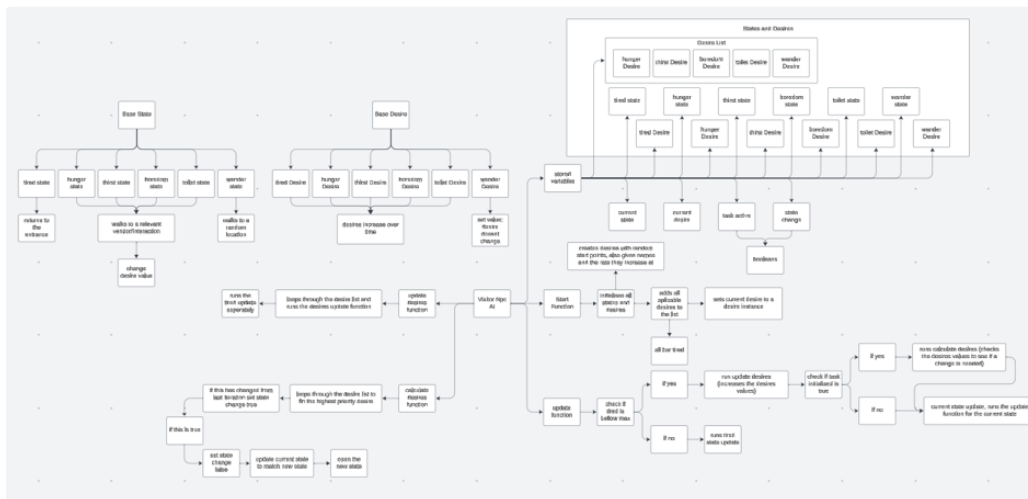


NPC AI system

Below is a diagram detailing how the system functions, but i will go into closer detail.



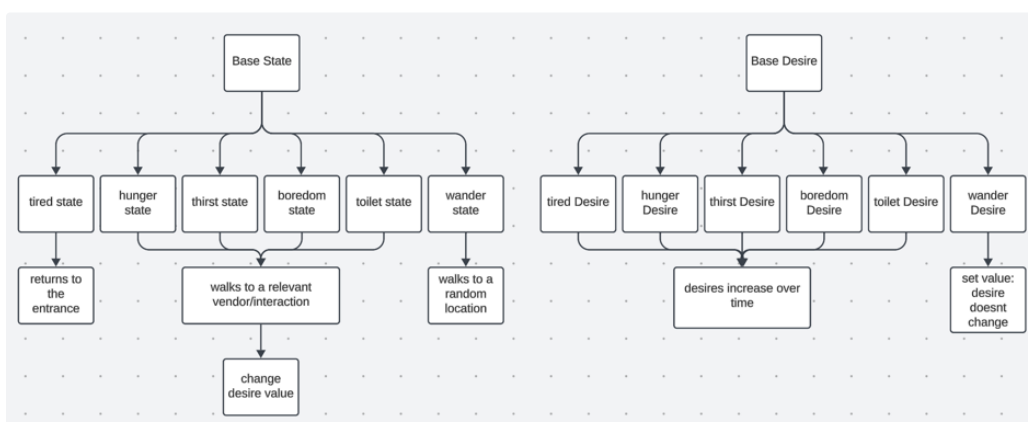
State and desire setup:

The states and desires are set up a such:

Where there is a base function for each despite some state having a different function to the others.

Current parent child diagram below, with some brief descriptions of what they do.

It is very easy to add new desires and states into the project, just not sure what else to add for now.

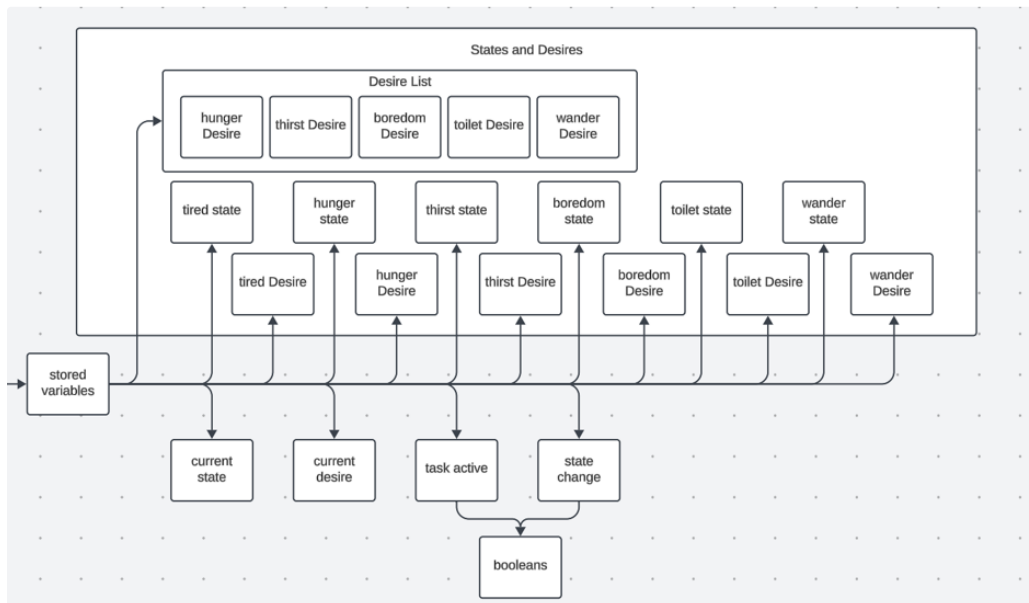


Visitor NPC AI Class - 1st iteration (outdated)

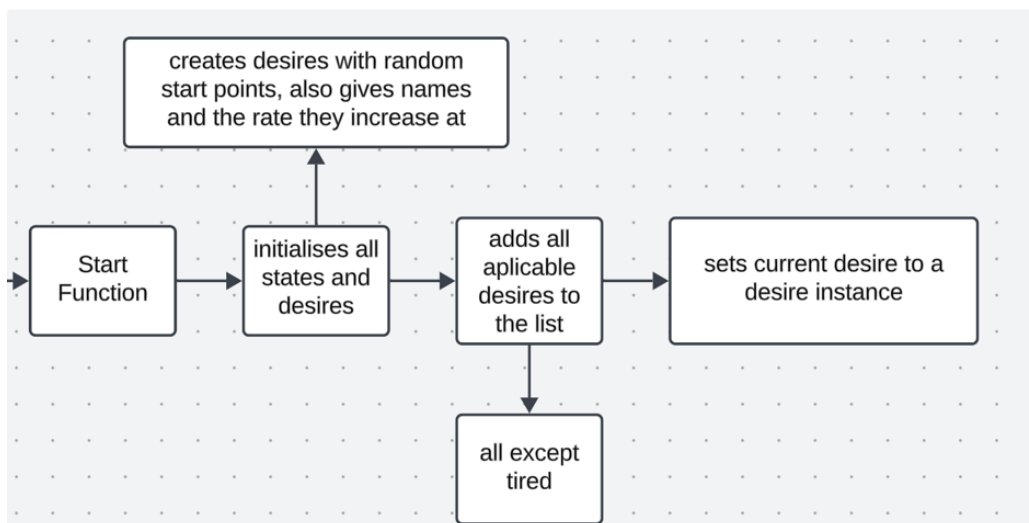
This is the class that runs everything, it uses the desires and states but it does all of the decision making.

Stored variables:

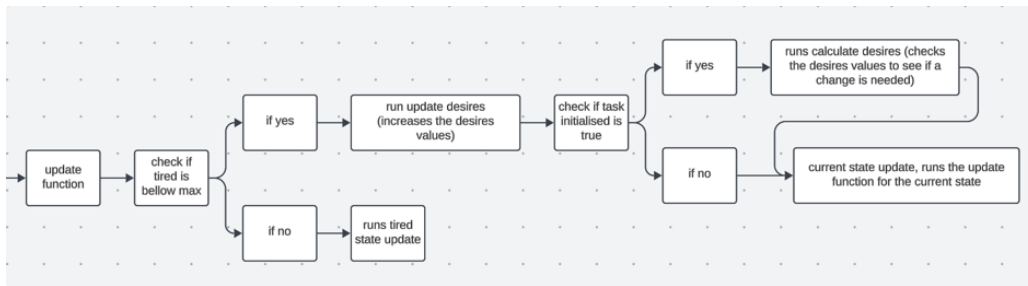
Below is a diagram showing all of the logic variable stored by the class, along with the desire list and which desires it stores.



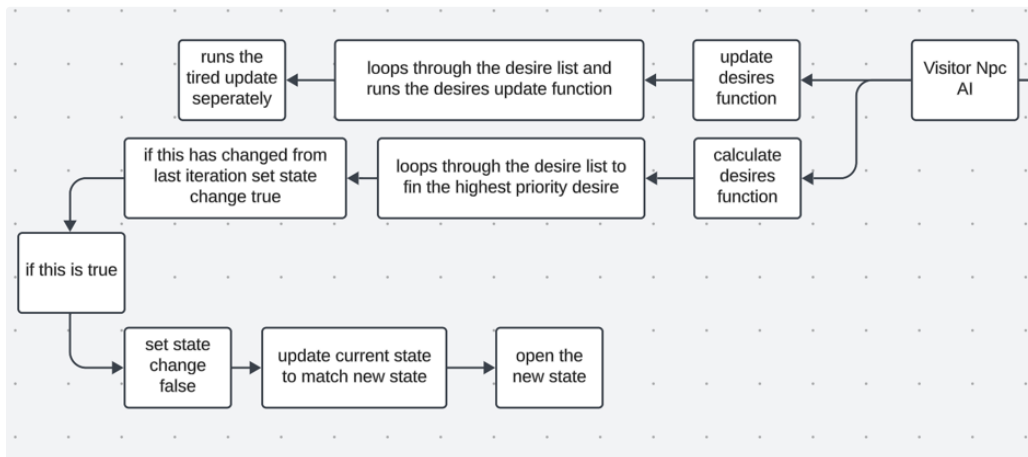
Start function is run on load of class, so immediately:



Update function is run every frame:



Calculate and update desires:



Due to the way the AI works we can add as many or as few desires as we want, it also hold all of the current desires values publicly, meaning they can be accessed easily for when we want the player to be able to inspect the NPCs.

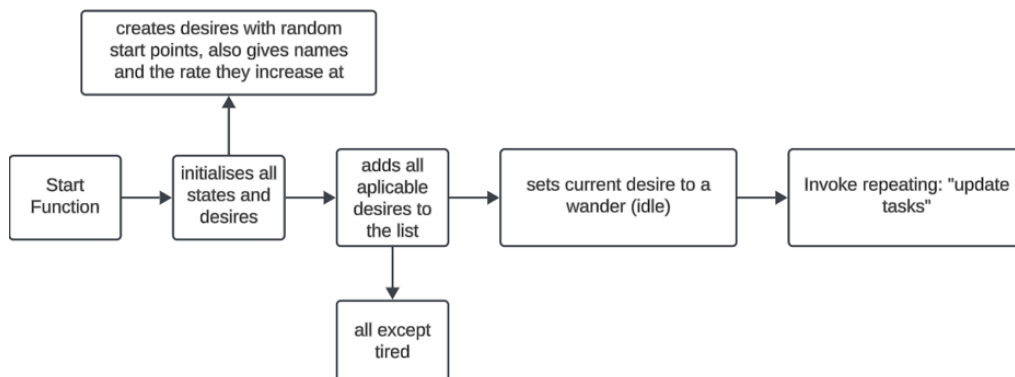
Visitor NPC AI Class - 2nd iteration (current)

Changes made to improve performance and legibility.

Changes to start and update:

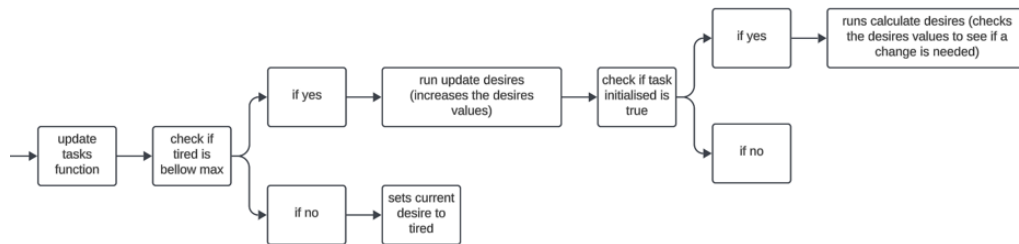
Start function now invokes a separate update tasks function.

This sets a function to be called after x seconds, repeating every x seconds.



Update function now just calls current state update.

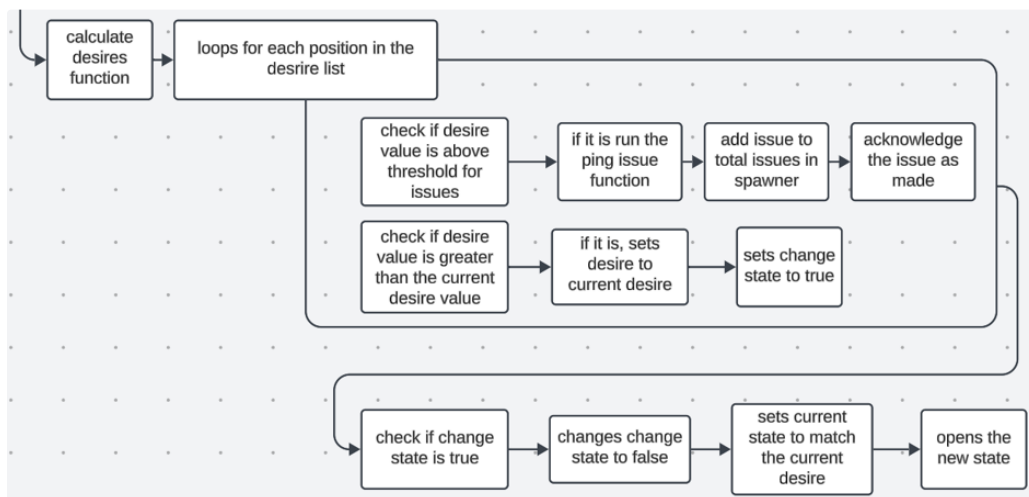
“Update tasks” function replaces the old update function as it can be controlled by invoke repeating; this function doesn't run update at all.



Everything else related has remained consistent to the last iteration.

3rd iteration

Added in issues where the visitor complains when a desire gets too high.



Ping issue function just outputs the complaint.

Moved the move function into the default state so that its virtual not repeated in each state.

Added satisfaction

NPC satisfaction is linked to the desire value for the current task, at time of task completion, the npc checks the value of the current desire checking to see whether the value is above or below the threshold. If the desire is above the threshold, the satisfaction is decreased, alluding to displeasure at how long it took to fulfill the issue, if it is below the threshold, the satisfaction is increased. When the NPC leaves the park, they leave their satisfaction value with the spawner which takes that value and changes the spawn rate based on that.