# The Half Dozen

# A Dark Moon
# Software Architecture Document

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 23/10/22 | 1.0 | First Draft | Harvey Ji |
| | | | |
| | | | |
| | | | |

# Table of Contents

# Software Architecture Document

## 1. Introduction

### 1.1 Purpose

This document provides an architectural overview of the system to show the architectural decisions that were made for A Dark Moon.

### 1.2 Scope

This Software Architecture Document provides an architectural overview of A Dark Moon.

### 1.3 Definitions, Acronyms, and Abbreviations

**HTML**

**Hypertext Markup Language**

**CSS**

**Cascading Style Sheets**

### 1.4 References

**A Dark Room: https://github.com/doublespeakgames/adarkroom**

## 2. Architectural Representation

This document presents the architecture by presenting a series of views. The use case view and the logical view are presented. Both of these views use the Unified Modeling Language (UML).

## 3. Architectural Goals and Constraints

A Dark Moon is a game that will run independent of internet connection once an initial connection has been made. All data will be kept on the user's computer. The game consists of three major components. The resources tab, upgrades tab, and exploration tab.

All tabs must execute on the user's computer and will be designed in JavaScript and HTML.

The three tabs will need to share data between each other such as resources and special items.

There should be no communication outside of the three tabs.

## 4. Use-Case View

Critical functionality that comes from the use-cases include. The player wants to progress by gathering and spending resources. The player wants to have more efficient resource collection and management, through the means of auto collection. The player has enough self-sufficient resource collection that they can devote their attention to the exploration screen.

### 4.1 Use-Case Realizations

Resource collection builds on itself and grows exponentially in how they are gathered and spent by the player. For instance, the player starts by manually harvesting them through buttons that give a specific amount of the desired resource. The player can then upgrade their tools, allowing for more resources to be harvested. At the same time the player could build housing for villagers. Villagers can then be allocated into harvesting resources automatically for the player, but starts out ineffective. This can be then improved through various means of improving the villagers tools or building more housing for more villagers and by doing *both* the efficacy then out produces the player who no longer has reason to harvest resources themselves and then can unlock the explore screen and start devoting resources into exploration.

# 5. Logical View

A Dark Moon architecturally will have multiple parts that make up the design model

The Gather Class: In This class the user will be able to gather materials and possibly be able to build tools that can be used for survival, part of the subclasses will be part of this will be:

Village Gather: As the number of villagers increase the user will have the option to have the village to the gathering for its own survivability

Shop: As the user gathers enough  materials, they will be able to create a shop at which they can trade utilities and tools that will help better the survival for them and the village. This is where the user will be able to purchase a NAV CPU to unlock the map.

The Explore Class: This class will only be unlocked after the user purchases the NAV CPU. some of the Subclasses will be.

Combat: This subclass will define how the user will be able to engage in combat during the explore phase

Endgame: This is the end subclass that will decide the end of the game.

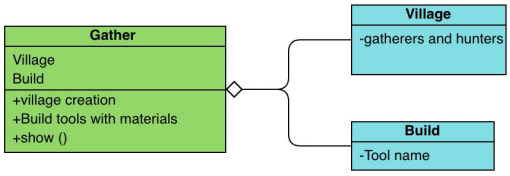
## 5.1 Overview

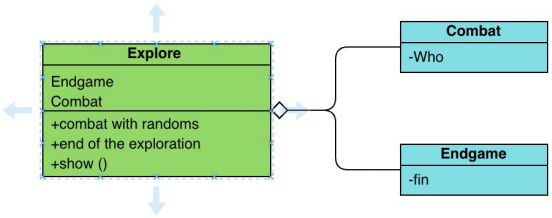
This subsection describes visually the overall decomposition of the design model in terms of its package hierarchy and layers.

## 5.2 Architecturally Significant Design Packages

Explore Package: This package will consist of the exploration of the map and the combat that will lead to the endgame of the player's journey.

Gather Package: This is the beginning of the journey; it will consist of the user gathering materials that will then be used to create tools that can be used for the player's survival. This will also include the growth of the village and the creation of the shop which will be used to trade items and be the way to unlock the exploration mode of the game.





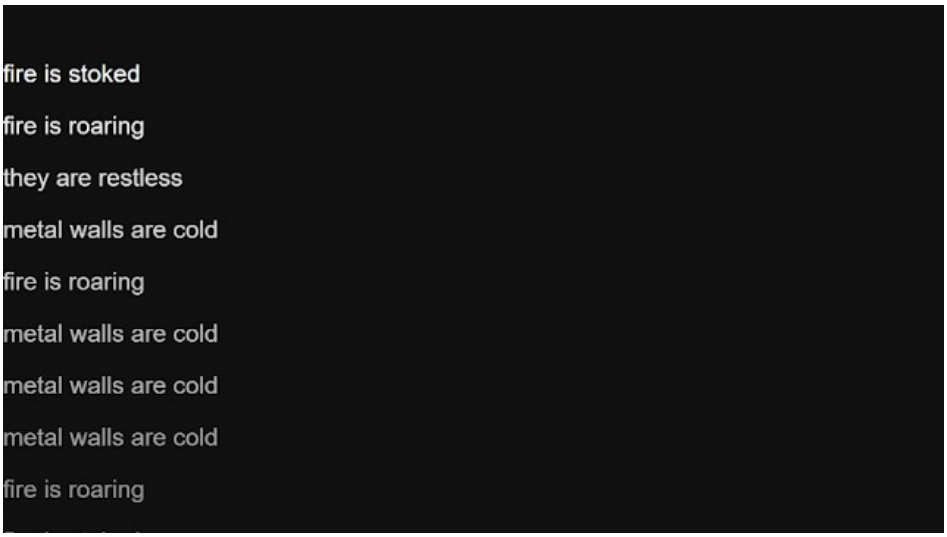

## 6. Interface Description

The user interface is a two part system. The first part, the entire left side of the screen, is used to display rolling lines of text. Each new line paints a word picture of the player character's story and actions as they survive, gather resources, build, fight, and explore. This text is the player's only window to the game world.

The second part, the entire right side of the screen, is how the player character interacts with the game world. There will be various buttons with labels for ways that the player character can interact with the game world. Each button will cause new text to appear on the left side as the consequences of their choices unfold. On this side there will also be choices for "rooms". Each of the "rooms" hold a unique selection of buttons for the user to interact with. For example the first "room" will carry all buttons for the player character to interact with the world. The second "room" will carry all the buttons for the player character to interact with the village and its people.

There will be no invalid input as the only input will be provided buttons.

This is an example of what the interface will look like:



## 7. Size and Performance

The implementation of the selected architecture supports the sizing and timing requirements of The Dark Moon. The components are designed to use minimal disk and memory space on a client's PC.

## 8. Quality

The software architecture supports the quality requirements, as stipulated in the Software Requirements Specification and Supplementary Specification.