Post Arcana / UI

Architecture/Design Document

**Table of Contents**

1. Introduction

2. Design Goals

3. System Architecture

4. Mid Level Design

5. Detailed Class Design

6. Process View

7. Use Case View

**Change History**

**Version:** 0.1

**Modifier:** Sebastian Comette

**Date:** 04/09/2022

**Changes:** Design Document Created

**Change History**

**Version:** 0.1

**Modifier:** Sebastian Comette

**Date:** 04/11/2022

**Changes:** Design Document Finished

1. Introduction

This document describes the design and architecture of Post Arcana by the Hex Decimals. Post Arcana is a single player open world RPG, set in a post-apocalyptic small Canadian town after the introduction of magic to the real world.

The purpose of this document is to define the architecture and design of the stat systemin a manner that assists the interests of all major stakeholders. The major stakeholders and their interests are as follows:

* Developers: A design that is easy to implement that minimizes complexity
* Project Manager: A design that can be easily divided among the skills of the programmers
* Maintenance Programmers: A design that can be improved upon easily

1. Design Goals

The design goals for the stat system are as follows:

* The design must minimize complexity
* The design must allow for easy access to abstract and concrete stats
* The design should be easy to amend and balance

3.System Behavior

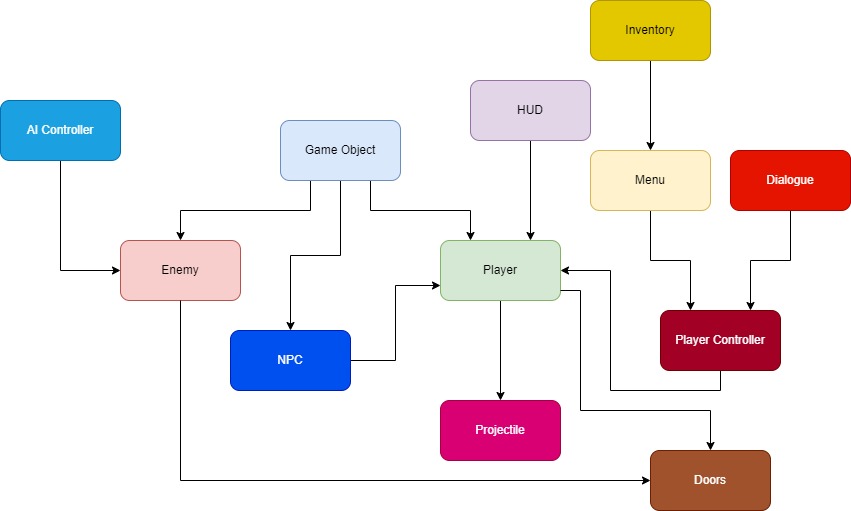
The UI or HUD system consists of four progress bars, an image, and 4 Text blocks. The HUD is updated by the player game object and displays import stats that relate to the player. The stats that are displayed are Health, Mana, Defense, and if the player has unspent skill points. When the player takes damage the health bar is depleted, this is shown through a drop in the top health bar and an animated drop in the bottom health bar. When the play spends mana the mana bar is depleted, this shown through a drop in the top mana bar and an animated drop in the bottom mana bar. When the player increases their toughness their defense is raised and this is show through the number that overlays the shield image. Last if the player has leveled up and not spent their new skill points text will appear below the defense image to remind the player they have unspent skill points.

1. Logical View

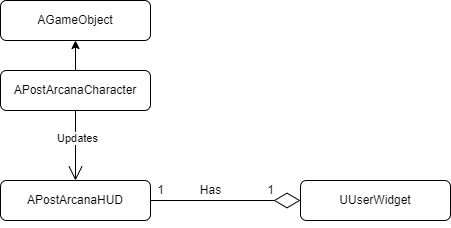
The logical view describes the main functional components of the system. This includes modules, the static relationships between modules, and their dynamic patterns of interaction.

In this section, the modules of the system are first expressed in terms of high-level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

4.1. System Architecture

****

* GameObject: Base class for the game entities (Players and enemies). Stores abstract and concrete stats.
* Enemy: The class used for enemy-only stats and components
* AI Controller: Moves and controls the enemies
* Player: The class used for player-only stats and components
* Player Controller: Receives player input and controls the player
* Projectile: Is launchd by the player, and deals damage to Game Objects
* Menu: Places a UI element on the screen given a specific input
* HUD: Shows the player important stats\
* Doors: This Module consists of multiple door classes that all provide a different method of access control.
  1. Mid Level Design

****

**AGameObject:** A character with health, mana, stats, and a level. It can take damage, and if its health goes down to 0, it will die. The amount of damage it takes is lowered based on the value of the game object’s defence stat. When a spell is casted mana will deplete based on the cost of the spell. Mana regenerates over time based on the will stat. When a game object levels up it gains skill points that can be put towards raising stats.

**APostArcanaCharacter:** A game object controlled through player input that can shoot projectiles which may damage enemies when they connect, and cost mana to use. Can also be damaged by contact with the enemy. When damaged by the enemy the player will be temporarily invincible. This game object also updates the HUD to inform the HUD of changes to the game object mana, health, defence, and skill points.

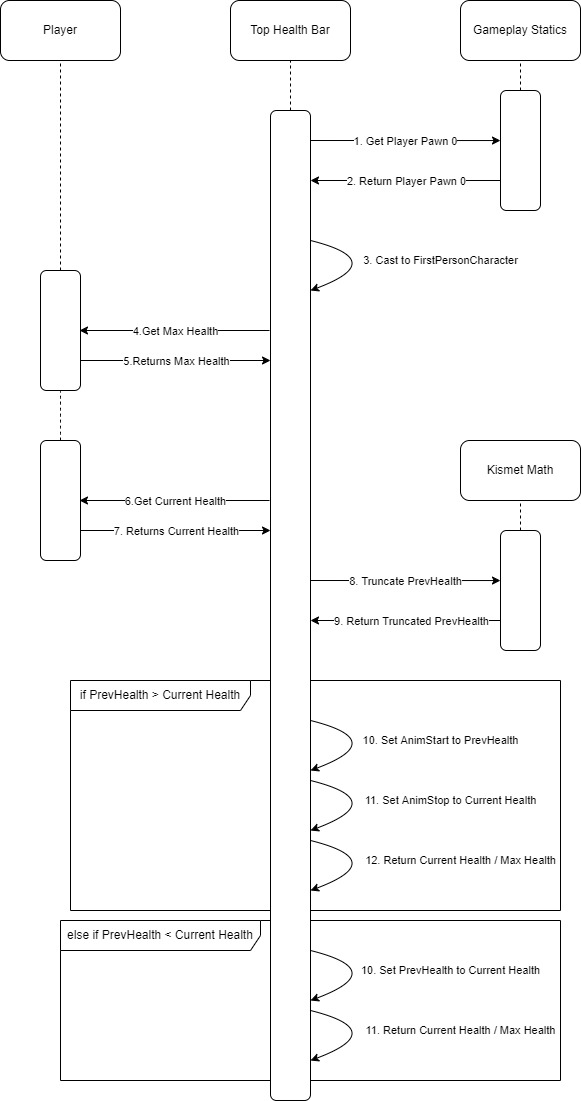
**APostArcnaHUD:** A HUD class that has a user widget and draws that use widget to the screen. The HUD class also draws the player’s crosshair on the screen

**UUserWidget:** A widget that has progress bars, text blocks and an image to create the health bars, Mana bars, defence stat, and skill points notification. The HUD draws this widget to the screen.

* 1. Detailed Class Design

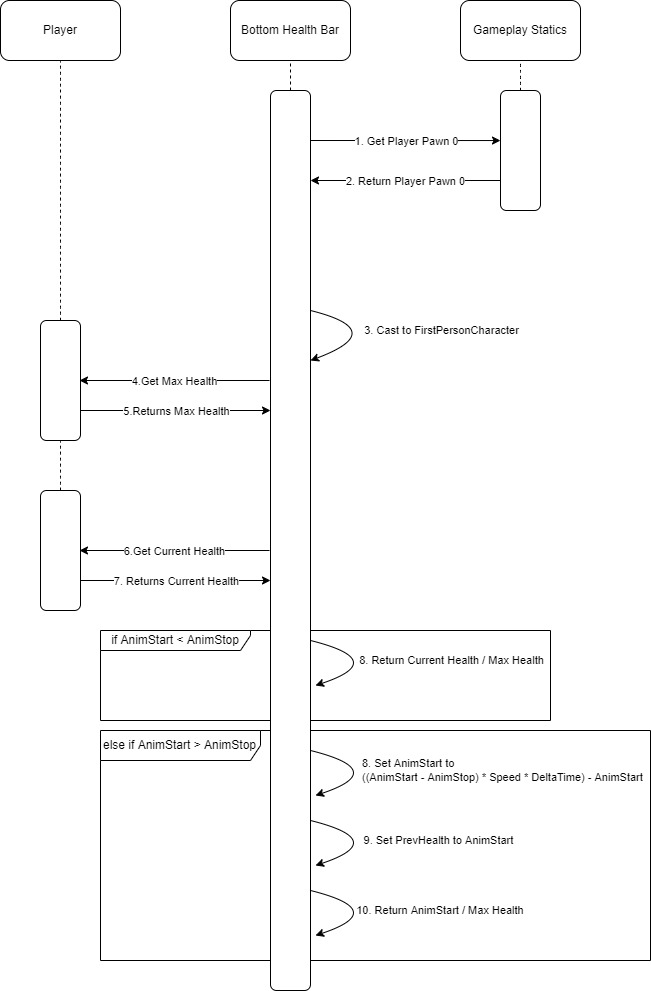


1. Process View
   1. Top Health Bar



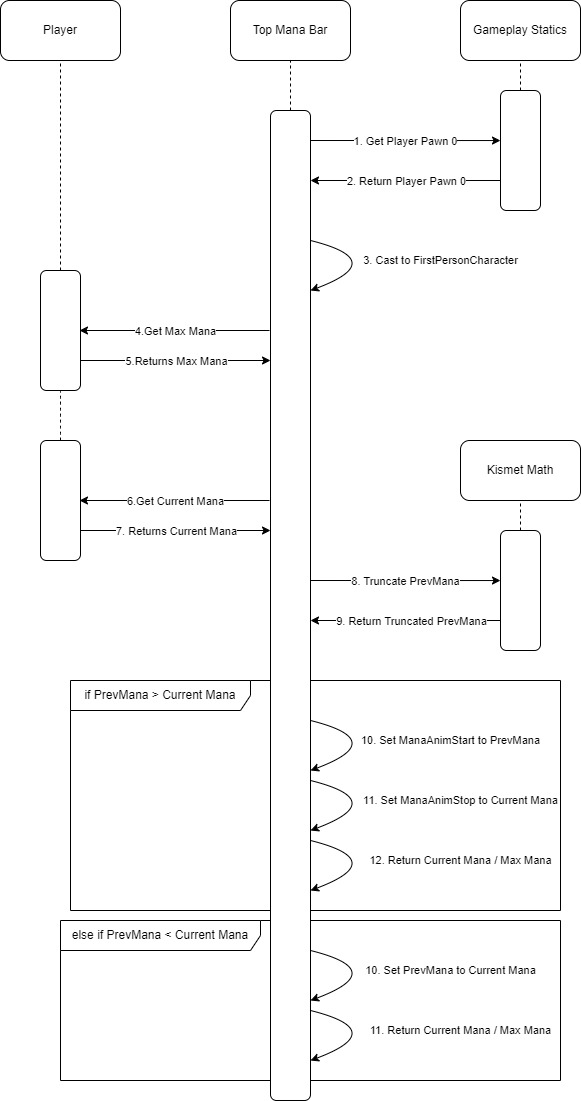
The top health bar displays the player’s current health. The health bar consistently looks for a negative change in the current health. The top health bar gets the player pawn and casts it to a first-person character. It then grabs the players current health and max health. The top health bar checks if there has been a change to the health value between checks. If there has been a change it sets the Anim start value to the health before the change, sets the Anim stop value to the new current health and then returns the new percentage the health bar should be filled. If there has been no change it makes sure that the previous health and current health match and then returns the percentage, the health bar should be filled.

* 1. Bottom Health Bar



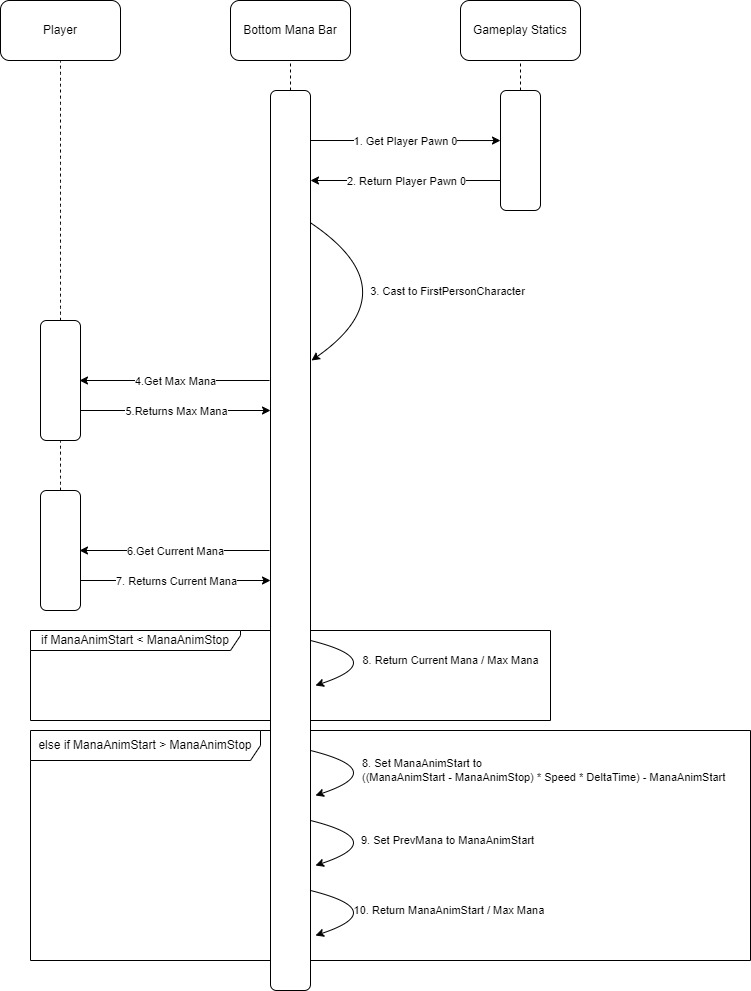
The bottom health bar becomes visible when the health has been lowered. Like the top health bar it grabs the players current health and max health. It then checks if there is a difference between the Anim start value and Anim stop value. The Anim start represents the health before a change was made and Anim stop represents the health after a change has been made. If there is no difference between them than the bottom health bar returns the percentage of the health bar that should be full, which will be the same as the top health bar. If there is a difference than the bottom health bar starts to animate from the old value to the new value slowly depleting until the values match.

* 1. Top Mana Bar



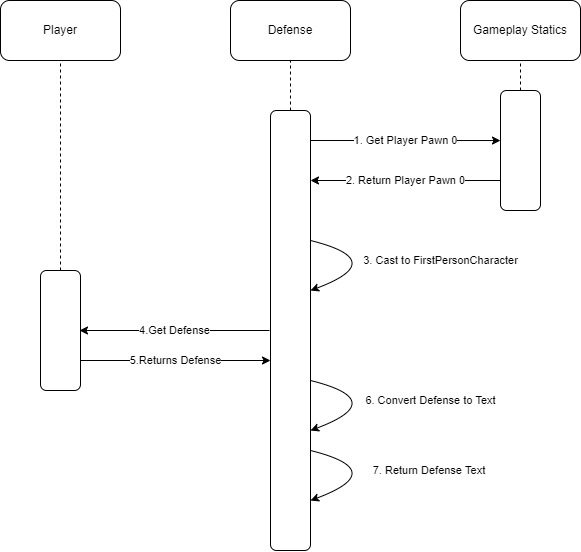
The top mana bar displays the player’s current mana. The mana bar consistently looks for a negative change in the current mana. The top mana bar gets the player pawn and casts it to a first-person character. It then grabs the players current mana and max mana. The top mana bar checks if there has been a change to the mana between checks. If there has been a change it sets the Mana Anim start value to the mana before the change, sets the Mana Anim stop value to the new current mana and then returns the new percentage the mana bar should be filled. If there has been no change it makes sure that the previous mana and current mana match and then returns the percentage, the mana bar should be at.

* 1. Bottom Mana Bar



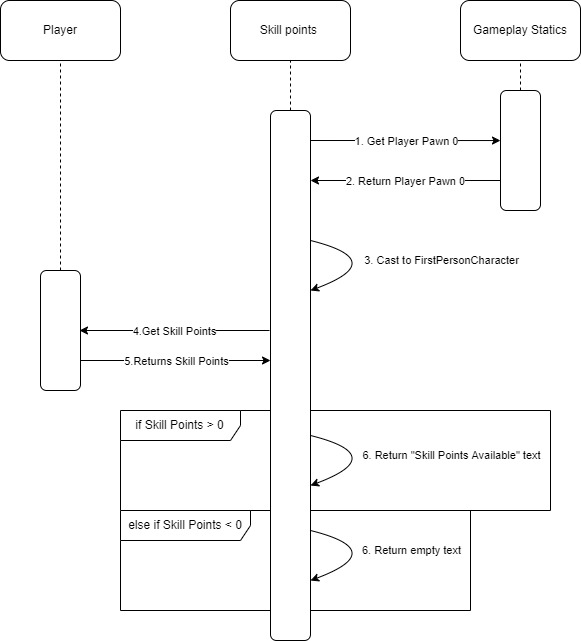
The bottom mana bar becomes visible when the mana value has been lowered. Like the top mana bar it grabs the players current mana and max mana. It then checks if there is a difference between the Mana Anim start value and Mana Anim stop value. The Mana Anim start represents the Mana before a change was made and Mana Anim stop represents the mana after a change has been made. If there is no difference between them than the bottom mana bar returns the percentage of the mana bar that should be full, which will be the same as the top mana bar. If there is a difference than the bottom mana bar starts to animate from the old value to the new value slowly depleting until they match.

* 1. Defense Text



The Defense text displays the current defense value that the player has based on their toughness level. It displays the current defense value over an image of a shield. It works by grabbing the player pawn, converting it to a first person character and then grabbing the defense value from the player. It converts this value to text and then returns the text to the widget to be displayed.

* 1. Skill Points Text



The skill points text works as a notification system. It notifies the player if they have unspent skill points. It works by grabbing the player pawn, converting it to a first-person character and then grabbing the current amount of skill points the player has. If the value of the skill points is greater than 0 then it displays text saying, “Skill Points Available”. If the value is equal to 0 then it returns empty text and is not visible to the player.

6. Use Case View

In “Content/FirstPersonCPP/Blueprints/HUD” there is a player ui widget. This is what the HUD displays.

Graphical user interface, text, application

Description automatically generated

In this widget there are 2 progress bars at the top, 2 in the left and 2 in the right. The left represents health while the right represents mana.

Graphical user interface, table

Description automatically generated

In the middle there is an image and a text block to display the defense icon and value.

A picture containing graphical user interface

Description automatically generated

Last there is a text block below the defense area that is where the skill points notification will appear.

A picture containing graphical user interface

Description automatically generated