**Fantasy World**

Dan Singer

Section 2

**Description of World:**

Outdoor fantasy environment, in the vein of *Skyrim*. This is nature-packed world with rivers, mountains, trees, and lively villagers foraging for food.

**World Exploration:**

The user can cycle between a large range of cameras, some of which are stationary and some of which follow targets. For finer control, you can adjust each camera view by clicking and dragging to rotate and scrolling to zoom.

**Steering Behavior Descriptions**

*Flocking:*

The birds are the flocking population; they continually move about the world by wandering about the sky. They’re also constrained such that they cannot leave the level or touch the ground. One bird is tasked to wander, while the others only flock so there is not too much random movement.

*Path Following:*

The foraging humans are the path followers. They’re going to collect food, and then return to the village to drop off the food. They then repeat this cycle. I designed the path in this manner such that the humans could take a long, repeatable journey.

*Flow Field Following:*

The fish in the river are the flow field followers. The flow field vectors were calculated based on a sine wave. The number of waves are specified in the inspector, and then the vectors are generated such that that number of waves fills the plane. I used cosine to get the direction of the sine wave, as cosine provides the instantaneous velocity of sine.

*Area(s) of Resistance:*

My area of resistance is the dark, cloudy patch in the sky. The flock of birds moves through this area, and the air resistance causes them to slow down.

**Resources**

I used Craig Reynold’s steering behaviors paper and *Nature of Code* to guide me in coding these behaviors. My algorithms are fairly close to these resources; Reynold’s descriptions tend to be somewhat vague, so I had to fill in the blanks with my own code, in addition to using the more in-depth descriptions in *Nature of Code*.

**Asset Resources:**

|  |  |  |
| --- | --- | --- |
| **Asset** | **Author** | **Source** |
| Simply Dynamic Clouds | Butterfly World | Unity Asset Store |
| Andan | Will B | Unity Asset Store |
| Bird Model | mathsman | clara.io |
| Goblin Ranger | Shunsuke Yamamoto | Unity Asset Store |
| Modular Fantasy Bridges | Julien Tonsuso | Unity Asset Store |
| Strong Knight | 3DMaesen | Unity Asset Store |
| Medieval Lowpoly Pack | Hit Jones | Unity Asset Store |
| Nature Volume 1 | Hedgehog Team | Unity Asset Store |
| Nature Starter Kit 2 | Shapes | Unity Asset Store |
| Root Vegetables Set | Vertex Cat | Unity Asset Store |
| RPG Character Mecanim Animation Pack Free | Explosive | Unity Asset Store |
| Sardine | Junnichi Suko | Unity Asset Store |

**Other**

I implemented Craig Reynold’s path-following algorithm rather than simple path following.