Cat Golf Devlog

# 29/05/2023 – cat senses

We have figured out how we will implement cat senses.

Each cat will have a list of objects that it has in its notice. These are the things it has noticed are there. We have developed the system that the cats use to notice things. We will have a sensory update, where the notice calculations are run again after the delay is done.

This is a quick graphic of the cat’s senses. To start, vision. Cats have two types of vision: peripheral and main. If something is in its main vision, it will detect it based on sight alone. If its in peripheral vision, the sight detection probability will be multiplied by a function of the objects speed. Cats also detect sound all around them. A picture containing drawing, child art, circle, sketch

Description automatically generated

These calculations give a percentage chance that the cat will notice the object. Because of the consistent sensory update timing, even quite low chances are actually quite frequent. If the update happens every seconds, then it only takes 5 seconds for the cat to probably notice something with a 20% chance of getting noticed.

Cats have individual sense parameters that mess with the functions we have for calculating these chances. These parameters are:

certainSight: the sight distance where a cat is certain to notice an object

maxSight: the maximum sight a cat can see something

certainSpeed: the minimum speed of an object where speed doesn’t affect the percentage calculation any more.

Hearing: something or other.

Here is the sight distance calculation:

Here’s the velocity calculation:

Here’s the sound calculation. Note here that this uses the sound’s intensity as a part in the formula. The idea is that a cat’s hearing can reduce the intensity of a noise linearly.

In order to simulate things like sound occlusion, instead of developing a complex sound system our idea is that if there is an object in-between the sound source and the cat, we remove a constant from the intensity of the sound.

If an object is in the cat’s primary vision arc, the narrow one, then we use the distance calculation to determine the probability it will be seen. If its in the cat’s peripheral vision, then we multiply the distance calculation by the velocity calculation.

For sound, when a sound event is made it runs the calculation for all cats.

These fancy curves probably won’t matter when we actually do the game. Things will either get noticed within a boundary or they won’t. Anyway, that’s all.

# Cat Animation

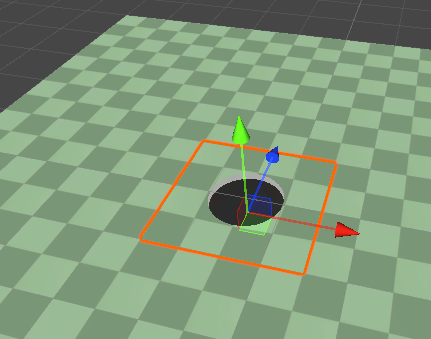
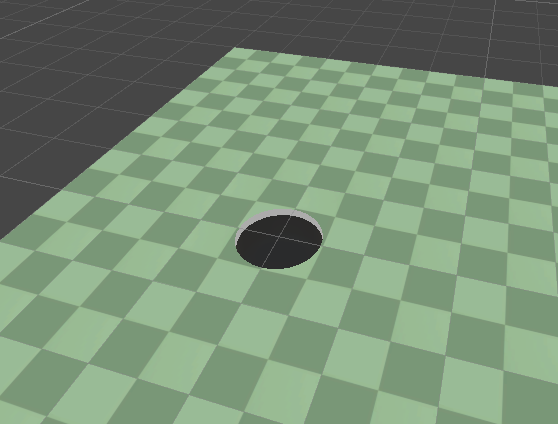
We have decided to handle the cat animations with in-game scripts rather than set animations. This is because its impossible to have an animation for every turn, every look, every walk, and so forth.

The current model and rig for the cat is also going to be replaced for a more realistic one with useful bones for this style of animation.

There’s not much to say here, but the generic idea is to use inverse kinematic animation so that the body realistically follows the feet and the head.

There’s a lot more research to be done into procedurally handling cat animations, so these are mostly the early ideas. These animations are going to be intertwined with the game so much that we won’t be able to work on them until a decent portion of it has been made.

# Golf Cup



By using unity’s render feature to change the render order the cup can be inserted into the ground anywhere visually, without changing the ground mesh to prevent “z fighting”. The collision of each level will be made using primitive shapes.