Team 5 - Capstone Game Dev

Island Song: Preliminary Report

Fall 2025 – Spring 2026

Revised: 10 October 2025

Logan Faulstich

Steven LaGoy

Zach McGill

Harrison Niswander

Ricardo Saldana

Riley Tate

Table of Contents

Introduction

Island Song is a single-player puzzle and exploration game with an emphasis on music and color. Players will solve music- and sound-based puzzles themed around instrument families, bringing a mysterious and enchanted island back to life by spreading music and color. This game will appeal to players who enjoy living worlds, discovery, and a playful atmosphere with auditory and visual experiences.

Definitions

POI – Point of Interest, any location designed with a certain experience or user interaction in mind.

UI – User Interface, the buttons, prompts, text, et cetera which users interact with most immediately.

HUD – Heads-Up Display, a persistent element of the UI which communicates details about the in-game player avatar or world to users.

FPS – Frames Per Second, a performance metric for the speed at which a game is running.

VCS – Version Control System, used to manage shared code.

MVP – Minimum Viable Product, the expectation for any subset of the game which sufficiently fulfills necessary functional and nonfunctional requirements.

Unreal Engine – An industry-standard game engine operated by Epic Games. Specifically, this project will be built with Unreal Engine 5.

Product Vision

Problem Statement

Many modern games offer expansive, realistic worlds which showcase technical achievement but compromise on the charm of a curated, whimsical world. With this game, the team aims to explore the full potential of music and color in video storytelling. An emotional world will connect with many players and allow them to explore their own capacity for art as they restore life to a quiet world.

Stakeholder descriptions

Key Stakeholders

Project sponsor – Aga Pito Studios, LLC; Jared Lehman, Esq.

Faculty Advisor – Prof. Thomas Bolinger

Development Team:

- Steven LaGoy: Team lead, World designer, Composer
- Zach McGill: Project manager, Lead architect
- Harrison Niswander: Treasurer, Art & Sound designer
- Riley Tate: Recordkeeper, Puzzle designer
- Logan Faulstich: UI designer
- Ricardo Saldana: Story designer

Creative Collaborators:

At various stages of the project, students or freelancers in art or music may be commissioned for certain assets, sound elements, or musical performances.

* Listed roles are tentative and do not indicate the full extent of a team member's roles and contributions.

User summary

Target Audience: People who are familiar with casual video games, looking for a slow-paced immersive story, exploration, and an outlet for creative thought. ("Players")

Play Testers: Ideally, play testers are members of the target audience. Play testers will be asked to play a version of the game for some period of time, and their gameplay will be recorded for iterative analysis to improve game systems.

User environment

Target Platform:

Specification	Minimum	Recommended
OS	Windows 7 64-bit	Windows 11 64-bit
Processor	Intel Core i5	Intel Core i5
Memory	4 GB RAM	8 GB RAM
Graphics	Intel HD 630	GTX 750 Ti
Storage	10 GB available space	10 GB available space

Minimum and recommended specifications are likely to change throughout development.

Operating environment

Game Engine: Unreal Engine 5

Programming Language: C++, Blueprints Visual Scripting

IDE: Microsoft Visual Studio

Version Control: Perforce p4

Communication: Microsoft Teams

Documentation: Microsoft SharePoint

Project Management: Jira

User stories

As a sponsor, I want to provide a challenging but approachable learning experience.

As a sponsor, I have a minimum viable product in mind and want to see it executed.

As a developer, I want to create a fun, cohesive, and complete game based on our scope.

As a developer, I want to take a game idea through the project development life cycle.

As an artist, I want the opportunity to create art that connects with the game's tone.

As a player, I want to understand my goals and motivations within the game.

As a player, I want a relaxing but immersive gameplay experience.

As a gameplay tester, I want to share my thoughts on what could make the game better.

Product Overview

Gameplay

Core gameplay loop:

The primary game loop consists of exploration of an area of the game world, finding, understanding, and completing an interactive puzzle, restoring color to an area of the map, and progressing to newly-opened areas.

Pillars of experience:

The pillars of gameplay experience define the most important factors in communicating the team's vision to players. These factors will receive the most focus when applying unique themes to areas of the map, gameplay elements, et cetera.

- Exploration
- Puzzles
- Music
- Color

Major features

These features are explored in terms of product goals in the Requirements Model section. They realize the Pillars of Experience by enumerating key details of player interaction with the game world, story, and mechanics.

- Player movement
- Non-linear island exploration
- Fixed map arrangement
- Unlockable areas
- Area theming
- Musically and visually integrated puzzle design
- Functional UI / HUD
- Multi-stage puzzle design
- Ambient and area-specific soundtracks
- Music cues and theming
- Color restoration mechanics
- Low-poly art style

Requirements Model

Functional Requirements

Functional requirements define the features and behaviors of the game.

- The game must include a main menu screen
    - Creative game logo design and team name / logo
    - “Start Game” should load the most recently played saved game
    - “Load Game” should allow selection of any saved game
    - “Create New Game” should create a new game and start the story
    - “Settings” should allow players to change their game settings
- During gameplay, players should be able to save their progress, adjust their settings, or leave the game to the menu or to their desktop
    - Settings should include audio volume, FOV, graphics quality
- The game should include several interactive, challenging, and unique puzzles
- Progress should be communicated to the player though an inventory, unlocking areas of the map, story progress, and other means
- Visual/Audio style should impart a unique and immersive look to the game
- The map must consist of 5 unique areas with 3 unique puzzles; completion of the game involves completing all implemented puzzles
- Map and area boundaries should be communicated clearly to the player
- Movement should involve using the mouse and WASD keys (or other key remaps)
- Completing the game should result in some amount of end-game content (special cutscene, customizable, or other unlock)

Non-functional Requirements

Non-functional requirements include qualities and aspects of the game.

- No more than 30 seconds should elapse between launching the game and being presented with the menu screen
- Loading a previously saved game from the menu should take less than 20 seconds
- Players should not experience a loading screen while in the game world
- Performance on target systems with recommended settings should remain at or above 60 FPS
- Noticeable "lag spikes" should be minimized between areas
- 1 hour of continuous, regular gameplay should be possible without experiencing game crashes
- Responsive elements of the world (puzzle elements, buttons, doors, characters) should visually respond to proper input within 0.5 seconds of interaction

Accessibility

To promote accessibility for many kinds of people, efforts will be made to provide settings or accommodations for people with limited mobility, sight, or hearing. This may include: colorblindness settings, high-contrast filters, audio narration, on-screen audio prompts / closed captions, support for multiple input devices, remappable controls, adjustable text size, adjustable control sensitivity, inverted view axes, in-game reminders for controls, and/or resizable UI. In the design of game elements and puzzles, thought toward accessibility will inform decisions, including minimizing quickly changing or flashing lights and colors, allowing players to choose their own pace for dialog and navigation, clarity and consistency in game text and descriptions, and clearly marking interactable objects visually.

Project Plan

Project management

The ~32 weeks during which this game will be developed are divided into 2 semesters and 4 quarters. Quarter 1 (Q1) and quarter 2 (Q2) fall during the Fall semester, with the end of Q1 corresponding to Fall midterms (17 October 2025) and the end of Q2 corresponding to Fall finals (14 Dec 2025). Quarter 3 (Q3) and quarter 4 (Q4) fall during the Spring semester, with the end of Q3 corresponding to Spring midterms (8 March 2026) and the end of Q4 corresponding to Spring finals (3 May 2026).

During each quarter, around four 2-week sprints are planned. Each sprint will begin with a Sprint Planning meeting to create, assign, and refine tasks for the next 2-week period. In the middle of every week which is part of a sprint, a Scrum meeting will allow team members to share their progress, blockers, and goals with each other and with the faculty advisor.

Deliverables

Each quarter contains an important deliverable goal.

Quarter	Deliverable
1	Game map, player movement demo, game story bible
2	MVP for 1 interactive puzzle, color mechanic demo
3	MVP for 3 interactive puzzles, finalized music & assets
4	Testing reports, completed game, hosted installer

Q1

Game map: Complete an explorable iteration of the game map, with some assets populating 6 distinct areas.

Player movement demo: Demonstrate player movement, including camera movement, walking, running, jumping.

Game story bible: Document in a "story bible" the story of the game to serve as a guide and inspiration for story and thematic elements.

Q2

MVP for 1 interactive puzzle: Develop and test one interactive puzzle for one of the 5 themed areas. Players should be able to understand, interpret, and solve a challenge which connects to the game world and an instrument family theme.

Color mechanic demo: Demonstrate the color mechanic, including the player spreading color in a radius around them, in footsteps, and by interacting with the world (I.E. completing a puzzle).

Q3

MVP for 3 interactive puzzles: As in Q2, develop and test a total of at least 3 interactive puzzles in 3 different themed areas (including the puzzle from Q2).

Finalized music & assets: Have the final assets and music placed properly into the game world. This includes soundtrack, audio elements, world features, UI elements, and game art.

Q4

Testing reports: Conduct user testing and prepare a report on how user gameplay feedback informed development decisions and feature iterations.

Completed game: Deliver a MVP for the game as a whole, fulfilling all outlined necessary requirements and key user stories.

Hosted installer: Provide an installer for the game possibly hosted on a site like Steam or Itch.io. Provide for continued personal or institutional access to the game for a minimum number of years following Spring 2025.

Timeline

Quarter #	Sprint #	Sprint Start	Sprint End
1	0	22 Aug 2025	07 Sep 2025
1	1	08 Sep 2025	21 Sep 2025
1	2	22 Sep 2025	05 Oct 2025
1	3	06 Oct 2025	17 Oct 2025
FALL RECESS	FALL RECESS	18 Oct 2025	21 Oct 2025
2	4	22 Oct 2025	02 Nov 2025
2	5	03 Nov 2025	16 Nov 2025
2	6	17 Nov 2025	25 Nov 2025
THANKSGIVING	THANKSGIVING	26 Nov 2025	30 Nov 2025
2	7	01 Dec 2025	14 Dec 2025
FALL FINALS	FALL FINALS	15 Dec 2025	21 Dec 2025
WINTER BREAK	WINTER BREAK	22 Dec 2025	11 Jan 2026
3	8	12 Jan 2026	25 Jan 2026
3	9	26 Jan 2026	08 Feb 2026
3	10	09 Feb 2026	22 Feb 2026
3	11	23 Feb 2026	08 Mar 2026
SPRING RECESS	SPRING RECESS	09 Mar 2026	15 Mar 2026
4	12	16 Mar 2026	29 Mar 2026
4	13	30 Mar 2026	12 Apr 2026
4	14	13 Apr 2026	26 Apr 2026
4	15	27 Apr 2026	03 May 2026
SPRING FINALS	SPRING FINALS	04 May 2026	10 May 2026

Risk Assessment

Scope assessment

This project will span two semesters between Fall 2025 and Spring 2026, a total of about 32 weeks. This involves designing, developing, testing, and revising an idea from concept start to end product. It will be necessary to continuously evaluate the scope of the project to ensure that it can be realistically developed by this team in this span of time. The risk of the project becoming too expansive in its goals or features is called "scope creep" and must be avoided by deliberately focusing on the minimum viable product (MVP). The goal of the team and its leadership is to promise little and deliver much, for the benefit of stakeholders and the team's own well-being and educational opportunities. Mitigation will also involve comparing the project's current stage of development to an expected timeline of deliverables based on research in similar game projects and guided by experienced advisors and stakeholders. The team's "velocity" will determine which of the stretch features will be feasible while meeting applicable deadlines.

Development priority

As this game idea contains a heavy emphasis on music arts, audio elements and the soundtrack will necessarily require extra time and effort compared to other game projects with similar scopes. As the primary focus of the project itself is in Computer Science, and as the team members are more experienced with and oriented towards other aspects of game development, it is paramount that the effort spent towards writing and performing music for the game does not overshadow the development of game functionality. To mitigate this risk, other choices about the art direction were made, including selecting a "low-poly" art style rather than a more realistic, art-intensive style. Additionally, collaborators outside of the CS capstone with an interest in music will be commissioned some work on creating art and assets for the game world, rather than team members creating those assets themselves.

Data management

Managing internal project data is an important factor in the development of a game project. Not only will the team regularly access and modify large files containing code and asset information, which itself requires special technology and approaches to manage, but some data may be necessary to keep secure or confidential. As for the first of these concerns, the VCS provides a solution. Using a specially licensed form of Perforce P4 Version Control, developers will be able to reliably share large files, with features like locking, stream management, and rollback to minimize file conflicts and breaking changes. The second concern is most applicable to the testing phase of development, where gameplay testers will voluntarily provide their time and information to play the game and give feedback. Keeping this data secure and managing potentially sensitive information must be considered before starting user testing. Accepted best practices for data security in HCI studies will inform the team at this stage.

Learning approach

As team members are primarily learners, the foremost goal of this project is to provide experience and education to each team member. It is certain that each member of the team will be using technologies they have never used before or using familiar technologies in new ways. There is a risk that these tools prove too daunting to learn in the timespan afforded by this project. To mitigate this risk, team members will, as much as possible, share their learning experiences with other members of the group to help all members learn from one another. Existing tutorials, guides, and documentation will prove indispensable for the learning process. Learning will be prioritized before development tasks, so that team members can overcome learning challenges before implementing new features.