to ensure the codebase is well-structured and comprehensively documented. Additionally, Adobe Photoshop and Stable Diffusion will be utilised to create visuals for the game.

Lost Isles will pose challenges to players by introducing a variety of obstacles, enemies, and puzzles. The game will require players to defeat adversaries, explore for treasure, complete missions and optional puzzles, and evade environmental dangers.

Our primary objective is to produce a 2D fantasy PC adventure-combat (turn-based) game, Lost Isles, that offers a pleasurable and demanding experience for players while showcasing our team's expertise in game development, design, and programming.

2. Project Progress

2.1. Description

Within our academic unit, "Building IT Systems," we were required to form a group and choose a project to complete within 12 weeks. Through the Building IT Systems discussion board on Canvas, our group of five members communicated our skills, experience, and project ideas. We agreed upon developing a 2D Island Game for the adult PC gaming community, which included a project manager, UX designer, game developers, a design team, and quality assurance specialists.

The project began with a kickoff meeting in Microsoft Teams, where the team discussed their skills, experience, expectations, and brief ideas for the 2D game. The project manager created a Trello board with project tasks, information, and requirements, allocating assessment tasks to each team member. This organised approach enabled each member to complete their allocated tasks efficiently, knowing their responsibilities and working towards desirable outcomes.

During the project, the team executed the plans discussed in two weekly MS Teams meetings, closely collaborating to bring the game to life. The project manager gathered assets for the game via Unity, Stable Diffusion, and Adobe Photoshop. Game developers and programmers worked on technical aspects such as game mechanics, optimization, and more. The UX developer designed the user interface, including introduction, main menu, settings, and party/command systems. Quality assurance specialists provided feedback, ensuring the game progressed as planned. The design team focused on aspects such as puzzle systems, class systems, environment design/art, easter eggs, environmental hazards, and enemies.

As the project progressed, changes to the original plan were necessary due to time constraints and workload. The team agreed to cut two islands and two puzzles from the game, ensuring the game met standards and features discussed within the project scope. While the team would have liked to add extra characters and classes, it was not feasible within the project timeline.

Each team member played a critical role in the project's success. The project manager ensured the project remained on track and all members communicated effectively. The designers, UX specialists, and developers collaborated to create a seamless user experience. The quality assurance specialist ensured requirements were met and the game was free of any bugs and glitches.

Overall, the project's success was due to effective project management, regular communication, and collaboration among the group members. Despite changes to the original plan, the team adapted and delivered a higher-quality product to the audience, meeting the standards and features within the project scope and plan.

2.2. Outcomes to date

MVF's:

- Landscape variety
 - The game generates a unique landmass every time it is booted up, filled with multiple different tiles (oceans, sand, grass, etc)



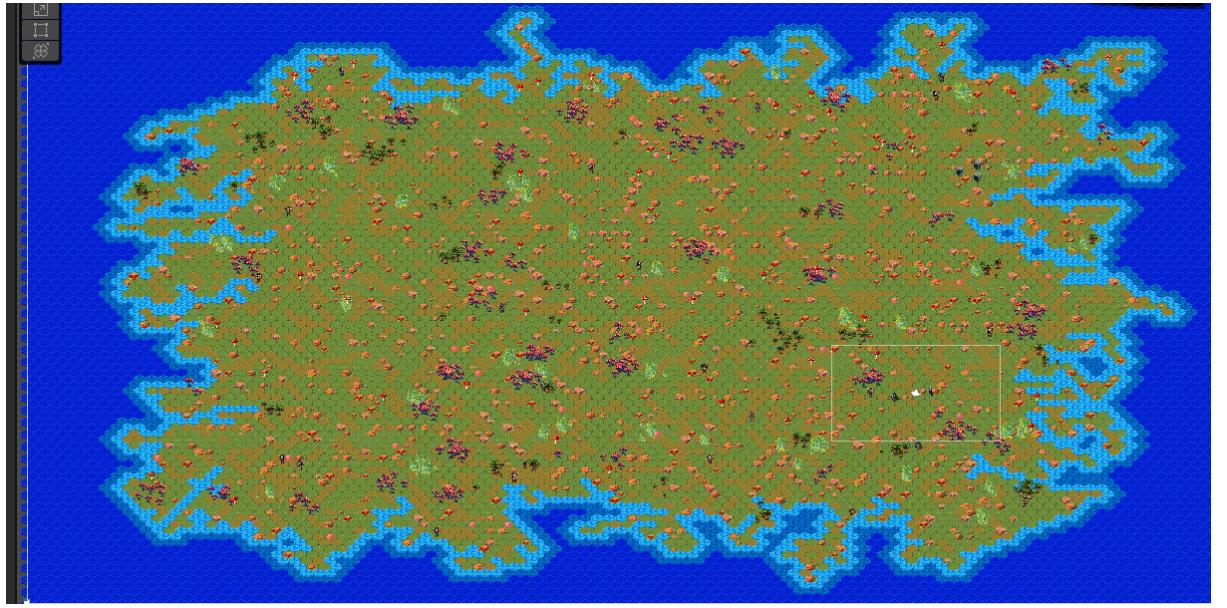


Fig 1 & 2: Two distinct procedurally generated islands on the same scale.

- There is only one explorable island, which is a tropical theme
- The view of the game dynamically changes based on the time spent in it



Fig 3 & 4 - The day and night implementation - note the time by the sky at upper left.

- The implemented island matches the tropical theme
- The player does interact with objects unique to the island

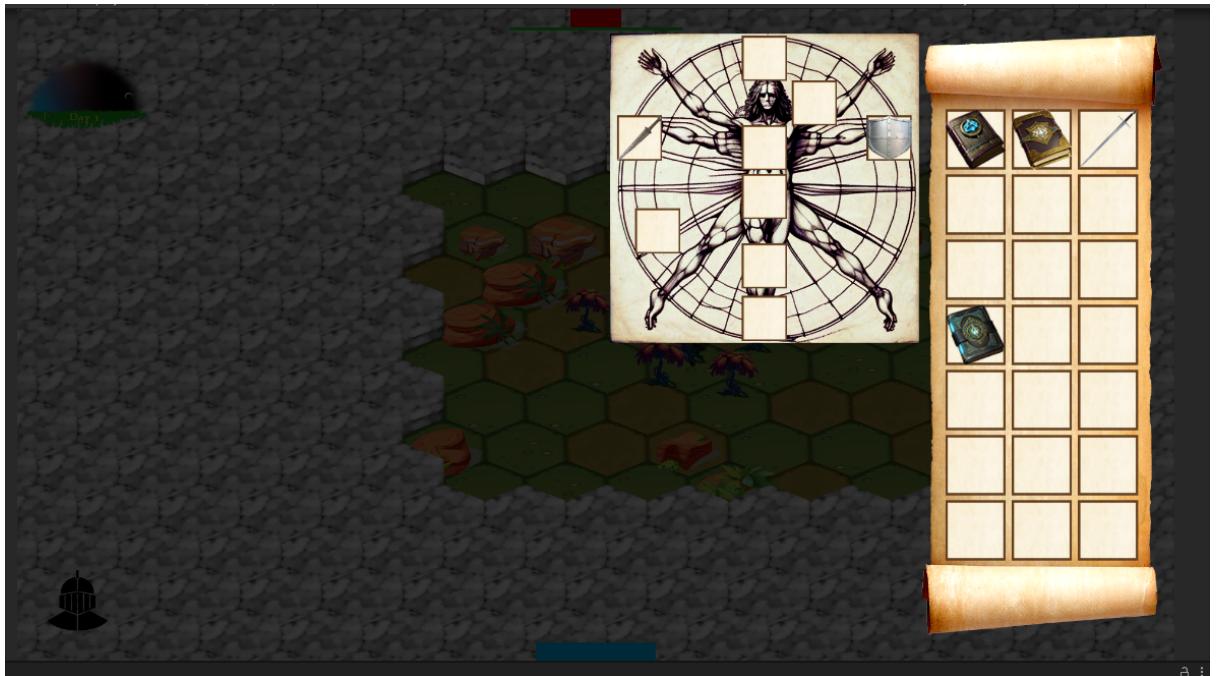


Fig 5. The player can collect items as they explore the island and store them in an inventory.

- No lore is currently present
- Combat system
 - There is one enemy type currently implemented, they spawn randomly across the map, have a detection radius and are capable of triggering a combat scene, but as of now, said combat scene is not implemented and shows up as a debug message



Fig 6. Enemies have a detect radius and a Debug.Log is triggered.

- Puzzle system
 - Right now, the puzzle consists of finding books around the island, and placing them into the altar which is placed randomly on the island on map generation. This teleports you to another area with more puzzles
 - Once combat is implemented, we plan to have it so that the books drop from enemies upon defeating them.



Fig 7 & 8. A book has been found and so have the three altars.



Fig 9 & 10. Books are placed upon the Altars, opening a portal.



Fig 11 & 12. The portal transports the character to a new location from which they can return.

- Character system
 - Originally we planned on implementing a full on class system similar to those found in other rpg games, but we found it would not be possible to develop it properly given our limited time. If we were to continue this, such a system would be implemented fully



Fig 13. Our preliminary Character Sheet - the two implemented classes advance differently as they gain levels.

- At the start of the game you get a choice between two characters, and when they level up their stats increase in slightly different ways.

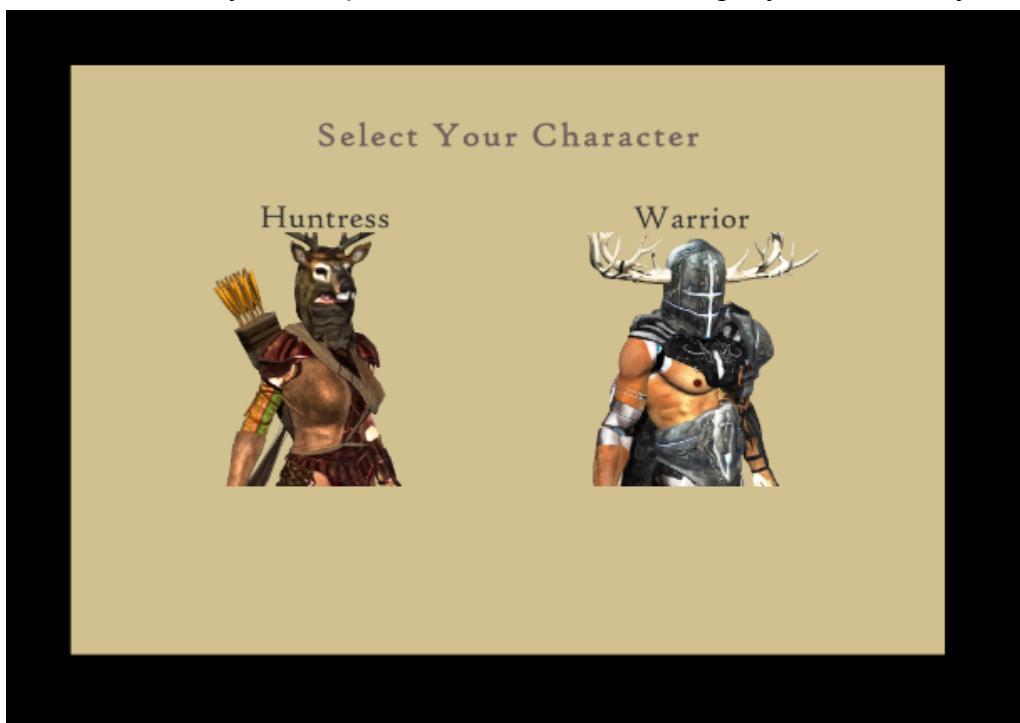


Fig 14. The launch of game character chooser.

- Realtime exploration
 - The player can navigate to any point on the island within view (assuming its reachable) by clicking on a tile on the map with their mouse
 - The game is in isometric mode
 - The player can interact with objects found on the ground
 - The player can not pass through ocean tiles or rocks, which act as blockers to movement
 - Combat is not currently implemented
 - The player gets experience by exploring the map and clearing the fog of war that covers it

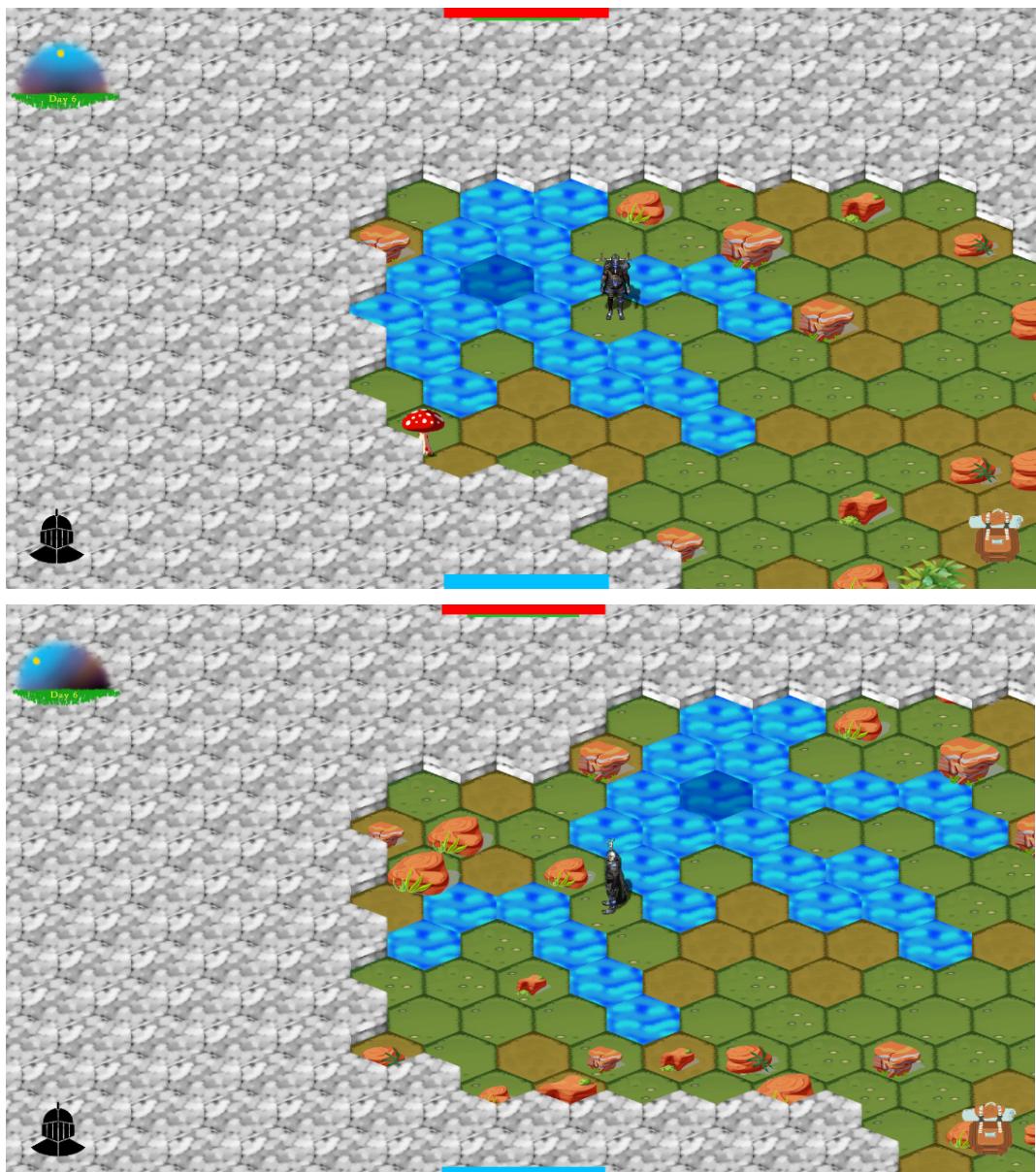


Fig 15 & 16. The player explores in real time and clears the fog of war as they go. Note that by the sun clock time has passed as the character moved from the location in 15 to the location in 16.

2.3. Scope creep

Multiple Islands feature removed

A part of MVF 1 with the procedurally generated environments was to include multiple levels with different environments.

Development of the world environment took a considerable amount of time, a conscious effort was made to manage the scope of the project and exclude this subset feature.

The base code exists to allow for variations of environments to be generated by easily adding assets, but how this occurs in game and the assets themselves do not currently exist and would require a reallocation of resources to complete.

Exciting these extra island themes allowed us to deliver a more polished Tropical island environment and ensure this environment was more immersive than 3 sparsely populated ones.

Weather Effects Removed

Landscapes also included a weather effects feature. This was dropped due to complexity and time constraints required to implement this feature. Some research and training was allocated to this feature but the implementation time was too high for the outcome of the goal, the risk of spending too much time on this feature outweighed the goal for it.

Instead a progressive day/ night time was added to the game to give some immersion to a changing environment without the need for complex animations and weather systems linked to specific islands/ environments.

Game Time Feature Added

After review of the gameplay, the topic of replayability was raised and a feature was required to be added to entice the player to want to replay the game. The idea was floated for an ‘in game time’ system to display while the player is in the game, where players could compare their times of other players or their own times to see who could beat the game the fastest. Although the game does not store this online for players to easily compare, a feature was added to the design to show in-game ‘days’ on the progressive day/ night UI element, where the days passed in the game appeared while the player was progressing. The time analysis to design and implement this feature was deemed low and the reward for implementing it was high, therefore it was added to the feature list with a higher priority. Implementation occurred quickly on this feature and was successfully implemented in a shorter timeline than planned resulting in an improvement of the final product.

Exploration Timed Jumps Removed

Exploration had a documented feature where Players could interact with timed jumps and movements, using their keyboard and mouse skills, to advance to secret locations or to reach a chest with rare items.

This feature was not clearly communicated in the team and in the original documentation, after completing all other requirements for the Exploration MVF once implementation started for jumping and movement it became clear that this feature was under specified and a vast amount of time would be required to re-design and implement it.

The removal of this feature did not appear to have a great impact on the exploration feel of the game, rewards were still available by exploring just without the player skills required to gain these rewards.

Focus was required on other features of the game that were not completed and this feature was removed from the current plan. It is still included as a future feature, but it will not be available in version 1.

Character Class System Simplified

The original character class system had detailed stat points where the player could delegate points to a specific stat allowing for custom characters to exist.

After implementation began on the class system it became clear that custom stats could result in unbalanced characters and potentially sour the players gaming experience by forcing them into stats that will ‘always work’.

It became clear that this feature was under-documented and that complex balancing would be required to make this a good experience for the player and it was removed.

Puzzles One Per Island Changed

The original plan was to have more than one (two or more) puzzle systems within a specific level (or island).

After implementing the first puzzle type, a more accurate timeline of how long it takes to implement became apparent and adding more puzzle types would require more time and resources than we had available. Rather than improving on one specific MVF, we decided to reduce the feature for this one particular feature and focus more resources on others

Combat Character Placement Removed

Character placement was a feature for the combat MVF where the player could pick the location of their character at the beginning of the combat event to give an advantage to the player class or disadvantage to the enemy class.

While developing the combat feature, each requirement for combat was unique and complex in its own right, such as the attack animations, the class/ stats analysis, etc. and it became clear that the time required to implement all these features would go beyond the allocated time. The timeline was adjusted to remove this from the game so that it would not impact the overall timeline.

With the removal of this feature, more attention was paid to quality assurance processes that ensured the rest of the game was well polished and with fewer bugs or game breaking events. The player experience will be impacted by this feature as it appears in similar games, however research was completed which made apparent that not all turn based games included movement or placement features like this which made the decision easier to remove this feature.

2.4. Progress

We have made fairly good progress, broadly in alignment with our timeline. The design process took two weeks longer than anticipated which placed the development process on the back foot and resulted in a necessary reduction in scope as our grand plans came face to face with the reality of the tight timeline presented by a four to five week implementation period.

Our timeline was designed on the premise that each team member would contribute around four hours a week towards implementation during the implementation period and we have all hit that target. It has however become clear that the expected time commitment was vastly inadequate particularly in relation to the level of learning required to operate in the Unity development environment. Several team members have committed many more hours than the timeline would suggest in order to keep the development process on track overall.

2.5. Testing

Test Category: 1 Landscape Variety

These cases are designed to test the Landscape Varieties when the character is exploring and interacting with the world in The Island Game.

This test case is designed to test how the custom environment assets can be swapped and tweaked in Unity to develop a unique world environment based on this simple asset swap and settings changes.

Test Category ID	1
-------------------------	---

Test Case ID	101
Test Case Summary	Verify a player can explore unique environmental landscapes across the different islands
Related Requirement	MVF1: Landscape Variety
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. Click on the 'WorldBuilder' object in the main SampleScene 3. In the Inspector modify the Island Size from a number between 50 and 250 4. Choose a different Tilemap 5. Add different elements to 'Plants' array 6. Add different elements to 'Rocks' array 7. Add different elements to 'Edibles' array 8. Add different elements to 'Enemies' array 9. Save and press 'Play' mode 10. Explore the world game environment generated with your entered map size and discover these added array elements
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>A developer can swap assets Terrain Tilemaps, Plants, Items and impassable objects (like rocks) in the WorldBuilder tool to create unique environments</p> <p>The world environment built should be unique on each Play.</p> <p>Each asset should be present in the world as a ground tile, plant or foliage, an impassable item such as rocks or an item to pickup like food or weapons, a custom enemy to fight such as a pirate.</p>

Actual Result	A unique procedurally generated world is generated on each play. A custom size world is generated The selected tile pallet generates the walkable tiles. The selected Plants generated as walkable foliage. The selected Rocks generated as impassable objects. The selected Items appeared as pickable food/ puzzles The selected Enemies appeared as sprites to trigger combat
Status	Passed
Remarks	Not all assets have been created, you can explore with given assets or search online for samples to add custom assets to build a custom world.
Created By	John Burkett
Date of Creation	12/05/2023
Executed By	John Burkett
Date of Execution	16/05/2023
Test Environment	Blank world generation build

This test case is designed to test how the look and feel of a specific environment is consistent throughout the island that is created based on the labelling and asset folders for the assets selected.

Test Category ID	1
Test Case ID	102
Test Case Summary	Verify when a player is exploring an environment that this unique environmental landscape is consistent across the specific island.
Related Requirement	MVF1: Landscape Variety
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later

Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. Click on the ‘WorldBuilder’ object in the main SampleScene 3. Choose a different Tilemap 4. Add different elements to the custom settings to the ‘Plants’, ‘Rocks’, ‘Edibles’ and ‘Enemies’ arrays. 5. Each different array items should be grouped in asset folders specific to that environment. I.e. all Tropical Island assets are grouped with the tags and folders for ‘Tropical’. 6. Save and press ‘Play’ mode 7. Explore the world game environment generated to the specific selected environment type based on the asset label and folders.
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>A developer can swap assets Terrain Tilemaps, Plants, Items and impassable objects (like rocks) in the WorldBuilder tool to match the expected specific environments (tropical).</p> <p>Each Tropical asset should be present in the world as a grass ground tile, Red Palms, Green Palms, Little Ferns, IslandRock1-3 as impassable, Apple and Mushroom items to pick up and Pirate Enemies.</p>
Actual Result	<p>A unique procedurally generated world is generated on each play.</p> <p>A custom size world is generated</p> <p>The selected tile palette generates the walkable tiles.</p> <p>The selected Plants generated as walkable foliage.</p> <p>The selected Rocks generated as impassable objects.</p> <p>The selected Items appeared as pickupable food/ puzzles</p> <p>The selected Enemies appeared as sprites to trigger combat</p>
Status	Passed
Remarks	Not all assets have been created, you can explore with given assets or search online for samples to add custom assets to build a custom world.
Created By	John Burkett

Date of Creation	01/05/2023
Executed By	
Date of Execution	
Test Environment	Custom world generation build

This test case is designed to test how the look and feel of the Tropical environment is consistent throughout the island.

Test Category ID	1
Test Case ID	103
Test Case Summary	Verify when a player is exploring the Tropical environment that assets spawn specifically for this environmental landscape and is consistent across the Tropical island.
Related Requirement	MVF1: Landscape Variety
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. Click on the 'WorldBuilder' object in the main SampleScene 3. Leave the default Tilemap 4. Leave the default elements for 'Plants', 'Rocks', 'Edibles' and 'Enemies' arrays. 5. Save and press 'Play' mode 6. Explore the world game environment generated to a Tropical theme
Test Data	NA

Expected Result	The expected result of the test. Each Tropical asset should be present in the world as Tilemap: a grass ground tile Border: Tropical Water. Plants: Red Palms, Green Palms, Little Ferns Impassable: IslandRock1-3 Pickup Items: Apple and Mushroom items Enemies: Pirates
Actual Result	The default Island Grass tile pallet was generated. The default Island Border tiles of water was generated. The default Island Plants Red Palm, Green Palm and Ferns were generated. The default Island impassable rocks 1-3 were generated The default Island Apple and Mushroom Items were spawned. The default Island Pirate Enemies were spawned.
Status	Passed
Remarks	
Created By	John Burkett
Date of Creation	01/05/2023
Executed By	Sasha Bekier
Date of Execution	12/05/2023
Test Environment	Default world generation build

This test case is designed to ensure the generation of the Tropical environment does not include assets, items or environment settings from different environments (e.g. volcanic).

Test Category ID	1
Test Case ID	104
Test Case Summary	Verify when a player is exploring an environment that this unique environmental landscape is consistent across the specific island.

Related Requirement	MVF1: Landscape Variety
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. Click on the 'WorldBuilder' object in the main SampleScene 3. Leave the default Tilemap 4. Leave the default elements for 'Plants', 'Rocks', 'Edibles' and 'Enemies' arrays. 5. Save and press 'Play' mode 6. Explore the world game environment generated to a Tropical theme
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>Each Tropical asset should be present in the world as</p> <p>Tilemap: a grass ground tile Border: Tropical Water. Plants: Red Palms, Green Palms, Little Ferns Impassable: IslandRock1-3 Pickup Items: Apple and Mushroom items Enemies: Pirates No other assets or environments are present from other Island such as the puzzle Island or Volcanic/ Winter Islands</p>
Actual Result	<p>The Island Grass tile pallets only was generated.</p> <p>The Island Border tiles of water only was generated.</p> <p>The Island Plants Red Palm, Green Palm and Ferns only were generated and no other plant types present.</p> <p>The Island impassable rocks only 1-3 were generated and no other types present.</p> <p>The Island Apple and Mushroom Items were spawned and no other types present.</p> <p>The Island Pirate Enemies only were spawned and no other enemy types present.</p>
Status	Passed
Remarks	

Created By	John Burkett
Date of Creation	04/05/2023
Executed By	Sasha Bekier
Date of Execution	14/05/2023
Test Environment	Default world generation build

This test case is designed to test the weather effects for the in-game environment to create an immersive

Test Category ID	1
Test Case ID	105
Test Case Summary	Verify when a player is exploring in game that weather effects occur.
Related Requirement	MVF1: Landscape Variety
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. Press 'Play' mode 3. Explore the world game environment generated and notice weather effect changes
Test Data	NA
Expected Result	The expected result of the test. The lighting of the environment in game should progressively get lighter or darker based on the time of day.
Actual Result	The daytime/ night-time dial will slowly move between day and night. The game environment becomes darker then lighter while the daytime/ night-time timer moves.

Status	Passed
Remarks	Weather effects became out of scope due to the complexity and time constraints, instead a day/ night-time effects were added to give variety to the game exploration.
Created By	John Burkett
Date of Creation	07/05/2023
Executed By	Sasha Bekier
Date of Execution	12/05/2023
Test Environment	Blank world generation build

Test Category: 2 Combat System

These cases are designed to test the Combat System when the character is fighting enemies in The Island Game.

This test case is designed to test how the player exploring the world environment can interact with enemy characters.

Test Category ID	2
Test Case ID	201
Test Case Summary	Verify a player can interact with enemy objects within the generated environment.
Related Requirement	MVF2: Combat
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later

Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and approach an enemy asset. 4. Left click on the enemy Character when standing on a tile adjacent to the enemy 4. Review the console log for the interaction
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>The world environment built should contain enemy characters spawned.</p> <p>When left mouse clicking on an enemy asset, nothing should log in the console unless the character is in a tile immediately next to the enemy tile.</p> <p>Other mouse buttons should not log.</p>
Actual Result	<p>A console log of 'Enemy left click' was present when standing next to the enemy in a random spawn location.</p> <p>No logging occurred when clicking on the enemy from 2 tiles over or any other location.</p>
Status	Passed
Remarks	
Created By	Sunggil Kang
Date of Creation	12/05/2023
Executed By	John Burkett
Date of Execution	14/05/2023
Test Environment	Default world generation build

This test case is designed to test how the player combat UI should initiate after clicking on enemy characters.

Test Category ID	2
-------------------------	---

Test Case ID	202
Test Case Summary	Verify a player can initiate the combat UI after clicking on enemy objects.
Related Requirement	MVF2: Combat
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and approach an enemy asset. 4. Left click on the enemy Character when standing on a tile adjacent to the enemy 4. The UI should start a Combat based screen with a choice to run away or fight
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>A fight selection screen should appear with the option to fight or run away.</p>
Actual Result	A small splash screen appeared but the Fight option was not present.
Status	Fail
Remarks	
Created By	Sunggil Kang
Date of Creation	12/05/2023
Executed By	John Burkett
Date of Execution	19/05/2023
Test Environment	Default world generation build

This test case is designed to test how the player can run away from a combat event.

Test Category ID	2
Test Case ID	203
Test Case Summary	Verify a player can escape combat right after initiation only.
Related Requirement	MVF2: Combat
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and approach an enemy asset. 4. Left click on the enemy Character when standing on a tile adjacent to the enemy 4. On the UI screen select the 'Run away' option.
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>After clicking Run away the Fight UI should go back to the exploration UI.</p> <p>The enemy asset should remain in the game.</p>
Actual Result	
Status	
Remarks	
Created By	Sunggil Kang
Date of Creation	16/05/2023
Executed By	
Date of Execution	
Test Environment	Default world generation build

This test case is designed to test how the player character can choose a move to fight.

Test Category ID	2
Test Case ID	204
Test Case Summary	Verify a player can escape combat right after initiation only.
Related Requirement	MVF2: Combat
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and approach an enemy asset. 4. Left click on the enemy Character when standing on a tile adjacent to the enemy 5. On the UI screen select the 'Fight' option. 6. Choose the 'Attack' Move from the list of options
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>After clicking attack the character should make an attack animation and the Enemies health should reduce.</p>
Actual Result	
Status	
Remarks	
Created By	Sunggil Kang
Date of Creation	16/05/2023
Executed By	
Date of Execution	

Test Environment	Default world generation build
-------------------------	--------------------------------

This test case is designed to test how the player can die during a combat event.

Test Category ID	2
Test Case ID	205
Test Case Summary	Verify a player can die during combat and lose the interaction.
Related Requirement	MVF2: Combat
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and approach an enemy asset. 4. Open the Player inventory 5. Remove all items of the inventory so there is little attack damage. 6. Find and Left click on the enemy Character when standing on a tile adjacent to the enemy 7. On the UI screen select the 'Fight' option. 8. On the screen select the 'Block' option each time until the player runs out of health.
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>While being attacked, if the players health reaches 0 the player should enter a 'dead' state with a death animation playing and a 'restart' option appearing. This should close the combat screen.</p>
Actual Result	

Status	
Remarks	
Created By	Sunggil Kang
Date of Creation	16/05/2023
Executed By	
Date of Execution	
Test Environment	Default world generation build

Test Category: 3 Puzzle System

These cases are designed to test the Puzzle System when the character interacts with puzzle items while exploring the world in The Island Game.

This test case is designed to test how the puzzle items spawn within the exploration environment.

Test Category ID	3
Test Case ID	301
Test Case Summary	Verify a player can interact with puzzle items and objects within the generated environment.
Related Requirement	MVF3: Puzzle System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and walk over a puzzle item/ book. 4. Left click on the player inventory and view the item.
Test Data	NA

Expected Result	The expected result of the test. The world environment built should contain 3 books items spawned at random locations on the ground. When walking over these items they should disappear from the tile and appear in the player inventory screen.
Actual Result	3 Book items were found in the exploration environment. No more. Walking over the book items moved them from the ground into the inventory system simulating an item being picked up.
Status	Passed
Remarks	
Created By	Maddison Jackson
Date of Creation	04/05/2023
Executed By	Sasha Bekier
Date of Execution	04/05/2023
Test Environment	Default world generation build

This test case is designed to test how the picked-up books can be placed on the puzzle altar.

Test Category ID	3
Test Case ID	302
Test Case Summary	Verify a player can interact with puzzle items and objects within the generated environment.
Related Requirement	MVF3: Puzzle System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later

Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and walk over a puzzle 3 books 4. Find and walk to the centre of the altar. 5. Open the inventory tab. 6. Left click on each book in the inventory tab.
Test Data	NA
Expected Result	The expected result of the test. Each book should move from the inventory tab to the 3 altars in the environment.
Actual Result	Click on h each book has moved it to the altars with each unique book appearing on an altar
Status	Passed
Remarks	
Created By	Maddison Jackson
Date of Creation	12/05/2023
Executed By	Sasha Bekier
Date of Execution	14/05/2023
Test Environment	Default world generation build

This test case is designed to test how the puzzle altar can transport the player through a portal to a 'prize' realm.

Test Category ID	3
Test Case ID	303
Test Case Summary	Verify a player completing the puzzle is moved to the 'Prize' realm.

Related Requirement	MVF3: Puzzle System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and walk over a puzzle 3 books 4. Find and walk to the centre of the altar. 5. Open the inventory tab. 6. Left click on each book in the inventory tab. 7. Close the inventory tab.
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>The player should be instantly teleported to a magical room with an escape portal.</p>
Actual Result	Player was moved to a small room with a different environment with a portal in the middle.
Status	Passed
Remarks	
Created By	Maddison Jackson
Date of Creation	12/05/2023
Executed By	Sasha Bekier
Date of Execution	14/05/2023
Test Environment	Default world generation build

This test case is designed to test how the prize realm contains random weapons and items.

Test Category ID	3
Test Case ID	304
Test Case Summary	Verify the 'Prize' realm has random items and weapons (loot) as prizes for completing the puzzle.
Related Requirement	MVF3: Puzzle System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and walk over a puzzle 3 books 4. Find and walk to the centre of the altar. 5. Open the inventory tab. 6. Left click on each book in the inventory tab. 7. Close the inventory tab 8. Walk around the small prize realm room.
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>The floor of the prize realm should contain different weapons and items. Replaying the test procedure should result in randomly different items.</p> <p>Walking over the items should move them to the player inventory.</p>
Actual Result	
Status	
Remarks	
Created By	Maddison Jackson
Date of Creation	12/05/2023

Executed By	
Date of Execution	
Test Environment	Default world generation build

This test case is designed to test how a puzzle can be partially completed and returned to at a later time.

Test Category ID	3
Test Case ID	305
Test Case Summary	Verify a player can interact with a puzzle, leave it incomplete, come back to it at a later time and complete it.
Related Requirement	MVF3: Puzzle System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Explore the world game environment generated and walk over a puzzle 2 books 4. Find and walk to the centre of the altar. 5. Open the inventory tab. 6. Left click on each book in the inventory tab. 7. Close the inventory tab 8. Walk around the environment and find the last puzzle book. 9. Return to the centre of the altar and open the inventory. 10. Left click on the one puzzle book showing in the inventory
Test Data	NA

Expected Result	The expected result of the test. The first attempt at the puzzle should result in 2 books appearing in the altar and being removed from the player's inventory. The second attempt at the puzzle should teleport and reward the player.
Actual Result	2 Book items were found in the exploration environment and put on the altar. Exploration continued and one more book was found and placed on the altar. The puzzle completed and player moved to the prize realm
Status	Passed
Remarks	
Created By	Maddison Jackson
Date of Creation	15/05/2023
Executed By	Sasha Bekier
Date of Execution	18/05/2023
Test Environment	Default world generation build

Test Category: 4 Class System

These cases are designed to test the Class System when the player can select unique character classes in The Island Game.

This test case is designed to test how the player can pick 2 visual unique classes of characters.

Test Category ID	4
Test Case ID	401
Test Case Summary	Verify a player can pick two visually unique characters.

Related Requirement	MVF4: Class System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	<p>Step-by-step procedure to execute the test.</p> <ol style="list-style-type: none"> 1. Open the Island Game in Unity 2. 'Play' mode 3. Left click on the Huntress character. 4. Walk around the world. 5. Restart the game and press 'Play' mode. 6. Left click on the Warrior character. 7. Walk around the world.
Test Data	NA
Expected Result	<p>The expected result of the test.</p> <p>The Character Select screen should show two distinct classes for Ranged and Melee</p> <p>Left clicking on the Classes should load the game with that character object appearing in the environment.</p> <p>The selected Character Type should animate as per that specific Character.</p>
Actual Result	<p>Select screen shows two characters.</p> <p>After selecting one character the generated world contains a sprite animations of that character.</p>
Status	Passed
Remarks	
Created By	John Burkett
Date of Creation	04/05/2023
Executed By	Sasha Bekier
Date of Execution	04/05/2023
Test Environment	Default world generation build

This test case is designed to test whether the player can pick a Melee type class character.

Test Category ID	4
Test Case ID	402
Test Case Summary	Verify a player can choose a melee type character.
Related Requirement	MVF4: Class System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Left click on the Warrior character. 4. Walk around the world.
Test Data	NA
Expected Result	The expected result of the test. The splash screen should contain an option for a Warrior/ Melee type character. Selecting Warrior should show the Warrior character in the exploration environment. Exploring the environment should animate for the Warrior class.
Actual Result	The warrior appears in the character select screen. The warrior animation moving in the environment matches this class.
Status	Passed
Remarks	
Created By	John Burkett
Date of Creation	08/05/2023
Executed By	Sasha Bekier

Date of Execution	04/05/2023
Test Environment	Default world generation build

This test case is designed to test if the player can pick a Ranged type class character.

Test Category ID	4
Test Case ID	403
Test Case Summary	Verify a player can choose a Ranged type character.
Related Requirement	MVF4: Class System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Left click on the Huntress character. 4. Walk around the world.
Test Data	NA
Expected Result	The expected result of the test. The splash screen should contain an option for a Huntress / Ranged type character. Selecting Huntress should show the Huntress character in the exploration environment. Exploring the environment should animate for the Huntress class.
Actual Result	The Huntress appears in the character select screen. The Huntress animation moving in the environment matches this class.
Status	Passed
Remarks	

Created By	John Burkett
Date of Creation	08/05/2023
Executed By	Sasha Bekier
Date of Execution	04/05/2023
Test Environment	Default world generation build

This test case is designed to test whether the picked Melee class character stats are unique.

Test Category ID	4
Test Case ID	404
Test Case Summary	Verify the warrior type character has stats matching a melee class.
Related Requirement	MVF4: Class System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Left click on the Warrior character. 5. Left click on the Character stats icon in the bottom left
Test Data	NA
Expected Result	The expected result of the test. When opening the stats screen the following base stats should appear Strength 25 Dexterity 5 Attack 50

	Ranged 10 Movement 10
Actual Result	Opening the stats panel displayed base stats as Strength 25 Dexterity 5 Attack 50 Ranged 10 Movement 10
Status	Passed
Remarks	
Created By	James Gregory
Date of Creation	17/05/2023
Executed By	Sasha Bekier
Date of Execution	18/05/2023
Test Environment	Default world generation build

This test case is designed to test if the picked Ranged class character stats are unique.

Test Category ID	4
Test Case ID	405
Test Case Summary	Verify the warrior type character has stats matching a ranged class.
Related Requirement	MVF4: Class System
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later

Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Left click on the Huntress character. 5. Left click on the Character stats icon in the bottom left
Test Data	NA
Expected Result	The expected result of the test. When opening the stats screen the following base stats should appear Strength 5 Dexterity 25 Attack 10 Ranged 50 Movement 10
Actual Result	Opening the stats panel displayed base stats as Strength 25 (no match) Dexterity 5 (no match) Attack 50 (no match) Ranged 10 (no match) Movement 10
Status	Failed
Remarks	
Created By	James Gregory
Date of Creation	17/05/2023
Executed By	Sasha Bekier
Date of Execution	18/05/2023
Test Environment	Default world generation build

Test Category: 5 Real Time Exploration

These cases are designed to test the real time exploration of the world environment in The Island Game.

This test case is designed to test how the character will face the direction of movement by following the mouse location.

Test Category ID	5
Test Case ID	501
Test Case Summary	Verify a character will always face the direction of the mouse when stationary.
Related Requirement	MVF5: Real Time Exploration
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Pick any character. 4. Move the mouse around the world environment without clicking
Test Data	NA
Expected Result	The expected result of the test. The Character idle animation should face the direction of the mouse. moving the mouse should change the facing direction to accurately portray where the character would move if the mouse was clicked.
Actual Result	Mouse movements and idle animations matches where the character is facing
Status	Passed

Remarks	
Created By	John Burkett
Date of Creation	01/05/2023
Executed By	Sasha Bekier
Date of Execution	02/05/2023
Test Environment	Default world generation build

This test case is designed to test the character moves to the clicked free location on the main grid.

Test Category ID	5
Test Case ID	502
Test Case Summary	Verify a character will move the clicked location
Related Requirement	MVF5: Real Time Exploration
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Pick any character. 4. Click on a tile location where no objects are present.
Test Data	NA
Expected Result	The expected result of the test. The Character should start a walk animation to the direction they are facing. The movement animation should match the direction they are moving in.
Actual Result	The character and camera follows the character moving and the walk animations accurately match the direction they are travelling.
Status	Passed

Remarks	
Created By	John Burkett
Date of Creation	07/05/2023
Executed By	Sasha Bekier
Date of Execution	14/05/2023
Test Environment	Default world generation build

This test case is designed to test the character does not move to the clicked free location on the main grid if the tile is an impassable type (e.g. rock or water grid).

Test Category ID	5
Test Case ID	502
Test Case Summary	Verify a character will move the clicked location
Related Requirement	MVF5: Real Time Exploration
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Pick any character. 4. Click on a tile location where a rock or impassable objects are present.
Test Data	NA
Expected Result	The expected result of the test. The Character should not move. The movement animation not should occur and the idle animation should match the direction they were clicked

Actual Result	The character does not move and the idle animation remains.
Status	Passed
Remarks	
Created By	John Burkett
Date of Creation	09/05/2023
Executed By	Sasha Bekier
Date of Execution	11/05/2023
Test Environment	Default world generation build

This test case is designed to test the character movement pass results in the character moving around an impassable type (e.g. rock or water grid) not walking through it or stopping their movement.

Test Category ID	5
Test Case ID	504
Test Case Summary	Verify a character will move around a blocked object if clicked location is a free tile beyond an impassable one.
Related Requirement	MVF5: Real Time Exploration
Prerequisites	The Island Game: current 'main' branch. Unity Version 2021.3.21f1 or later
Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Pick any character. 4. Click on a tile location where a rock or impassable objects is in between the character and the destination tile (blank walkable tile)
Test Data	NA

Expected Result	The expected result of the test. The Character should not move. The movement animation not should occur and the idle animation should match the direction they were clicked
Actual Result	The character does not move and the idle animation remains.
Status	Passed
Remarks	
Created By	John Burkett
Date of Creation	12/05/2023
Executed By	Sasha Bekier
Date of Execution	12/05/2023
Test Environment	Default world generation build

Pickup objects while moving.

This test case is designed to test a character picks up a ‘pickupable’ object when directly walked over.

Test Category ID	5
Test Case ID	505
Test Case Summary	Verify a character will pick up objects on the ground when they directly walk over them
Related Requirement	MVF5: Real Time Exploration
Prerequisites	The Island Game: current ‘main’ branch. Unity Version 2021.3.21f1 or later

Test Procedure	Step-by-step procedure to execute the test. 1. Open the Island Game in Unity 2. 'Play' mode 3. Pick any character. 4. Click on a tile location where an apple or mushroom is located
Test Data	NA
Expected Result	The expected result of the test. The character should move over or to the item. Once reached the item should disappear from the main environment and appear in the character inventory.
Actual Result	The items walked over are moved from the environment to the characters possession.
Status	Passed
Remarks	
Created By	John Burkett
Date of Creation	12/05/2023
Executed By	Sasha Bekier
Date of Execution	12/05/2023
Test Environment	Default world generation build

2.6. Tools and Technologies

Below are the Tools and Technologies used to develop our 2D project 'Lost Isles.' All Tools and Technologies discussed below are all free-to-use, and are at zero cost.

Market Places:

Below are the 'MarketPlaces' we will be using to sell our 2D game 'Lost Isles' to our target audience, the Adult PC Gaming Community. These MarketPlaces will allow the PC Community to purchase our game and store the game on their chosen marketplace (e.g., Steam, Itch.io), where they can play the game whenever they desire.

Steam: <https://store.steampowered.com/>

Steam is referred to as a video game digital distribution service that allows PC gaming communities to purchase and store games onto their Steam account. We have selected Steam as one of our marketplaces because it is widely regarded as one of the most popular video game distribution services available today, with a significant concentration of our target audience (the Adult PC Gaming Community). Steam will be used to sell our 2D game 'Lost Isles' to our target audience, where they can store the game on their Steam account, allowing them to access the game when desired.

Itch.io: <https://itch.io/>

Itch.io is a platform where indie game developers can host, sell, and distribute their games. Our group has decided to use Itch.io as one of our marketplaces because it provides a great opportunity for new game developers to showcase their creations to the public. As a new development team creating a game, we believe that Itch.io's openness to promoting new games aligns with our goals. Therefore, we have decided Itch.io is another marketplace we have considered to sell our 2D game, 'Lost Isles.'. This will give the adult PC gaming community a chance to purchase and enjoy our game.

Development Tools:

Unity: <https://unity.com/>

Unity, version 2022.2.9, is a game engine that enables developers to create video games across multiple platforms. It is built using C# and UnityScript (JavaScript) and has gained popularity in the industry. Our group has chosen to use Unity for our project, as we believe it is more beginner-friendly and better suited to our needs for creating a 2D game. Learning C# and UnityScript can prove advantageous to group members, as these skills are in demand in various industries. For our project, we will be using C#.

As an example of our team's proficiency in utilising Unity, we have successfully built and rendered our game in a physical form, incorporating our assets and coding them in C# to perform specific actions such as moving the character upon click, ensuring the character does not collide with objects by utilising 2D collision features, and making the character face the direction of the mouse cursor and much more.

Adobe Photoshop: <https://www.adobe.com/au/products/photoshop.html>

Our group, primarily using Windows, will benefit from Adobe Photoshop, a raster graphics editor that operates on both Windows and macOS platforms. With its assistance, we can create various elements for our 2D Island Game 'Lost Isles,' such as UI icons, backgrounds, game art, assets, and more. Photoshop has been used for its AI Object Select Function to pull things out of the Stable Diffusion generated images.

Stable Diffusion: <https://stablediffusionweb.com/>

The CompVis group launched Stable Diffusion, a text-to-image model, in 2022. By inputting text, users can randomly generate intricate images. For our 2D Island Game 'Lost Isles' and other design projects, Stable Diffusion will be a valuable tool in creating graphics. Within Lost Isles, we have utilised Stable Diffusion to create various tiles, including grassy tiles, snowy tiles, water tiles, and deep water tiles.

Collaborative Workspaces:

The Island Game Group will use the following collaborative workspaces and tools for the development of our project:

Trello: <https://trello.com/b/L6SYBpii/oua-sp1-2023-the-island-game>

Our team utilises Trello on a daily basis as a means of tracking multiple tasks for each member and allocating tasks to group members. Trello has proven to be an invaluable tool for the development of Lost Isles, as it promotes organisation and enhances our awareness of each other's tasks and development progress.

Using Trello as a project management tool has been proven to promote the development of Lost Isles by streamlining task management and improving communication among each team member. With Trello, our team leader can efficiently assign tasks to each member, set deadlines, track progress, and ensure that everyone is working towards our own goals. The tool's visual nature also makes it easy to see the big picture of the project and identify any potential roadblocks or bottlenecks that need to be addressed. By using Trello, our team can work more efficiently and effectively, which ultimately leads to a faster and more successful development process of Lost Isles.

Microsoft Teams: <https://teams.microsoft.com/>

Our team uses Microsoft Teams as a means of maintaining regular communication related to the Lost Isles project. This includes sharing updates on project status, discussing any issues or challenges that arise, exchanging ideas and making decisions, as well as hosting weekly team meetings and sharing project-related documents and files. By utilising Microsoft Teams, the team can stay connected and

collaborate more efficiently, which helps to promote the development of Lost Isles by enabling better coordination and faster decision-making.

GitHub and GitHub Desktop: <https://github.com/SashaBekier/BITS-IslandGame>

We are utilising GitHub/GitHub Desktop as a platform for facilitating collaborative code development for our 2D game Lost Isles. This allows each member of our team to have access to the shared code repository created by Sasha under the name 'SashaBekier/BITS-IslandGame.'

As an example, our team has used GitHub to act as a repository, therefore anytime a group member makes a change to the code, GitHub will be used to then create a new branch and then merge it to the main project.

By using GitHub, we can easily track changes made to the code, merge different versions of the code, and resolve any conflicts that may arise during development. This streamlines our development process and enables us to work more efficiently as a team.

3. Challenges and Learning

3.1. Group challenges

There were two main types of challenges that we faced as a group which were time management and skill management. Time management challenges include commitments with group member's personal life events, other studies, and work life. This would result in members being unable to attend meetings creating potholes in our individual and general understanding of the project, increasing the risk of miscommunication and disputes. This would also affect how much work is able to be done by members and create conflict with time availability to complete a certain section of the project. This can further lead to delay in the project completion if a section needed to be completed as a prerequisite of another section. This challenge was expected as you cannot expect members to not have any other commitments other than this project.

Skill management was a huge challenge for the majority of our group members. No members of the group before this project had any experience with the game development software Unity and thus we were all going in blind. This led to members having to self learn how to use Unity via online tutorials and documentation, which took significant time and effort. Skill gaps and varying levels of expertise also posed challenges during the development process, as some members struggled to grasp

certain concepts or implement specific features. This resulted in delays, rework, and a higher learning curve for the entire group.

3.2. How the challenges were addressed

Time Management:

Clear communication channels were established, and regular team meetings were scheduled to discuss project progress, clarify doubts, and address time availability issues. We prioritized tasks based on their importance and dependencies, ensuring critical milestones were given higher priority. Flexibility was incorporated into the project schedule to accommodate unexpected personal commitments, and team members provided support and assistance to help individuals catch up on missed work. These efforts resulted in effective time management, minimized delays, and improved project progress.

Skill Development:

We fostered a culture of collaborative learning and knowledge sharing within the team. Experienced members with prior Unity experience provided guidance and assistance to those new to the software, enabling everyone to learn from each other's expertise. Online resources, tutorials, and documentation were actively utilized, and members dedicated time to self-learning and exploring training courses to enhance their Unity skills. Tasks were assigned based on individual strengths and skills, allowing team members to leverage their expertise and expand their skill sets. Regular code reviews and feedback sessions were conducted to improve code quality and reduce skill gaps within the team. These efforts resulted in significant skill development and improved proficiency in Unity.

Ongoing Solutions:

While the project has been successfully completed, we recognize the importance of continuous skill development and improvement. To further address skill management challenges, we propose the following ongoing solutions:

Continued Learning Opportunities: We plan to organize periodic training sessions and workshops to enhance our knowledge and skills in Unity and related technologies. These learning opportunities will keep us updated with the latest advancements and industry best practices.

Mentoring Programs: Establishing mentoring programs within the team will provide a platform for experienced members to guide and mentor those who are new to Unity. This collaborative learning approach will facilitate continuous skill growth for all team members.

Collaboration with External Experts: We aim to collaborate with external experts or consultants in the field of Unity to gain insights, receive guidance, and further expand

our skill set. Their expertise will provide valuable inputs and help us address any remaining challenges.

3.3. Learning from the challenges

The design side of this project has pushed us all to improve our clarity and required us to learn about many new design tools. Additionally we have learned a lot about the general design process and the detail of game design.

We have all had to improve our team work, and group integration has steadily grown throughout the course of the development process.

On our team the only member with prior experience with C# was John and none of us had any prior experience with Unity. This presented a pretty steep learning curve and all of us have grown as programmers as a result. We have collectively learned a lot about using GitHub in a realistic way for team development and the project has pushed us to go beyond the basic usages that we had prior experience with in the Git system.

While none of us had prior Unity or game development experience we found that we were able to maximise our overall group understanding by each learning particular aspects and then sharing that knowledge among the team.

3.4. Changes

Over the course of the development we found that we had to greatly reduce the scope of what we were trying to achieve and focus on systems over assets. The lack of a graphic artist or designer on the team led to us using more off-the-shelf creative commons and open source assets than we had anticipated.

While we originally had bold plans to implement three strategically unique player classes, we underestimated the challenge of that task. In the end we were able to implement a flexible character system that has scope for later expansion into a distinct classing system. Similar reduction in scope occurred for all our MVF's as we reassessed our definition of minimum.

Towards the end of development we engaged in a less egalitarian team structure and found that we had improved outcomes and efficiency when tasks were allocated specifically and guidance was provided on how each task should be tackled. An increased focus on code review also assisted in the production and helped the team to improve their work and ensure good integration of the disparate systems.

3.5. Project plan refinements

With the benefit of hindsight we would have restricted the design process more and provided a harder end point on the design. We would have benefited by placing one individual in a decisive role during the design process and as a group we suffered for the highly democratic approach we chose to take.

We would have benefitted from closer oversight and if we were developing this plan again we would set clearer, more specific targets or checkpoints in order to improve accountability and help maintain focus.

A firmer grasp of the level of both learnings and work required to achieve outcomes in the game development process would have helped us to dream a little smaller and develop a more realistic expectation of what could be achieved in the available time.

3.6. Timeline refinements

The timeline has held up well throughout the development process with the exception of early delays in the design process. Everything in the timeline took more time than anticipated however team members largely stepped up their efforts to keep the actual process on track.

We have largely dealt with inadequate time by reducing the scope of the project or making additional hours beyond the initial expectation rather than making alterations to the timeline.

Additionally the timeline would have benefitted from a greater awareness of the interdependence of several of the design goals. Some rearrangement would be of benefit as several tasks had to be delayed as they required other tasks to be at least semi complete before they could be commenced.

In summary, the expected hours in our timeline would need to be roughly doubled to be realistic, however this time is largely driven by the degree to which the team needed to learn entirely new skills. Allocating more time in the early timeline to learning would have resulted in a more realistic timeline overall. If we were to tackle the timeline again we would restrict the design time significantly to allow for more time spent learning the technologies and experimenting with implementation in the Unity environment. Finally we would be more aware of necessary sequencing when putting together the timeline and given greater consideration to ensuring that underlying structures were in place earlier in the timeline.

3.7. Risks and unexpected events

Describe if the risks from Assessment 1 (Project Proposal) were actualized. Include any unexpected events and how they were addressed.

Risk 1) Misjudging the Project - The team certainly ran afoul of this risk, devising a plan that required a significantly larger time commitment than we had available. However the team discussion related mitigation strategies proved effective and we were largely able to wind back our over commitment to more manageable levels.

Risk 2) Scope Creep - We struggled with this throughout the development process and the mitigation strategies we had in place did not prove to be viable. While it seemed practical to maintain control over the scope of the project via the Trello system, in practice we did not utilise this tool to maximum effect. We would have benefited from a smaller control system with less moving parts and more direct

engagement from team members with the project management tool than we were able to muster.

Risk 3) Low Performance - There were moments when our project was challenged by performance issues and the very late development of the character system certainly put the combat system on the back foot, only able to truly be started late in the development cycle. The process of moving from broad conceptions of our product down to one specific implementation ended up taking far longer than would be ideal. Ultimately these risks were mitigated by a more direct and guided approach to task delegation with specific deadlines for specific requirements proving necessary for some team members.

Risk 4) Team Direction / Vision - We lost a couple of weeks of development time to this risk and with the benefit of hindsight we would have nominated a design decisive who's decision on design choices was final. While the group struggled to hear all members equally and work on a consensus basis, that strategy left us with a nebulous design that we could all agree on and a distinct lack of concrete design around which to develop the software itself. As team leader, I should have been more assertive, earlier in the design process.

Risk 5) Low Engagement - Our team had a wide range of engagement levels, mostly in direct alignment with the varying complexities of the team members lives. We should have been clearer in allocating high dependency tasks to the more engaged team members and development overall hit a few stagnant points while waiting for key pieces from the less engaged team members. That said no team member contributed less than the minimum requirements of the course (and project timeline).

Risk 6) Interpersonal Conflict - There were occasional minor tensions among the group but nothing that was allowed to escalate beyond some venting. The group as a whole did a good job of bringing matters of potential conflict to light before they grew into actual problems and various team members were able to mediate issues with satisfactory outcomes.

4. Marketing Pitch

Our 2D PC adventure-combat game is an exciting new take on the genre, offering players a unique mix of exploration, puzzle-solving, and thrilling turn-based combat. Set in a fantastical world filled with mystery, magic, and fantastical foes, players will take on the role of a skilled ranged or melee class adventurer tasked with saving the land from a powerful evil that threatens to destroy everything the island's civilization worked for.

With intuitive controls, stunning graphics, and a deep and immersive storyline, our game offers a PC gaming experience like no other. Whether you are a hardcore gamer looking for a new challenge or a casual player seeking fun and engaging adventures, our game has something for everyone. With a talented and experienced

team behind the project, we are confident in our ability to deliver a top-quality gaming experience that will captivate and thrill adult players. We are excited to bring our vision to life and believe that our game has the potential to become a major player in the gaming industry. Investors in the gaming industry are seeking innovative and profitable projects, and we believe that our 2D adventure-combat game has the potential to be both.

5. Skills and Jobs

Scrum Master - 2D Game Development

The Company

The Island Game is a small innovative video game startup that specialises in fantasy and role playing 2D games creating immersive gaming experiences for the global market.

We are seeking an experienced and passionate Scrum Master to help us develop our flagship product: The Island Game.

Job Description

As a Scrum Master you will oversee the development of our first game release: The Island Game. You will lead a team of developers, artists and designers, ensuring the successful implementation of agile methodologies throughout the project's lifecycle. Your primary focus is fostering a productive and collaborative work environment while promoting efficient and timely development practices.

Key Responsibilities

- Guide and support the development team using agile methodologies
- Facilitate sprint planning, daily stand-ups and sprint reviews to drive continuous improvement and well-organised delivery.
- Collaborate with product owners, stakeholders and team members to ensure a clean definition of project scope, task prioritisation and that aligns with the business objectives.
- Monitor and track project progress and resource allocation ensuring adherence to quality standards.
- Focus on open communication, knowledge sharing and constructive feedback within the team.
- Continuously identify and implement process improvements to optimise quality, productive and team dynamics.
- Stay up to date with industry trends, standards and best practices related to 2D game and Unity development

Skills & Experience

- Bachelors or Masters Degree in Information Technology or related industries
- Proven experience as a Scrum Master or Project Leader in game development, preferably 2D game development.
- Strong knowledge of agile frameworks like Scrum or Kanban, and experience implementing these in game development projects.
- High level leadership and communication skills, with the ability to motivate multi-function teams.
- Familiarity with Unity and its game development ecosystem.
- Passion for gaming and a keen eye for game design principles.
- Ability to manage multiple priorities, deadlines and stakeholders in a fast paced environment.
- Certifications like Certified Scrum Master or Professional Scrum Master are a plus

Senior Unity Developer

The Company

The Island Game is a small innovative video game startup that specialises in fantasy and role playing 2D games creating immersive gaming experiences for the global market.

We are seeking an experienced and passionate Unity developer with team-work experience to help us develop our flagship product: The Island Game.

Job Description

As a Senior Unity Developer you will be taking on the role of leading the software development

You will work closely with our design team and art team to help develop a game that is innovative and engaging for players worldwide.

Key Responsibilities

- Lead the development team for the games software components using unity and agile development methodologies.
- Function at a high level in a dynamic environment.
- Design and implement game mechanics, systems and features.
- Write clean, efficient and well documented code.
- Optimise the games performance on different platforms
- Troubleshoot and debug issues to maintain the games stability and functionality.
- Participate in code reviews and provide feedback to the team
- Stay current with industry trends to regularly improve the game and its development processes.

- Collaborate with the team to create a fun and immersive gaming experience for our players

Skills & Experience

- Bachelors or Masters Degree in Information Technology or related industries
- 5+ years of Software Engineering/ Development experience
- Knowledge of C# programming language
- Expert knowledge in Unity game engine development
- Understanding of 2D game development and best practices
- Ability to work in a team environment
- Experience with Agile

2D Game Artist

The Company

The Island Game is a small innovative video game startup that specialises in fantasy and role playing 2D games creating immersive gaming experiences for the global market.

We are seeking a skilled 2D Artist to join our team and help us develop our flagship product: The Island Game.

Job Description

Are you a talented and passionate artist with a passion for creating appealing designs and images? Do you dream of bringing a fantasy world alive through this passion and talent - then this is the job for you.

You will work closely with our development team to help design a game that is innovative and engaging for players worldwide.

Key Responsibilities

- Create high quality 2D artwork, concept designs for characters, environments, weapons and items, and user interfaces that align with the games fantasy/ island based theme
- Collaborate closely with the game development team to understand the game's vision and requirements and translate these into appealing game assets.
- Design and implement visual effects, animations and sprite sheets.
- Maintain a consistent art style, ensuring all assets align with the games artistic direction and meet the technical requirements.
- Continuous updates to designs based on feedback from the team, making adjustments and improvements based on this feedback.

Skills and Experience

- Proven experience in 2D Game design or similar role, with a focus on Fantasy theme
- Proficiency in industry standard tools, Photoshop, Illustrator or any other graphic design tools.
- Solid artistic and visual development skills, a key eye for detail - colour composition, lighting, etc.
- Excellent communication and collaboration skills, the ability and experience to work effectively within a team.
- Strong understanding of the game development process regarding asset design with technical techniques for creating 2D images and sprites.
- Ability to create original artwork from scratch and adapt styles while remaining consistent to a theme.
- Passion for gaming and an appreciation for the fantasy genre.
- A portfolio showcasing any previous artwork, concept designs, character designs, environment art or any animations/ visual effects.

Marketing Manager

The Company

We are a forward-thinking game development company dedicated to creating immersive and memorable gaming experiences. Our team comprises fresh and experienced creative and software development individuals who are driven by the same shared passion for gaming. We focus on creating a quality and innovative experience with high player engagement, to deliver exceptional games to our world wide playerbase.

Job Description

As Marketing Manager you will play a key role in shaping the success of our upcoming 2D fantasy game: The Island Game.

You will be responsible for developing and executing marketing strategies that drive awareness, engagement and bring on new players. Your ability to think creatively, strategically as well as adapt to changing market conditions will be vital to achieving our goals.

Key Responsibilities

- Develop and implement detailed marketing plans for our fantasy island game and refine tactics based on market feedback.
- Collaborate closely with multi-function teams including developers, artists and designers to align marketing initiatives with the game development milestones.
- Conduct market research and analyse player demographics, trends and competitors to identify opportunities and optimise marketing campaigns.
- Create marketing content, including game trailers, social media posts, and email campaigns, to excite and engage target audiences
- Manage social media platforms, community forums, Steam developer feedback portals and other online presences to foster an engaged player community

- Plan and execute user acquisition campaigns across various digital platforms, including social media, influencer partnerships and Steam store optimisation.
- Monitor performance KPIs and analyse these campaigns data to measure the effectiveness of the marketing strategies, providing insights for their continual improvement.
- Stay up to date with industry trends for marketing strategies and best practices

Skills & Experience

- Bachelor's degree in Marketing or Business related fields
- Experience as a marketing manager.
- Excellent written and verbal communication skills for creating engaging marketing material
- High analytical skills and ability to interpret data to drive decision making
- Passion for gaming, specifically in the fantasy genre with a keen understanding of players motivations and preferences
- Excellent communication and collaboration skills, the ability and experience to work effectively within a team.