



### **Activity: Architectural Design Process – Iteration 2**

Breakout room 11; November 17, 2021

Members Names	Student ID
Raphael Halim	100700318
William Robinson	100751756
Liam Rea	100743012
Phillip Jasonwiki	100751888

## Step 2: Establish Iteration Goal by Selecting Drivers

In this second iteration, the architect considers the system's primary use-cases:

- UC-1
- UC-2
- UC-5

## Step 3: Choose One or More Elements of the System to Refine

The Elements we will refine this iteration are the ones related to combat and its mechanics. In general, the game will rely on these modules the most, so they must be working before anything else can be added on.

## Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

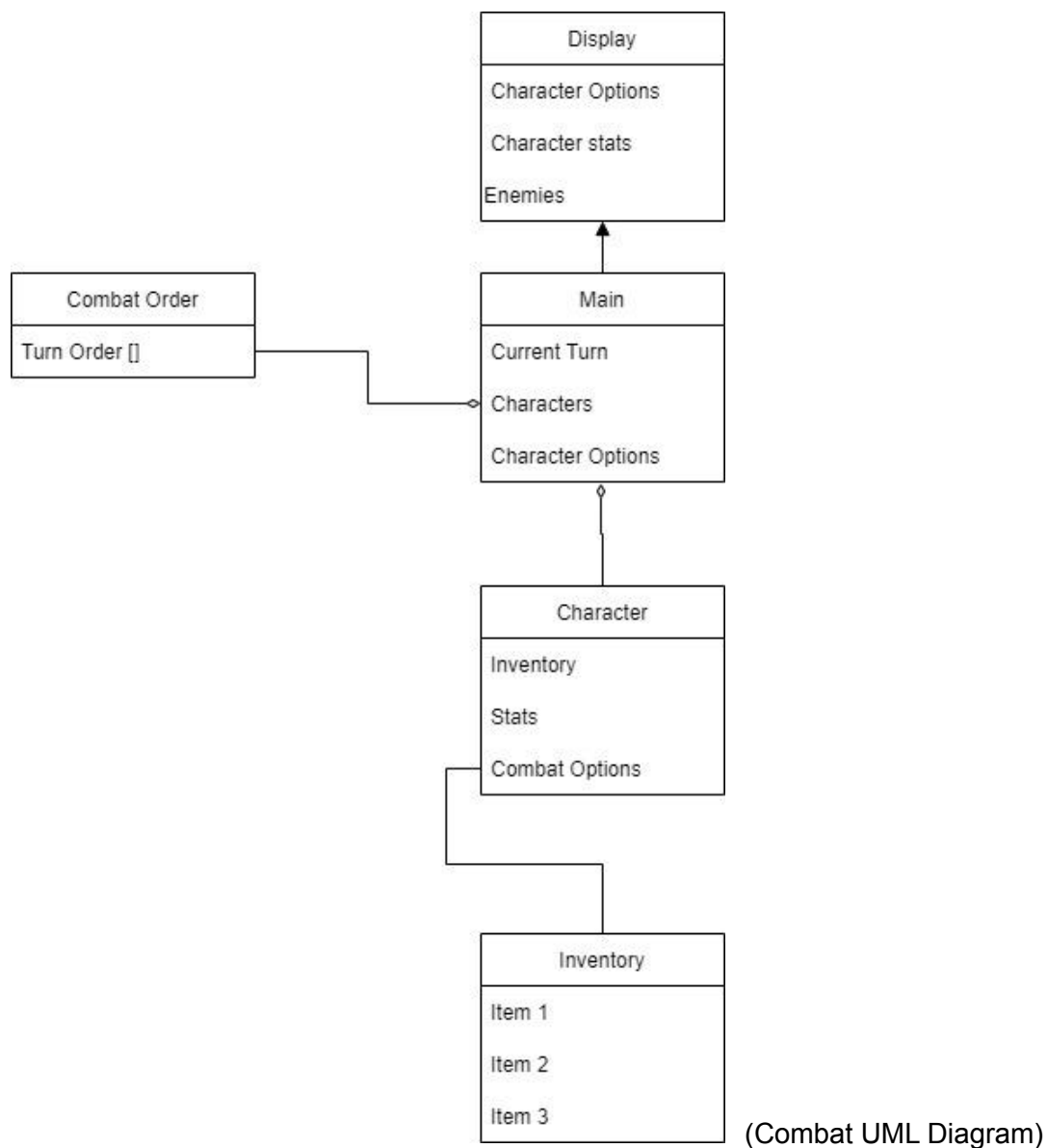
Design Decisions and Locations	Rationale and Assumptions
Create a Domain Model	A Domain Model must eventually be created. It allows us to identify major components and how they interact with one another.
Identify Domain Objects that map to functional requirements	A domain object will be used to encapsulate each distinct functional elements in a self-contained building block
Decompose Domain Objects into general and specialized Components	The domain will be broken up into specialized components. Each component will be responsible for a single main function of the code. This will help divide up the main tasks into smaller specialized components like the combat system and inventory system.
Use Java as our programming language with JavaFX as a ui frontend	<p>Our program is designed to be functional first then ui will be added down the line. Java's native design around interlocking classes and objects will make a RPG game easier to code and understand.</p> <p>For our frontend UI we have decided to use JavaFX as it is open source and one of the most modern java GUI frameworks</p>

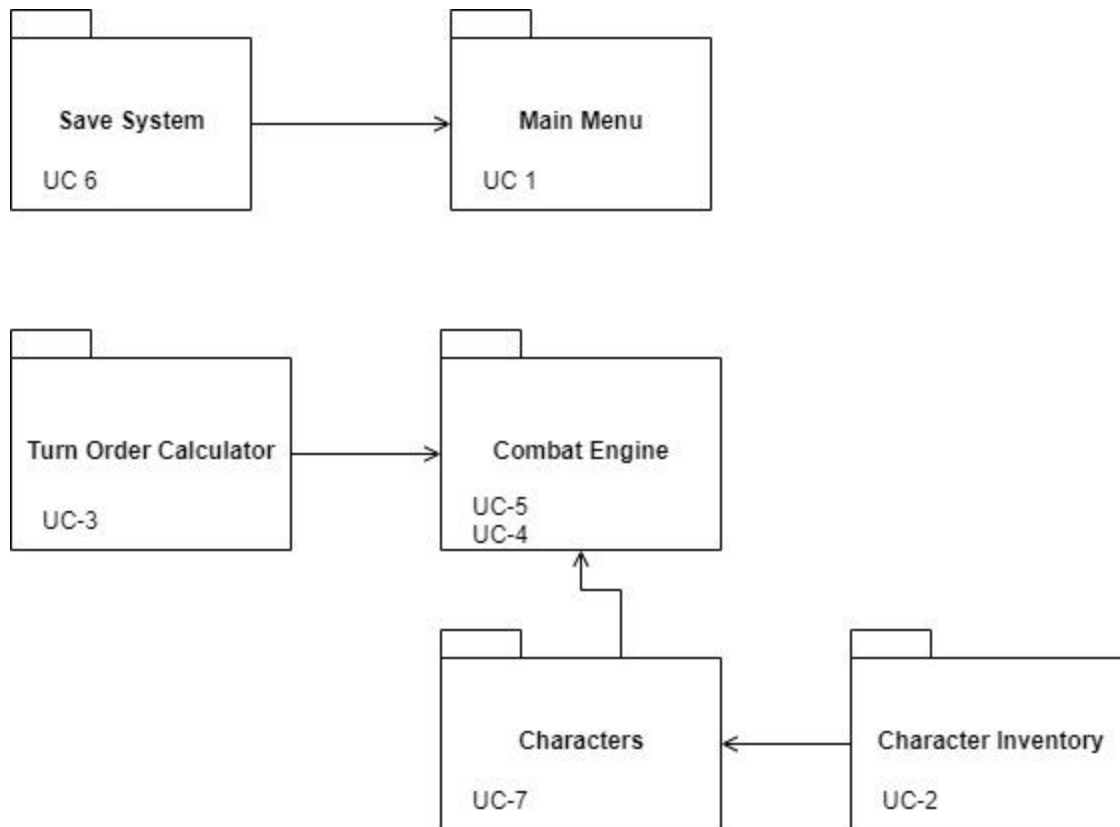
Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decisions and Locations	Rationale and Assumptions
Battle and inventory systems will be prioritized	The game won't be able to function without these two base systems, they will be the first to be implemented fully, then all other use cases will be built around these systems.

Larger Main System	The engine will consist of many components, however for combat, the main file will interact and pull things from other components as needed instead of handing the functionality off completely. This allows the program to keep track of things like turn order based on speed in a single file without high amounts of function calls.
GUI will be lowest priority	RPGs existed as text based games for many years, the end user will be ok with a completely text based game. A GUI would be considered extra flair so we will put it at the lowest point on our priority list.

## Step 6: Sketch Views and Record Design Decisions





Domain Object associated with use case model

## Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

The decision made in this iteration provided a further understanding of the functionality and how it is supported in the system. The modules associated with the use cases were identified and the modules associated with the rest of the functionality were also identified. Then from there a complete list of modules, a work assignment table was created as well.

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
		UC-1	Modules and preliminary interfaces to support this use case have been identified.
		UC-2	Modules and preliminary interfaces to support this use case have been identified.
		UC-5	Modules and preliminary interfaces to support this use case have been identified.
	QA-2		The elements that support the associated use case (UC-2) have been

			identified.
	QA-3		The elements that support the associated use case (UC-1) have been identified.
	QA-5		The elements that support the associated use case (UC-5) have been identified.
	CON-1		No relevant decisions made.
	CON-2		No relevant decisions made.
		CRN-3	Already established with more details making sure all components work in unison. Functional descriptions for components have been designed.