Project Awesome

Pickup Object Document

**Table of Contents**

[1 Introduction](#_Toc131876418) 3

[2 Design Goals 3](#_Toc131876419)

[3 System Behavior 3](#_Toc131876420)

[4 Logical View 4](#_Toc131876421)

4.1 High-Level Design ……………………………………………………………… 4

**4.2 Mid-Level Design** ……………………………………………………………..... **4**

4.3 Detailed Class Design for Project Awesome ………………………………….. 5

Change History

**Version:** <0.1>

**Modifier:** <Brandon Coates>

**Date:** 23/02/2021

**Description of Change:** First Edition for Alpha 1

**Version:** <0.2>

**Modifier:** <Brandon Coates>

**Date:** 12/04/2021

**Description of Change:** Implemented Recommended Changes to Charts and Information

# Introduction

This document describes the architecture and design for Project Awesome being developed for Team Awesome. Project Awesome is a Single Player casual puzzle game.

The purpose of this document is to describe the architecture and design of the pickup system which will be the most important feature to solve puzzles and move objects.

# Design Goals

The design priorities for the game system are:

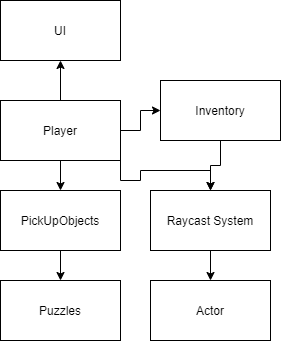
* The design should be highly open to iterations and changes, with no problems of duplicated objects.

# System Behavior

The game is built with only a single level completed. The complexity of the system is causally related to the limited scope of the project. If there were more levels to be added, along with more puzzles the complexity will grow exponentially as well as objects the player can pick up.

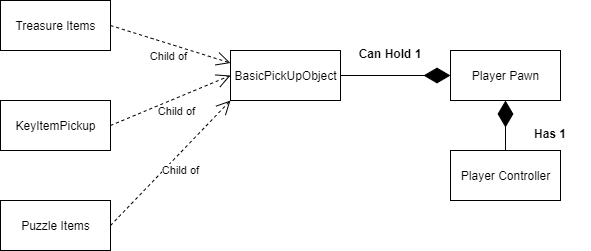
# Logical View

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



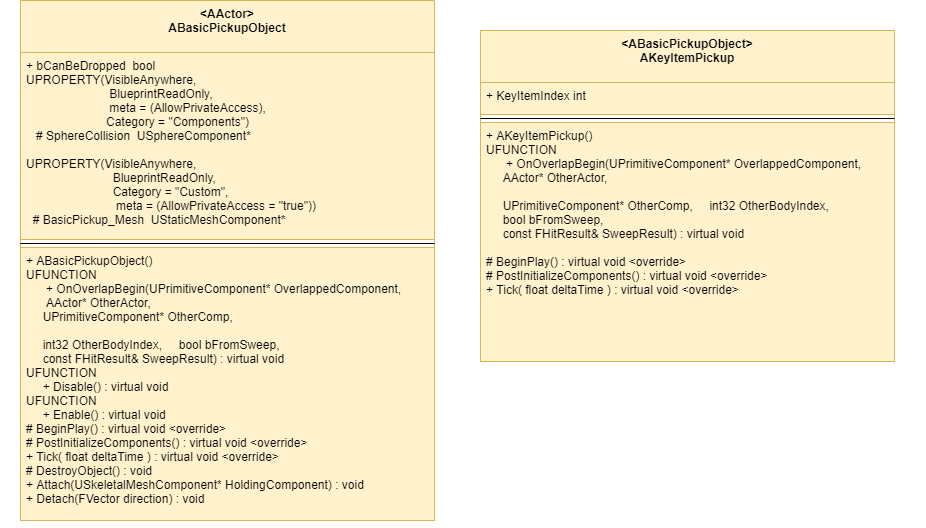
* The Player module is the main module.
* The Raycast module is based on the Unreal Engine Raycast system.
* The Inventory module allows for holding puzzle and treasure items.
* The Puzzle module contains all the puzzles in the level which the player can interact with and solve.
* The UI module contains item information for the player (to be completed).

## Mid-Level Design



* BasicPickUpObject class is used by all other objects which the player needs to pick up.
* The information on what the player is holding will be stored in the player object and displayed using UI (to be completed)
* All present and future object pick-ups will be derived from this class.
* For example, the KeyItemPickup class inherits from ABasicPickupObject so it can be interacted and moved around by the player. But the key will interact with a door object to unlock another area of the map in the next update.

## Detailed Class Design of Project Awesome



See Mid-Level Design for details.

## Use Case of Pickups

* This is how all pickups work in the game. The player calls their raycast, it detects if it is colliding with an object it can interact with then attaches to the player until dropped.