The Stroke of Midnight / Interaction

Architecture/Design Document

**Table of Contents**

1 Introduction.. 3

2 Design Goals.. 3

3 System Behavior.. 3

4 Logical View... 4

4.1 High-Level Design (Architecture) 4

4.2 Mid-Level Design. 5

4.3 Detailed Class Design. 6

5 Process View... 7

6 Use Case View... 9

Change History

**Version:** 0.1

**Modifier:** Omer Kocar and Zoe Purcell

**Date:** 03 / 09 / 2021

**Description of Change:** Module Design Document started.

**Version:** 0.2

**Modifier:** Omer Kocar

**Date:** 03 / 16 / 2021

**Description of Change:** Document finished

# **1 Introduction**

This document describes the architecture and design for The Stroke of Midnight being developed by Memento Game Studios. The Stroke of Midnight is a single-player psychological horror game where the player breaks into an art gallery on a dare, which turns into a fight for his life as he attempts to escape.

The purpose of this document is to describe the architecture and design of the Interaction Module application in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

* Developers – they want an architecture that will minimize complexity and development effort.
* Project Manager – the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies. For this to happen, the modules need well-defined interfaces. Also, because most individuals specialize in a particular skill or technology, modules should be designed around specific expertise. For example, all UI logic might be encapsulated in one module. Another might have all game logic.
* Maintenance Programmers – they want assurance that the system will be easy to evolve and maintain into the future.

# **2 Design Goals**

The design priorities for the Interaction system are:

* The design should minimize complexity and development effort.

# **3 System Behavior**

The Interaction Module is built from a single InteractionBase parent that all Interactable items and props inherit from. This allows all interactable objects to share common properties. Once an interactable item is placed in the level, the system should allow for the Player to interact with it or pick it up with a keypress. The behaviour or result of said interaction depends on the child class that inherits from the InteractionBase.

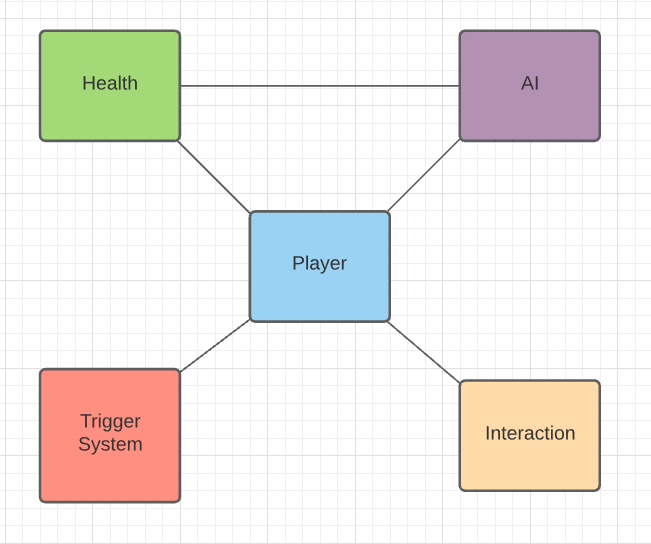
# **4 Logical View**

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.

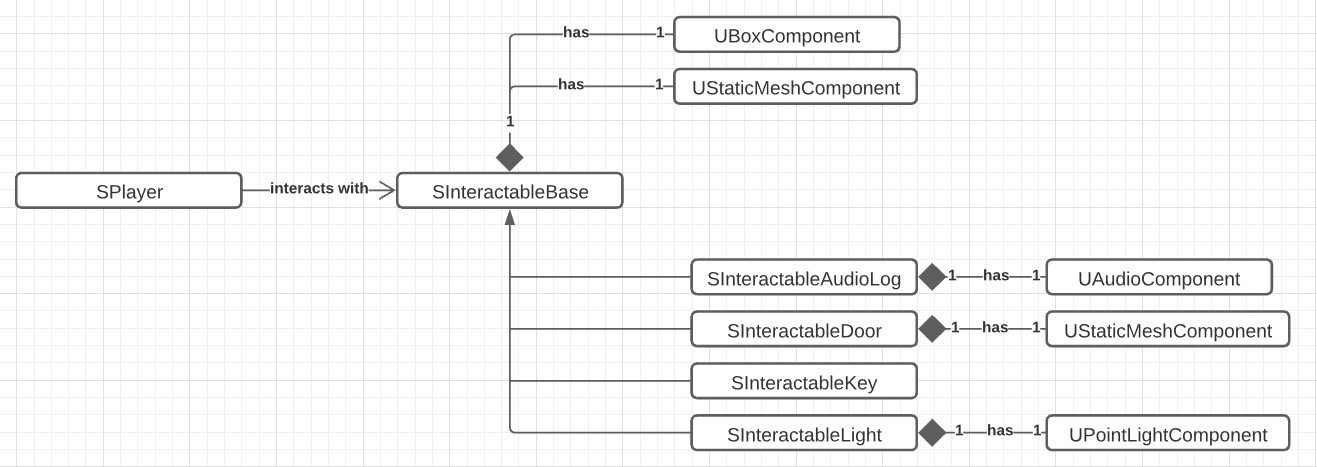
## **4.1 High-Level Design (Architecture of the Entire system)**

The high-level view or architecture consists of 5 major components:

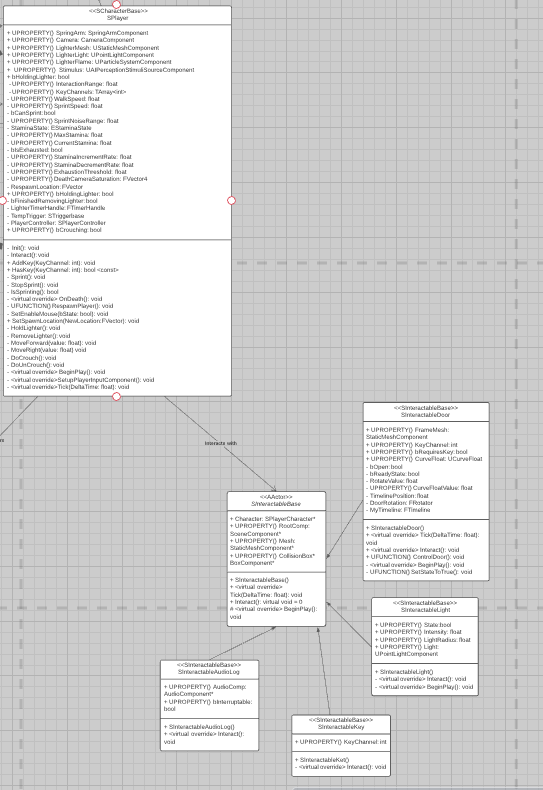


* Interaction System handles different objects that can be picked up, turned on, or interacted with by the player.
* Event Trigger Module holds all the unique event triggers and their effects on the Player character.
* Player System is the main system, which consists of a controlled character that takes in user input.
* AI System is used for all enemy behaviour.
* Health System handles how Characters take damage/receive health.

## **4.2 Mid-Level Design of the Interaction Module**



## **4.3 Detailed Class Design of the Interaction Module**



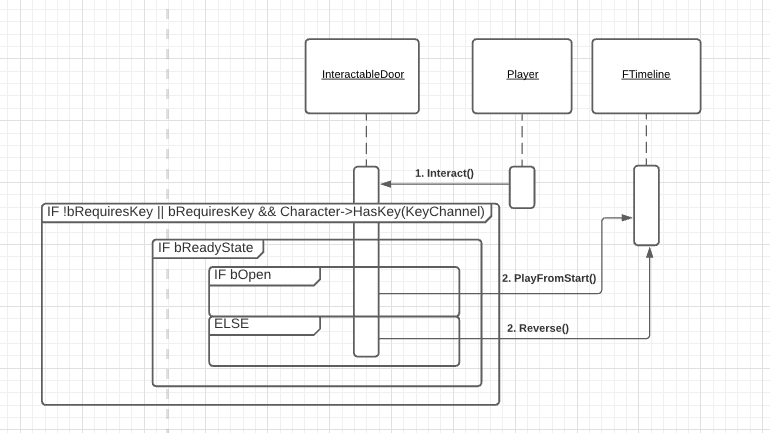
# **5 Process View of the Interaction Module**

Interaction:

All Interactable objects are triggered when the player hits the E key when close enough and looking at them. When the player interacts with an interactable object, its Interact function is called from the Player class.

Every Interactable object is a child of InteractableBase, which is an abstract class and has the parent pure virtual Interact function.

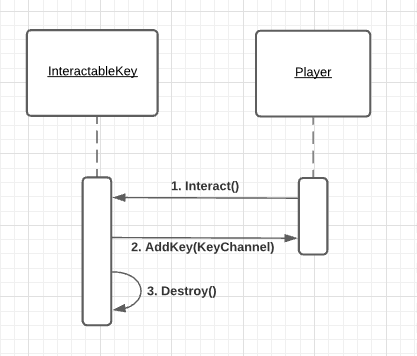
Interactable Door



Interactable Door plays a Timeline to open/close itself. As the door opens/closes, the rotation value in the CurveFloat increases/decreases.

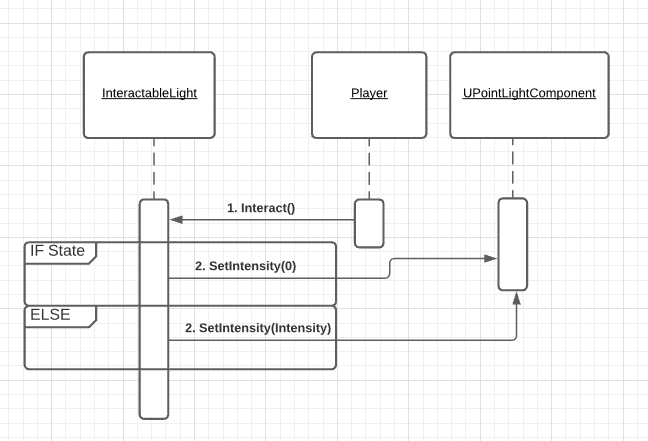
If the door does not require a key, it will open as soon as the player successfully interacts with it. If it does, the player has to have collected the right key to open it, first. After passing the HasKey check, the door will open as normal.

Interactable Key



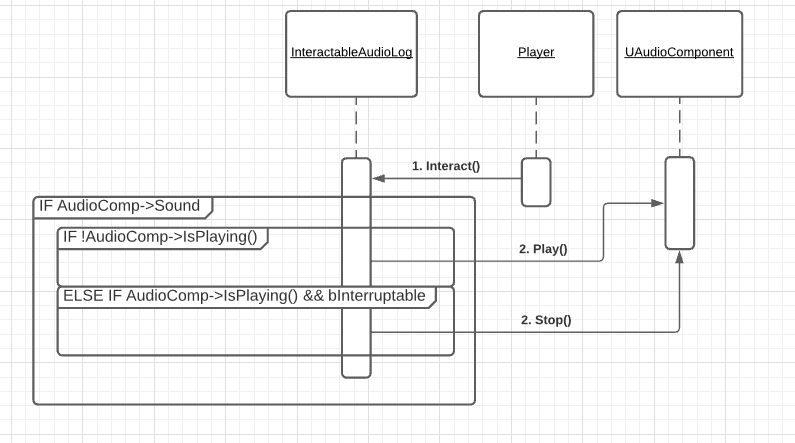
When interacted, Interactable Key adds the KeyChannel value to the Player’s KeyChannels list. After that, it will destroy itself.

Interactable Light



Interactable Light sets its Light Component’s intensity to either Intensity value of 0 based on the On/Off state.

Interactable Audio Log

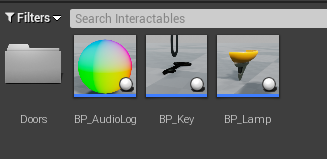


Interactable Audio Log plays a sound if it has a valid sound selected and the sound is not already playing. If the bInterruptable is set to true, it can be stopped while playing by the player interacting again.

# **6 Use Case View**

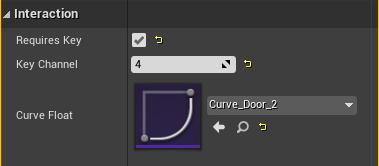
Placing an Interactable Object

All the interactable objects can be dragged from straight out of the editor into the world.



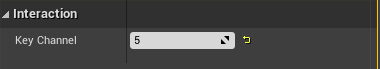
Interactable Door

In order for the door to properly Open/Close, CurveFloat needs to be set. So far, there are only 3 Door Curve Floats: 1, 1.25, 2 seconds. If the door requires a key, Requires Key needs to be set and KeyChannel needs to be set the same value as its key’s Channel value.



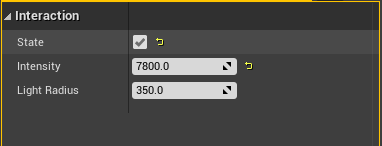
Interactable Key

Interactable Key is one of the simplest Interactable objects to set up. All you need is to set the Key Channel value the same as the door’s Key Channel.



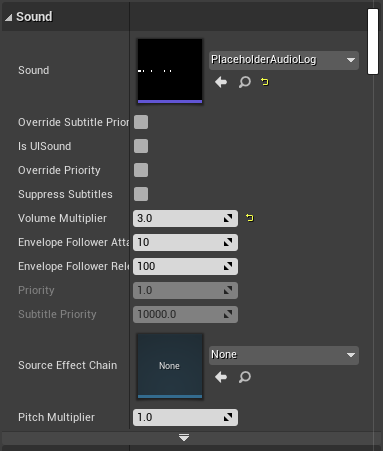
Interactable Light

Interactable Light will be on at the start if State is true and will be off if it is false. Intensity value sets the light’s intensity at the start and LightRadius value sets the light’s attenuation radius.



Interactable Audio Log

Audio Log has an Audio Component so you need to get to the Sound section to select a sound to play. You may also set up the other properties underneath.



Once that is done, get to the Interaction section and set Interruptible to true if you want the Audio Log to be able to be interrupted by the player while playing. Leave it as false otherwise.

