

Team 13 - Kevin Nguyen, Nikhil Maraboyina, Ousama Batais, Noah Bowman, Nisai Sun

CS 4361.001 - Dr. Pushpa Kumar

5 December 2024

Stranded: Project Final Report

- **Project Source Code Repository**

<https://github.com/nxs-se2016/Team13-UnityGame>

- **Project Description**

A first-person POV game blending adventure, mystery, and puzzle genres, where the player finds themselves stranded on a bizarre island. This island defies the conventional rules of the world, suggesting magical powers linger in its air. To escape, you must solve six puzzles scattered across the island—from the peaks of hilltops to the sands where beach waves encroach. Upon completing each puzzle, a nearby chest takes flight, and you are rewarded with a wood plank. These planks are essential for building a boat to sail back to the lands you once knew!

- **Team Members and Contributions**

- **Kevin Nguyen**

- Designed the game map.
 - Implemented player movement mechanics (e.g., walk/run speeds, mouse tracking sensitivity).
 - Designed and implemented the Floral Cipher level.
 - Designed and coded the User Interface (UI), including:
 - Main Menu
 - Pause Menu
 - Settings Menu

- **Nikhil Maraboyina**

- Puzzle Designer and Programmer
 - Assisted the latter half of development's project coordination.
 - Solved issues pertaining to merging code from the sub-branches to the main branch.
 - Various puzzles throughout the island, such as the Box Color Puzzle, Totem Neighbor Puzzle, Gun Puzzle, Tree Puzzle, and others
 - Utilized Womp to design custom assets to be used within the puzzles.

- **Ousama Batais**

- Piano Puzzle
 - Primary programmer and designer
 - Animated pressure plates pressed and reset
 - Created custom scripts for playing samples of each pressure plate in a specific sequence.
 - Aligned and designed the pressure plates on a part of the island.

- **Noah Bowman**

- Code Chests
 - Coded functionality
 - Designed and created animations for each chest.

- Boat Repair
 - Coded functionality
 - Designed transition from “damaged” to “repaired”
 - Created ending title scene
- General Codebase Management
 - Pruned old unused code to maintain readability
 - Helped manage merging branches
 - Polished map and puzzle designs for the demo
- **Nisai Sun**
 - Project Coordinator
 - Handled the first half of development’s project coordination.
 - Promoted the idea of escaping stranded on an island.
 - Piano Puzzle
 - Proposed idea and directed segments to be placed.
 - Tested and gave feedback to Ousama Batais.
- **Tools and Software Used**
 - **Software Programs**
 - **Unity version 2022.3.45f1**: Engine tool used to build the game scenes, mechanics, and overall project framework.
 - **Womp**: 3-D design software, an alternative to Blender. Used to generate the trees seen in the Unity scenes.
 - **Blender**: Used to create custom 3D assets, such as trees shaped as numbers that align when viewed from specific angles.
 - **Programming Languages**
 - **C#**: Utilized for scripting, programming functions and methods, and implementing game logic.
 - **Asset Libraries**
 - **Unity Asset Store**: Purchased and downloaded additional 3D assets to enhance the environment.
 - **OpenAI's DALL·E 3**: Generated background images for our Menu panels.
 - **Others Software**
 - **GitHub**: Used for version control, collaboration, and project codebase management.
 - **Discord**: Facilitated team communication and coordination.
- **Design and Implementation Details**
 - We have multiple levels that employ the design principle of anamorphosis, where objects in the game environment take on meaningful forms only when viewed from specific angles (Genevois Franchi, 2019). A popular example of anamorphosis is the 2D chalk art by Julian Beaver, where distorted drawings on sidewalks appear as realistic three-dimensional scenes, like swamps or staircases when viewed from a specific angle (Wikipedia contributors, 2024). We incorporate the design principle of anamorphosis by designing levels where the end goal reveals a number code-like shape, which is then used to input into a chest to unlock a boat part.

Examples include:

 - **Tree Perspective (Figure 1)**: Arranged to appear as numbers when viewed from particular perspectives, encouraging spatial exploration.

- **Floral Cipher Level** (*Figure 2*): Flowers planted on a separate island to form visible numbers when observed from a high vantage point.
- Menu systems were done by having a camera fixed at a 2-D perspective, to where one of the dimensions is wholly locked until the player clicks on the **Play** button. Then, it unlocks and switches to the player's camera perspective.
- Puzzles are implemented by modifying the island's demo scene and using pre-existing open-use assets to implement puzzle design ideas that were recorded, edited, and iterated in a separate Google document.
- Several of the puzzles on the island relied on using the same set of scripts, namely a Pressure Plate and Pressure Plate Manager script. The pressure plate script in these puzzles was used on items with a colloidal body so that it would activate when specific items were placed on top of the pressure plate. The determination of what was an acceptable item to be placed on top of the plate was based on the object's name. The pressure plate manager script tracked which specific pressure plates were activated. When a specific set of items was activated, an action was taken, such as animating the item moving up from the ground. This system of pressure plates was used in puzzles such as the following:
 - **Box Color Matcher Puzzle** (*Figure 3*) - A series of five pressure plates lay on the ground, all with their own respective colors. On the other side of the island are five other boxes of the same five colors. The user must place the correct boxes on the corresponding pressure plates to raise the chest and solve the puzzle.
 - **Totem Neighbor Puzzle** (*Figure 4*) - A series of plates with five distinct colors lay on the ground. To the left are five pillars where the plates must be placed on. Next to the pillars is a flag describing a set of cryptic rules the user must use to solve the problem. The user must decode the message and determine which colors are neighbors to each other to decide on the correct order to solve the puzzle.
- In addition to the pressure plate-based puzzles, we also have gun-based puzzles (*Figure 6*). This puzzle follows the same structure as the pressure plate, which is a series of target scripts attached to objects that need to be activated and a pressure plate manager script to manage the activation of the targets. The difference relies on the gun's functionality used to activate the puzzle. When the user picks up the item, the user can shoot rays up to a thousand units away. If a target lies in the path of the ray, then the target is marked as "hit" and turned green. When enough targets are hit, the chest rises from the ground.
- Unfortunately, the **piano puzzle** (*Figure 6*) was not demonstrated due to conflicts merging with the main branch code. This piano puzzle contains several custom C# scripts to make pressure plate assets behave differently from other pressure plate puzzles throughout the island. First, the *pressure plate manager* is a listening event script that monitors to ensure the player is pressing the correct piano notes in order. If the sequence is incorrect, the player's viewport flashes red, and the rest of the plates reset back to null on the array to give the player another chance to solve the puzzle. Correct sequences mean a chest spawns from underneath the sand that contains a part of the boat to collect and repair the boat.
- User interaction is controlled by a tag in combination with an interactor placed at the front of the character model. Whenever this interactor collides with an object, it triggers a function that sees if the tag equals "interactable." If an object is interactable, a canvas will appear with a prompt to open or enter a code. For the code input, there is a script called CodeChest.cs for the code input, which handles everything for the chests. It uses the interact method to display a variable-length array of text inputs. On change, the inputs are checked against the answer which is listed in the inspector for the chest. If the answer is correct, it unlocks the chest and allows the user to open it.

- The game's objective is to escape the island by repairing the boat. While testing, we had the planks placed at the user's feet whenever the chest was open. For our demo build, we have the planks placed near the boat to save time. These planks are assigned a sub-mesh of the boat. By default, the boat is made of a translucent material (*Figure 7*), but if a plank comes in contact with the pressure plate at the base of the boat, a repair sequence is triggered, and the material transitions to wooden. Once all planks have been gathered and placed, an ending cutscene is triggered, and the user floats into the distance (*Figure 8*).

- **Screenshots**



Figure 1. Tree Puzzle as seen from top of mountain.



Figure 2. Floral Cipher Puzzle seen from vantage point.

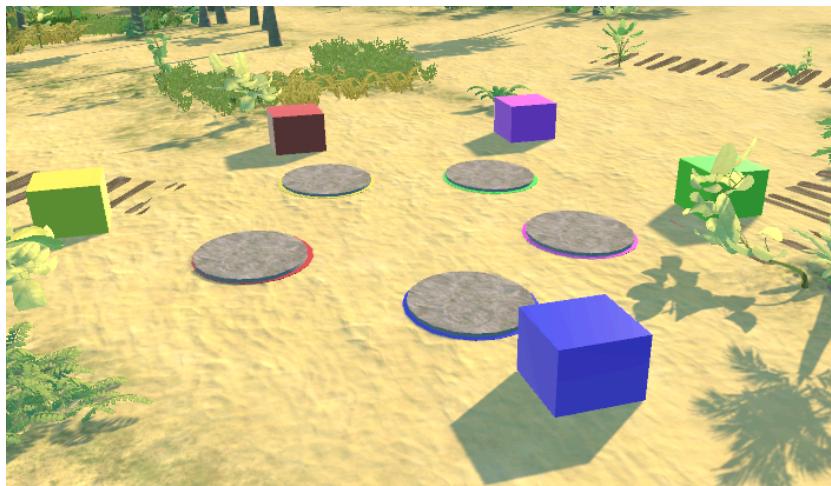


Figure 3. Box Color Matcher/Pentagram Puzzle



Figure 4. Totem Neighbor Puzzle



Figure 5. Target Puzzle pressure plates



Figure 6. Piano Puzzle



Figure 7. The player's boat outline and missing boards need to be found.



Figure 8. Halfway through the ending cutscene

- **Lessons Learnt**

- Project Management and Leadership
 - Lack of assertiveness did not get many plans being implemented in a timely manner.
- Merging branches for game engine projects

- .gitignore is not configured correctly on all branches, conflicts attempting to merge sub-branches to the main branches.
- This led to merging code the old-fashioned way, before version control was adopted into the mainstream.
 - Upon further review and research,
- Communication
 - Sometimes, outside of fixed scheduled meetups, it was challenging to get ahold of team members.
- Keeping Scope in Check
 - Keen observation of this project was the breadth and sheer ambition of this project, which seemed a lot to do in a span of only one university semester, balancing it with personal and professional obligations outside of educational commitments.
 - Making sure priorities are done in a timely manner before scaling up ambition should have been done to prevent scope creep.

- **Possible Future Improvements**

- Further testing feedback towards bugs, glitches, and exploits that may hinder the player experience for many. For example, the user can skip over large pieces of the tree puzzle by climbing up the puzzle from the other side of the mountain. Developing unreachable walls or respawn points can prevent cheating scenarios. As another example, in the Box Color Matcher Puzzle, when an item is placed on top of a plate, the plate seems to glitch, and the box doesn't seem to stick on. Solving glitches like this will help raise the overall engagement of the game.
- Creating more custom assets will give the game a more unique feel. Given our experience and timeframe, we opted to use mostly premade assets from the Unity Asset Store.
- Adding music, sound effects, and separate audio settings for the users' choosing and liking. There are currently sound effects in the Piano Puzzle, as specified above, as it is intended to develop the puzzle. However, this could be expanded upon.
- The user interface is rather barebones, but to a fault, as giving visual feedback to the user is essential to understand what is currently happening, and this build fails to inform the player.
- Providing a WebGL build of the project for team members' portfolio websites (ex., Github Pages, Weebly, WordPress, etc.) and source code to clone for their personal/professional git repository.
- Solving the rendering of some of the objects. Some objects, like the slingshot, were not adequately rendered as they were merged between branches. Fixing the rendering issues will help retain the user experience.
- Optimizing rendering and scenery to make it easier and less computationally expensive to run.
- Completely merge and integrate the Piano Puzzle into the rest of the game.

- **Conclusion**

As a team, we were extremely satisfied with the body of work that we produced. We came into this project with little to no game development experience and had an ambitious idea for our project. Despite this seeming like a recipe for disaster, our team came together. We used online resources (tutorials, asset packs, etc.) and lessons from class to attain the knowledge and tools necessary quickly. Our team did a good job overall of dividing work over the semester, alleviating some pressure as the semester ended.

Meeting weekly (sometimes even twice a week) was a massive part of our team progressing smoothly. We held each other accountable and kept the project flowing. In the end, our game had five puzzles on the main branch, which were presented, and another puzzle we completed but could not merge without conflicts. This exceeded our project goals and created an engaging and satisfying gameplay cycle.

- **References**

- Phideas. Stylized Tropical Island [3D asset]. Unity Asset Store, 2023. Available at: <https://assetstore.unity.com/packages/3d/environments/landscapes/stylized-tropical-island-264370>
- Monqo Studios. Responsive Pressure Plates [Tool asset]. Unity Asset Store, 2023. Available at: <https://assetstore.unity.com/packages/tools/input-management/responsive-pressure-plates-190860>
- Genevois Franchi, Thibaud. Optical illusion (or anamorphosis): the latest trend in graphic design. Pixartprinting Blog, 2019. Available at: <https://www.pixartprinting.co.uk/blog/optical-illusion-anamorphosis/>
- Wikipedia contributors. Julian Beever. *Wikipedia*, The Free Encyclopedia, last modified April 30, 2024. Available at: https://en.wikipedia.org/wiki/Julian_Beever
- Unity Real-Time Development Platform. 3D, 2D, VR & AR Engine. Unity Technologies, 2024. Available at: <https://unity.com/>
- Free 3D design software - Womp. Womp 3D Inc., 2024. Available at: <https://womp.com/index>
- Discord - Group Chat That's All Fun & Games. Discord Inc., 2024. Available at: <https://discord.com/>
- DALL-E 3 | OpenAI. OpenAI, 2024. Available at: <https://openai.com/index/dall-e-3/>
- GitHub. GitHub Inc., 2024. Available at: <https://github.com>