4.0 Software Development Plan

4.1 Plan Introduction

Akemi Isles is a 2D narrative-driven puzzle adventure game inspired by *Poptropica* and mythological stories. The player explores themed islands, solves puzzles, collects items, and interacts with NPCs to progress through the story. The project's goal is to develop one simple, fully functional island as a vertical slice demonstrating core gameplay systems: movement, inventory, dialogue, puzzles, and UI.

Development activities will include:

- Designing and implementing gameplay systems in Unity 2D using C#
- Creating and integrating custom art assets
- Testing gameplay responsiveness and stability
- Delivering a playable prototype and final presentation by semester end

4.1.1 Project Deliverables

- Project Proposal Document
  This document provides an overview of Akemi Isles, including a detailed description of the game concept and the motivation behind its development. It also outlines the tools and technologies required to build the game and presents a clear justification for undertaking the project.
- Requirements Specification Document
  This document defines the specific requirements that Akemi Isles aims to fulfill. It includes functional elements such as the game manager, player mechanics, and puzzle systems. The Requirements Specification serves to capture the low-level technical and functional requirements of the project and establishes a formal understanding between the instructor and the student team regarding the expected outcomes of the completed project.
- Initial Development Schedule
  This deliverable outlines the planned development timeline in an Agile format, identifying major milestones, sprints, and deliverables to ensure steady progress throughout the project lifecycle.
- Written Status Reports (starting from week 6)
  These are recurring written reports submitted weekly beginning in Week 6. Each report will describe the current progress of the project, summarize completed tasks, outline upcoming goals, and note any challenges or adjustments made to the development plan.
- Software Development Plan Document
  The Software Development Plan (SDP) describes the overall development process that will be followed throughout the semester. It specifies the methodologies, tools, and

quality assurance practices that will guide the production of all required documents and the final software product.

- Software Design Description Document
  The Software Design Description (SDD) presents both the architectural and detailed design of the game. The architectural section defines the high-level structure of the system, including its components and internal interfaces. The detailed design section follows the same organization as the architectural section but provides complete technical details, including specific implementations and design decisions.
- ALPHA/BETA/Critical Design Review
  These reviews will serve as major project milestones used to evaluate the progress and quality of Akemi Isles throughout development. The Alpha Review will assess the initial implementation of core mechanics and game systems. The Beta Review will focus on the completion of gameplay features and overall functionality, while the Critical Design Review (CDR) will evaluate the final design, performance, and readiness of the project for release.
- Presentation / Demonstration
  This deliverable involves a formal presentation and live demonstration of Akemi Isles to showcase the game's features, design, and functionality. The team will explain the development process, demonstrate gameplay, and discuss how the final product meets the established requirements and project objectives.
- User's Manual
  The User's Manual will serve as a comprehensive guide to help players understand and navigate Akemi Isles. It will include instructions on installation, controls, gameplay mechanics, objectives, and troubleshooting, ensuring that users can fully enjoy and operate the game.
- Final Product Delivery (Final Report and Code)
  The final deliverable will consist of the completed Akemi Isles game, accompanied by the Final Report and source code. The Final Report will summarize the entire development process, including design, implementation, testing, and evaluation, demonstrating how the project meets its initial goals and specifications.

4.2 Project Resources

In this project, three categories of resources will be utilized: human, hardware, and software resources.

Regarding human resources, the project team consists of three members with distinct roles and responsibilities. Jody Jasim will serve as the Scene Designer and Narrative Lead, Taylor Musso will assume the role of Art Director and Systems/UI Designer, and Matthew De Jesus will act as the Programmer and Systems Designer.

For hardware resources, the primary development device will be Matthew's Dell XPS laptop, which will be used for programming, testing, and integrating various project components.

Lastly, the software resources will include Unity 2D as the main game engine utilized for the design, development, and implementation of the project.

4.2.1 Hardware Resources

Category	Requirement
Processor	Intel Core i5 (minimum) / Intel Core i9-12900HK (development machine)
RAM	8 GB (minimum), 32 GB (development machine)
Hard Drive Space	2 GB free space for installation and assets
Display	1920×1080 resolution, integrated or discrete GPU support
Graphics	Integrated graphics (minimum), NVIDIA GeForce RTX 3050 Ti (development GPU)
Input Devices	Standard keyboard, mouse, optional gamepad support

The development machine used is an XPS 15 9520 (Intel i9-12900HK, RTX 3050 Ti, 32 GB RAM). Minimum specifications ensure broad accessibility for standard laptops/PCs.

4.2.2 Software Resources

Category	Requirement
Operating System	Windows 10/11 (64-bit) or macOS Monterey (or later)
Game Engine	Unity 2D (6000.0.31f1)
Programming	C# (within Unity scripting environment)
Version Control	GitHub for collaboration and source control as well as BOX to save weekly progress as insurance
Art Tools	Piskel (sprite creation), Adobe Photoshop or GIMP (optional for assets)

IDE/Editor	Visual Studio Code / Visual Studio Community Edition

Unity 2D with C# will be the primary development environment. GitHub ensures collaborative version control, with BOX as additional insurance.

4.3 Project Organization

Role / Team Member	Responsibilities
Jody Jasim – Scene Designer & Narrative Lead	Sets up Unity scenes and manages story scripting. Assists with narrative design and puzzle interaction logic to ensure story flow aligns with gameplay progression.
Taylor Musso – Art Director & Systems and UI Designer	Creates concept art and in-game assets. Designs and implements the dialogue system, oversees UI design and integration, and maintains visual consistency across menus and gameplay.
Matthew De Jesus – Programmer & Systems Designer	Implements core gameplay systems such as player movement, game manager, and inventory system. Supports narrative integration, puzzle logic, and overall system coordination.

Communication & Workflow

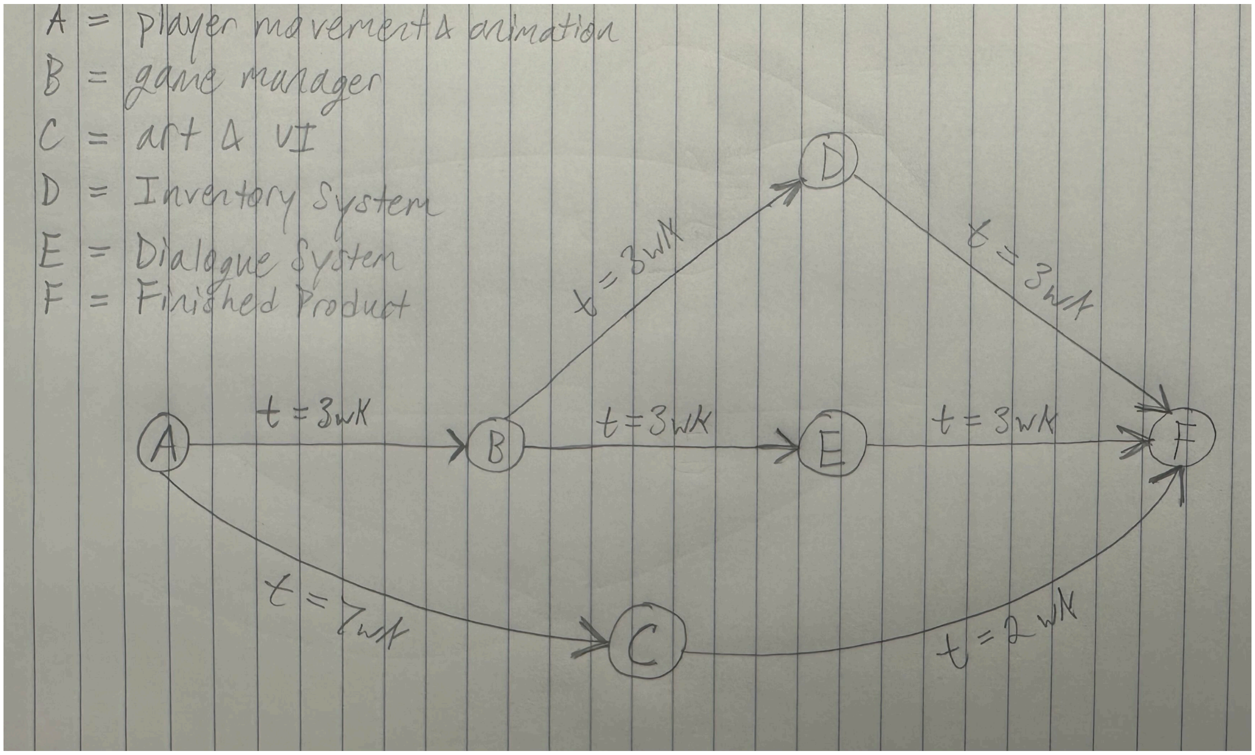
- Weekly team check-ins to review milestones and progress
- Version control and collaboration managed through GitHub
- Asset sharing and version backups maintained via BOX
- Scene-based task division in Unity to prevent merge conflicts

4.4 Project Schedule

Weeks	Tasks

5-7: September 21 - October 11	Player movement and animations + debugging
8-10: October 12 - November 1	Setup Game Manager, Inventory System, Dialogue System + debugging
11-12: November 2 - November 15	Implement Start Menu, Island Selection, and Puzzle (one scene)
13-14: November 16 - November 29	Debug, Test, art/UI polish
15: November 30 - December 6	Final Build + Presentation

4.4.1 PERT / GANTT Chart



4.4.2 Task / Resource Table

The following table outlines the primary development tasks for Akemi Isles, along with their assigned personnel, hardware, and software resources, in order to identify dependencies between tasks and balance tasks between group members.

Task	Assigned Member(s)	Hardware Resources	Software Resources	Description / Notes
Concept Art & Asset Creation	Taylor Musso	Drawing Tablet and Laptop	Piskel, Photoshop	Assets must be exported and optimized for Unity
Player Movement System	Matthew De Jesus	Windows Laptop (XPS15)	Unity 2D, C#, VS Code, BOX	Forms the foundation for interaction and gameplay
Game Manager Implementation	Matthew De Jesus	Windows Laptop (XPS15)	Unity 2D, C#, VS Code, BOX	Handles game state, save/load, and transitions
Dialogue System Integration	Taylor Musso	Macbook	Unity 2D, Visual Scripting Tools (e.g., Fungus or custom nodes)	Integrated with story scripting and UI
UI Design and Integration	Taylor Musso	Macbook	Unity UI Toolkit, Photoshop	Depends on finalized art assets and dialogue data
Inventory System Development	Matthew De Jesus	Windows Laptop (XPS15)	Unity 2D, C#, VS Code, BOX	Connects with Game Manager, Player System, and Dialogue System
Puzzle Interaction Design	Jody Jasim	Macbook Pro m1 2021	Unity 2D, C#, VS Code, BOX	Dependent on player and inventory systems for item-based puzzles
Scene Setup & Story Scripting	Jody Jasim	Macbook Pro m1 2021	Unity 2D, C#, VS Code, BOX	Requires finalized

				dialogue assets from Taylor for narrative integration
Build Testing & Debugging	Jody Jasim	Macbook Pro m1 2021	Unity 2D, BOX	Conducted multiple times a week to detect bugs in every scene; involves joint playtesting
Final Integration & Polishing	All members	Windows Laptop (XPS15) and Macbooks	Unity 2D, BOX, Github	Final asset merge, performance optimization, and demo preparation