Too Many Eyes / *AI*

System Design Document

# Changes

## Version 0.1

Editor: Alexander Bowerman

22/02/2022

* Document Created.
* Introduction, Design Goals, and Behaviour descriptions added.

## Version 0.2

Editor: Alexander Bowerman

23/02/2022

* Mid-Level view added.

## Version 0.2

Editor: Alexander Bowerman

23/02/2022

* Logical view added.
* Process view added.
* Case view added.

# Introduction

*This document details the design and purpose of an AI system designed for Too Many Eyes. Designed for the Blight Brew Game, the document will cover;*

*⦁* *Usage for the designers.*

*⦁* *UML for the Programmers.*

# Design Goals

The goal of this system is to;

* *Have immersive Artificial Intelligence for NPCs*
* *Make the game world more filled with life*
* *Add an additional layer of interaction between the player and the game*

# Behaviour

*The AI System is a custom Artificial Intelligence system, designed for ease of implementation in the editor. The AI System acts as a multi-faceted way for the world to be filled with life, add a layer of immersion, and add the ability for the player to run their shop.*

* ***Behaviour Trees (BT)*** *– Custom behaviour trees created for each NPC type*
  + *Wandering AI that follow spline path(s) around the world*
  + *Fauna AI that spawn in random amounts and types and walk around a defined radius*
  + *Shopper AI that move around the player’s shop, grab items off the shelves, and move to the checkout counter to be cashed out (purchase the item)*
* ***Custom Tasks****– A variety of custom Behaviour Tree (BT) Tasks such as:*
  + *Find random navigable target locations within a radius defined by the BT*
  + *Find navigable target locations along a spline path defined by the BT*
  + *Compare distances between target locations, and move to the closest one*
  + *Detect when the end of a spline path has been reached and either:*
    - *Move to the next spline path (if available)*
    - *Turn around and navigate the spline path in reverse*

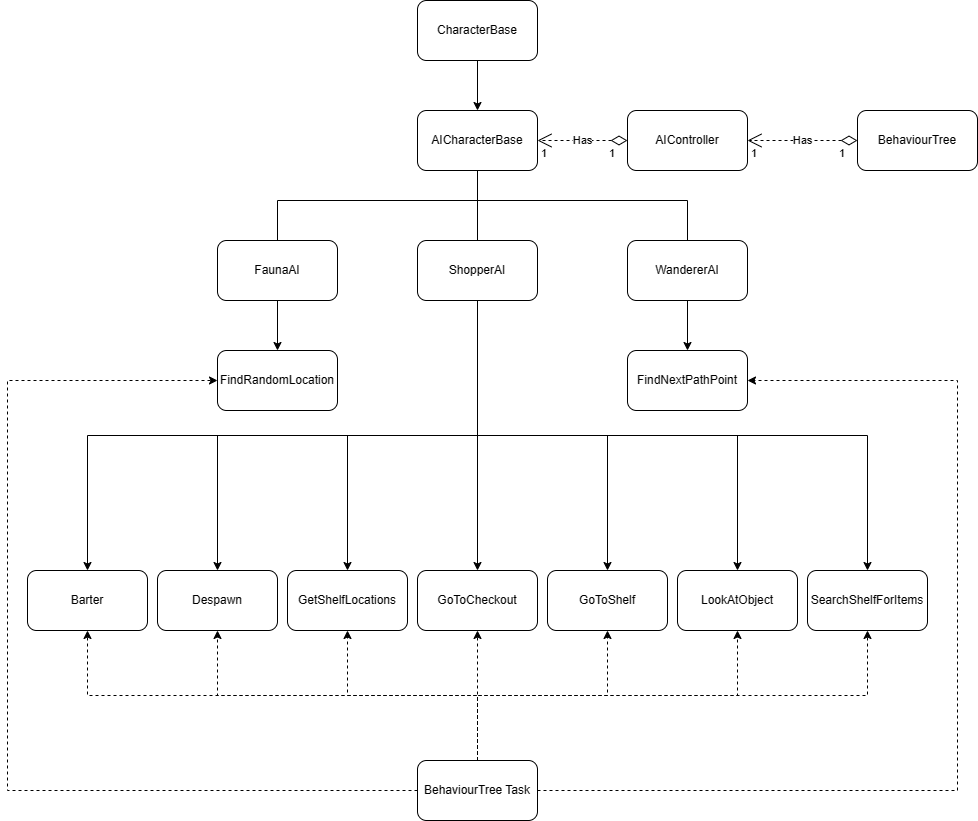
# High Level Design

Graphical user interface

Description automatically generated

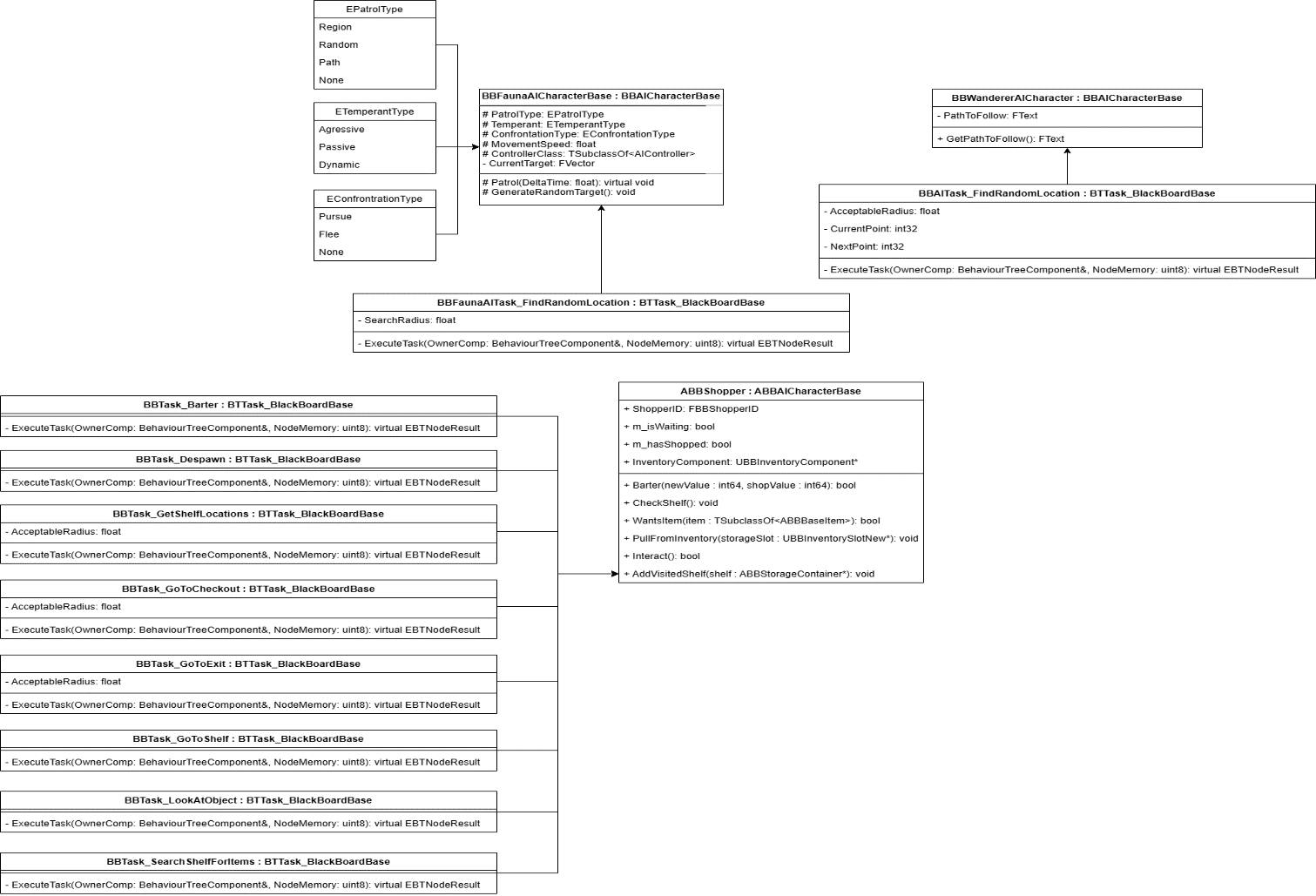
* Player system - the system is our core system and is what the player interacts with to use all our other systems.
* Locomotion system - an advanced movement system we’re using to give us more control over the players movement. This gives us variables we can control, such as stamina, and additional functionality, such as climbing and vaulting.
* Potion system – a system meant to be easy to use for designers. The system allows for the designers as much freedom as possible, while giving the coders a minimal amount of updating to do. This is the main system the player will use to complete puzzles and generate income.
* Inventory system – a system meant to be easy to use for designers. The system allows for the designers as much freedom as possible, while giving the coders a minimal amount of updating to do. The system works as a container to hold items that the player collects. This includes the players backpack, as well as storage containers around the world.
* Resource system – a system meant to be the first of the steps for the player to generate income. This system gives the player ways to harvest materials and shows off behavioural logic for when the system is used.
* Crafting System – the system in between the Resource system, and the Shop/Potion systems, in terms of actual gameplay. This systems job is to refine the resources the player gathers into other items or potions for use in one of the other systems.
* Shop system - a system to generate income, and allow for the player to improve. The system interfaces with AI, the player, and Storage objects, to give the player a way of selling items to NPC’s. This can have numerous effects on the AI and is the primary source of income, a necessary resource for improving the players arsenal, and serves to break the monotony of just grinding for resources.
* AI System - The system dealing with the NPC’s in the world, including fauna. The system interfaces with the shop system, allowing for NPC’s to be customers, as well as gives the AI their logic for movement, interaction, and anything else.
* Economy System – A system that goes unseen by the player, changes the value of items based on selling history, value of the local economy, and a degree of randomness. This should make the world feel a little bit more alive, and give the player a reason to continually adjust prices based on customer responses.
* User Interface System – A system used to show, hide, and allow the player to control the Interface. The interface controls almost every system in some way, but mainly through the player.

# Mid Level View



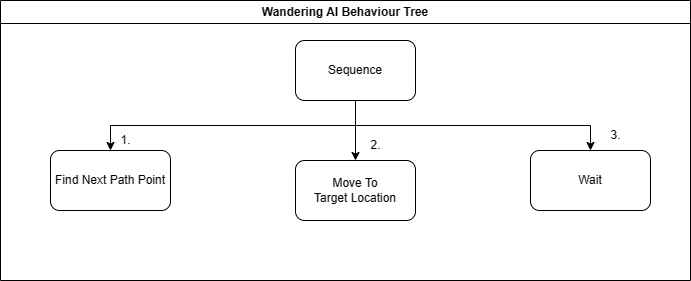
* ***WanderingAI*** *– An AI Class that has a list of Spline Path(s) they will follow. Uses the following BTTasks:*
  + *FindNextPathPoint*
* ***FaunaAI –*** *An AI Class that spawns randomly, and moves randomly in a given radius Uses the following BTTasks:*
  + *FindRandomLocation*
* ***ShopperAI*** *– An AI Class that spawns when the shop is open, navigates to the shop’s shelves, picks up items they are shopping for, and then purchases them from the player. Uses the following BTTasks:*
  + *Barter – Barters the sale price with the player*
  + *Despawn – Removes the owning AI from the shop*
  + *GetShelfLocations – Find all of the available shelves in the shop*
  + *GoToCheckout – Go towards the checkout in the shop*
  + *GoToShelf – Go to the next unvisited shelf*
  + *LookAtObject – Look at the object that the AI should be interacting with*
  + *SearchShelfForItems – Search the current shelf for items in shopping list, and add them to inventory*
* ***AI Controller(s)*** *– Each controller has its own Behaviour Tree and Blackboard*
* ***Behaviour Tree****– The main layer of communication between the AI class and the AI Controller, by using custom BTTasks.*
* ***BTTask*** *– The parent class for all Behaviour Tree Tasks*

# Logical View

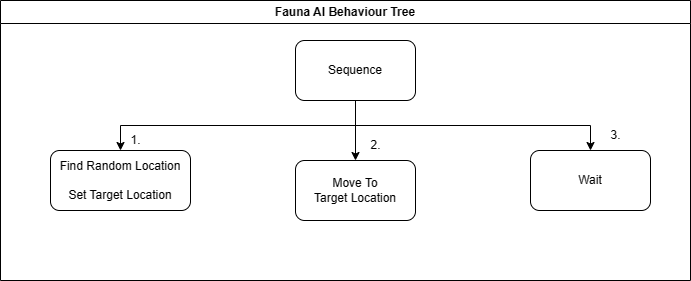


# Process View

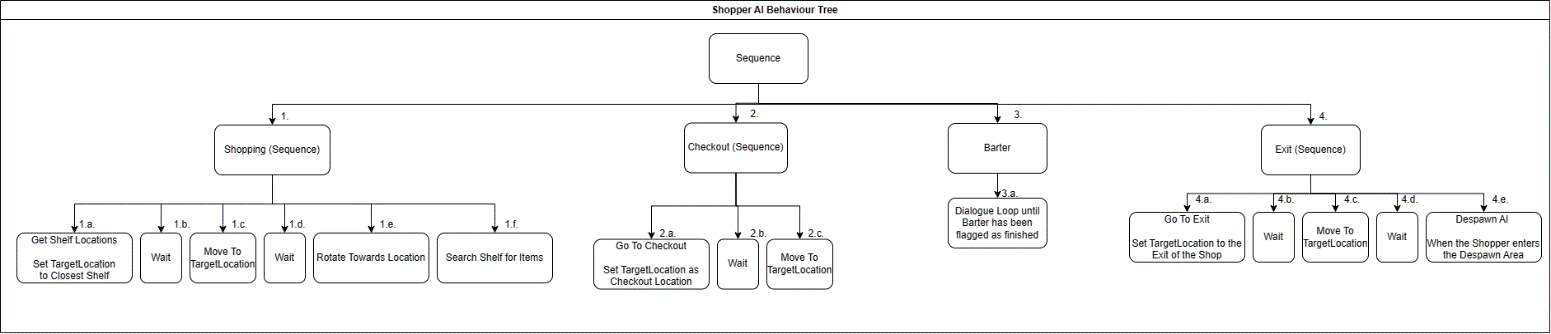
## Wanderer AI



## Fauna AI

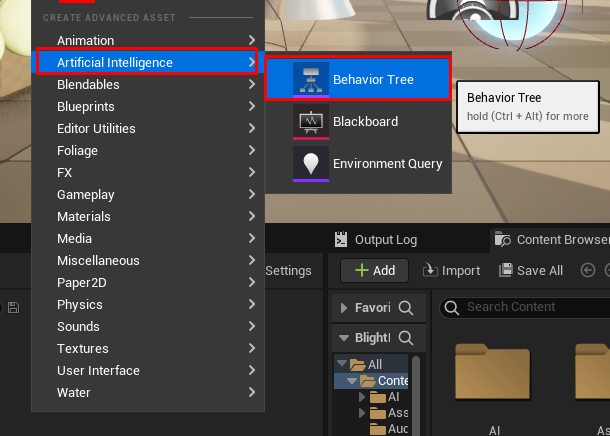


## Shopper AI

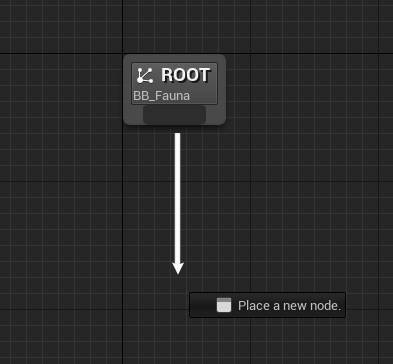


# Use Case View

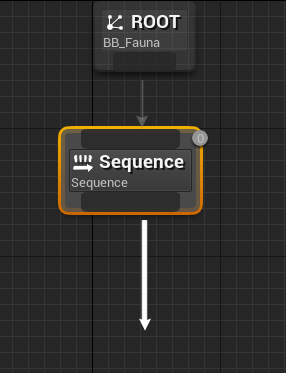
## Creating a Behaviour Tree



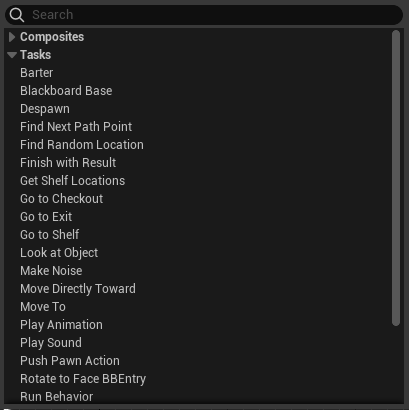
Right click , select Artificial Intelligence, select Behaviour Tree



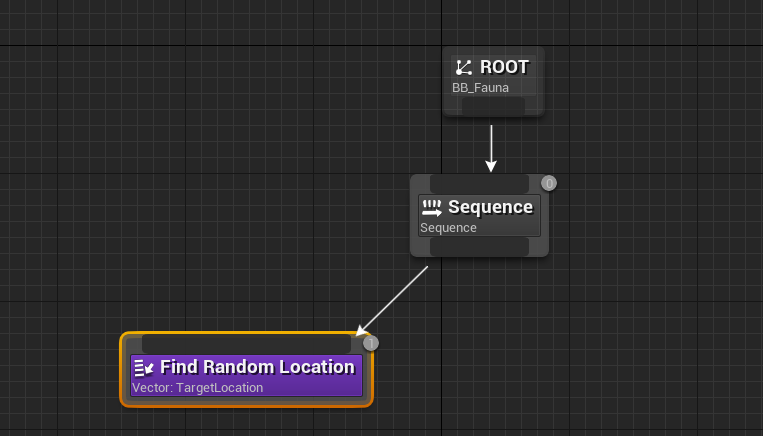
Drag off from Root, to place a new node, choose Sequence



Drag off from Sequence, to create tasks



Open the tasks drop-down to choose from Unreal & BlightBrew Tasks



Populate the Behaviour Tree with Tasks