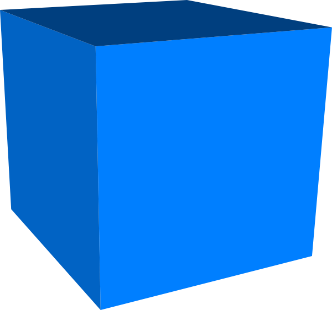
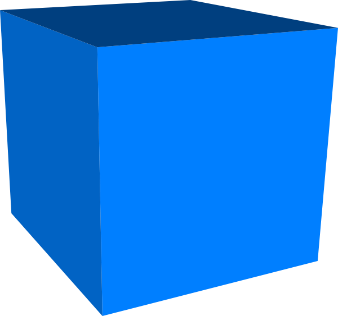
CubeBoids!

Simulating Flocking Algorithm

Game Design Document



Student: Ohmeko Ocampo

Professor: Steve Price

Date: 2 April 2017

SDSU Fall 2017

|  |  |  |
| --- | --- | --- |
| Table of Contents  [1 Game Overview](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [2 High Concept](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [3 Unique Selling Points](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [4 Platform Minimum Requirements](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [5 Competitors / Similar Titles](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [6 Synopsis](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [7 Game Objectives](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [8 Game Rules](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [9 Game Structure](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [10 Game Play](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [10.1 Game Controls](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [10.2 Game Camera](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [10.2.1 HUD](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [10.2.2 Maps](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [11 Player](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [11.1 Character](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [11.2 Metrics](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [11.3 States](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [12 Art](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [12.1 Setting](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [12.2 Level Design](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [12.3 Audio](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13 Procedurally Generated Content](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13.1 Environment](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13.2 Levels](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13.3 Artificial Intelligence NPC](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13.4 Visual Arts](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13.5 Audio](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [13.6 Minimum Viable Product (MPV)](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)  [14 Wish List](https://n-xovwktmtjsnaxyc2mwes2xu7pohqedmdm6zjw5q-2lu-script.googleusercontent.com/userCodeAppPanel)   |  | | --- | |  | | Game Development Team Members    PRODUCER  Ohmeko Ocampo  GAME DESIGNERS  Ohmeko Ocampo  PROGRAMMERS  Ohmeko Ocampo  TECHNICAL ARTISTS  Ohmeko Ocampo |

|  |
| --- |
|  |

# 1 Game Overview

Title: CubeBoids!

Platform: Windows 10

Genre: Simulation

Rating: E(Everyone)

Target: Everyone

Release date: 6 April 2017

Publisher: Meko Studios

CubeBoids! is simulation game in which you can view how the flocking algorithm works. The algorithm takes 3 simple concepts of cohesion, separation, and alignment and a little vector math to form a pretty believable simulation of birds flocking.

2 High Concept

The concept and idea behind cubeBoids is based on the book called Unity AI Game Programming – Second Edition. The algorithm will be implemented using Unity 3D game engine and contains two flocking modes: Lazy Flight and Follow The Leader. The main purpose of this game is a means of illustrating what can be accomplished with the Flocking Algorithm.

# 3 Unique Selling Points

Note: This game will not be sold, it is simply a means to illustrate the Flocking Algorithm

Nature Simulation

Nature Bird Sounds

Relaxing Cubeboid Movement.

# 4 Platform Minimum Requirements

PC Platform:

Windows 10

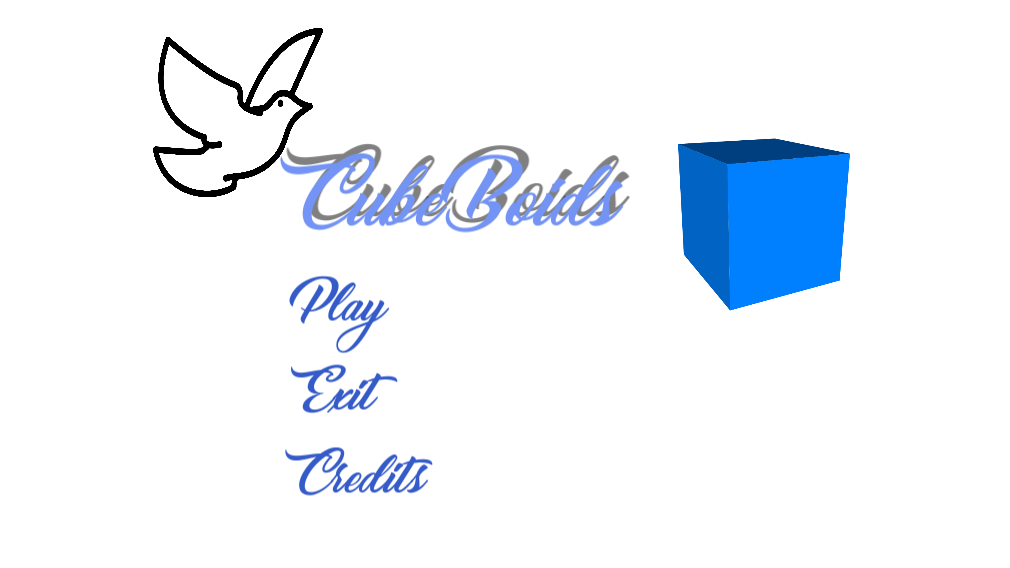


Figure 1. Start Screen of CubeBoids!

# 5 Competitors / Similar Titles

N/A

# 6 Synopsis

You are a little fledgling and you have to understand the rules that govern flocking. Will you be able to follow the flock or be left behind?

# 7 Game Objectives

No real objective, just sit back and relax. This game is designed so that you can view the Flocking Algorithm and how it works.

# 8 Game Rules

There is no physical player in the Flocking Algorithm but each CubeBoid has to follow the rules of the flocking algorithm in the Lazy Flight. Then in Follow the Leader one more rule is added and that is if the little CubeBoids see a red MotherBoid they would have to follow her. Otherwise the rules they follow are essentially weighted vectors telling them where to go based on the summed total of the cohesion, separation, and alignment vectors.

# 9 Game Structure

There is the only three possible places you can go through in this game.

Main Menu 🡪 Play 🡪 Choose Flocking Mode: Follow the Leader or Lazy Flight 🡪 Pause Menu 🡪 Return to Main Menu

Or

Main Menu 🡪 Credits 🡪 Return to Main Menu

Or

Main Menu 🡪 Exit 🡪 Choose: Either to Exit or Return to Main Menu

# 10 Game Play

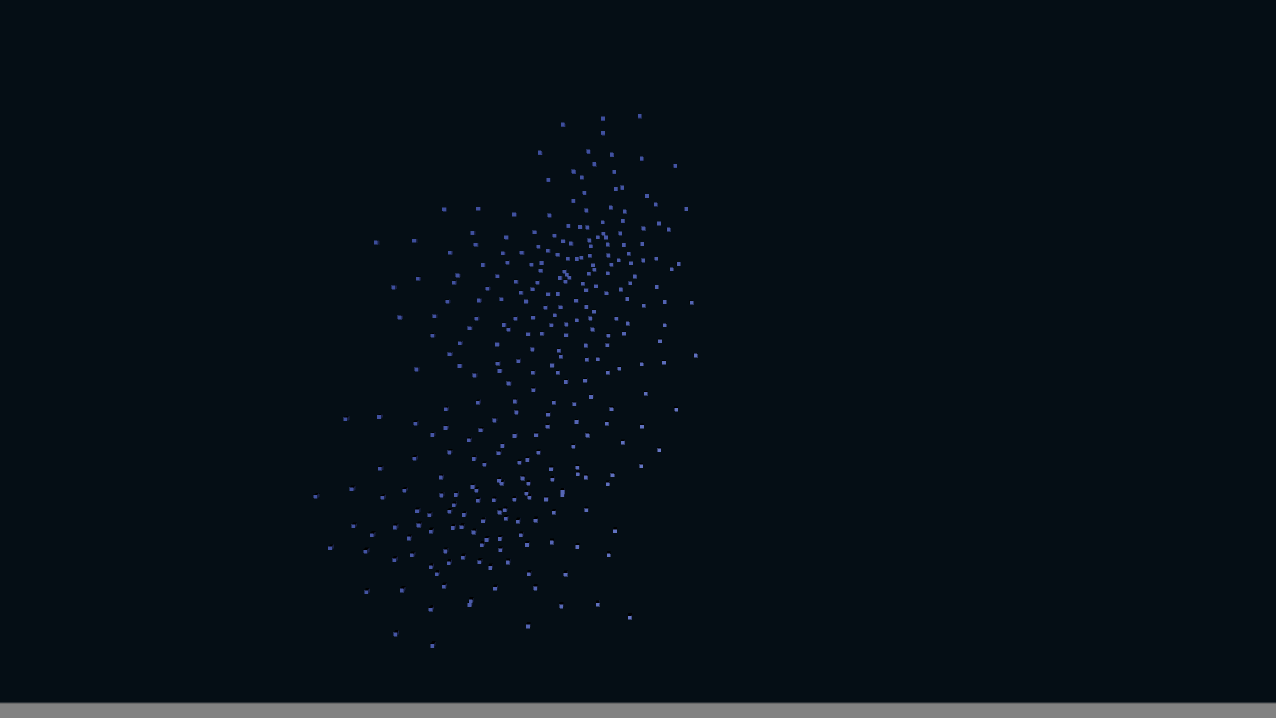


Figure 2. Screenshot of Lazy Flight Mode



Figure 3. Screenshot of Follow the Leader

## 10.1 Game Controls Windows

The only game controls here are just a left mouse clicking through the menu scene, credit scene, pause screen on the indicative text. The pause screen is activated in any of the Flocking Modes: Lazy Flight or Follow the Leader by simply pressing the Escape key on the keyboard.

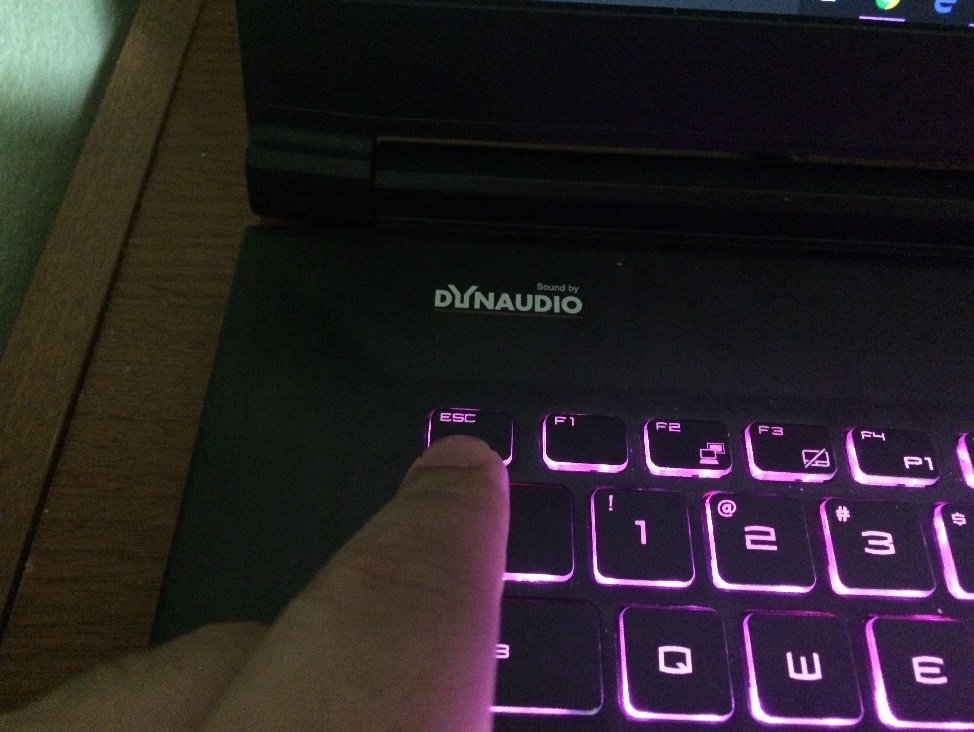


Figure 4. The Escape Key to Pull up the Pause Menu



Figure 5. Pause Menu in Action

## 10.2 Game Camera

When the game starts, the camera will be directly hovering fixed above the x and z plane. The CubeBoids will race around on the x and z plane chasing an invisible way point or if a mother bird comes along it will immediately follow her. The CubeBoids themselves are just cube object primitives built into Unity3D.

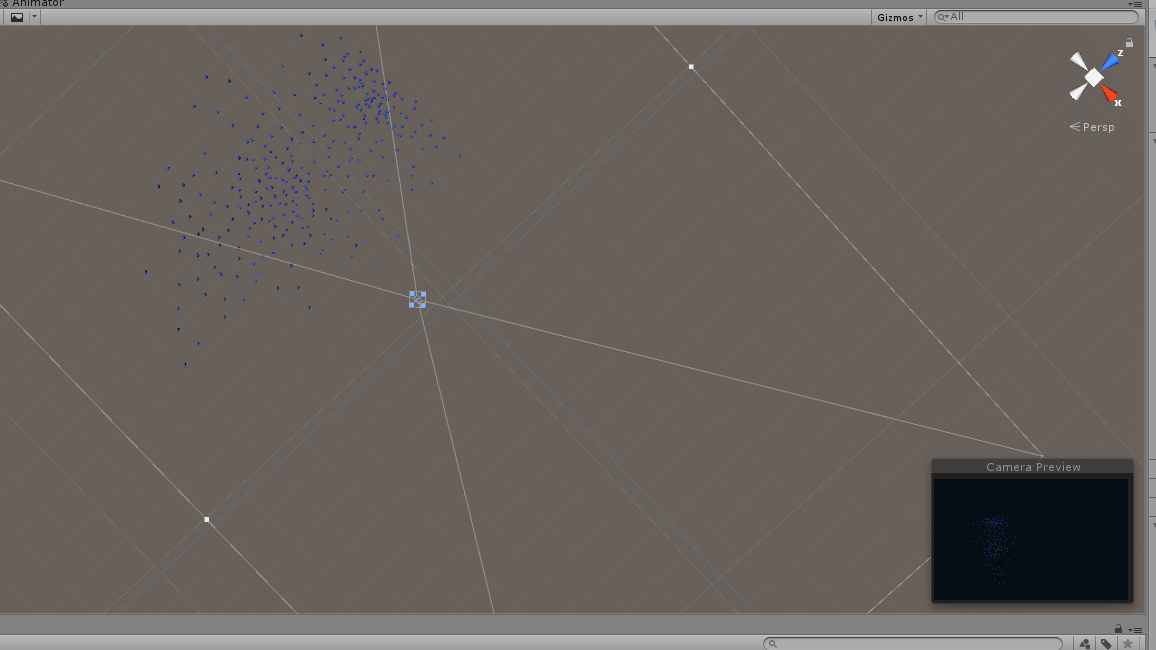


Figure 6. Camera Placement in CubeBoids!

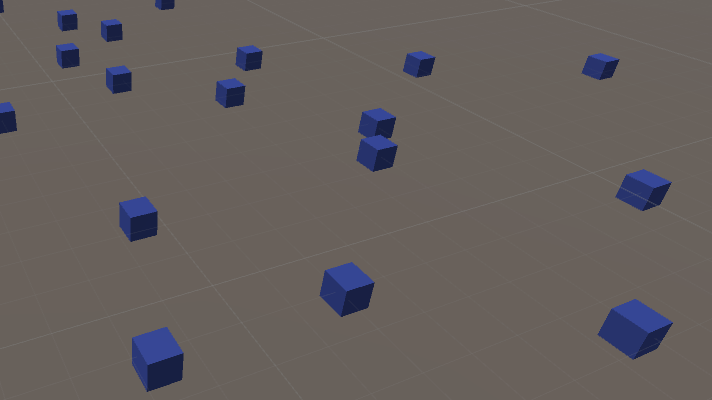


Figure 7. The CubeBoids.

### 10.2.1 HUD

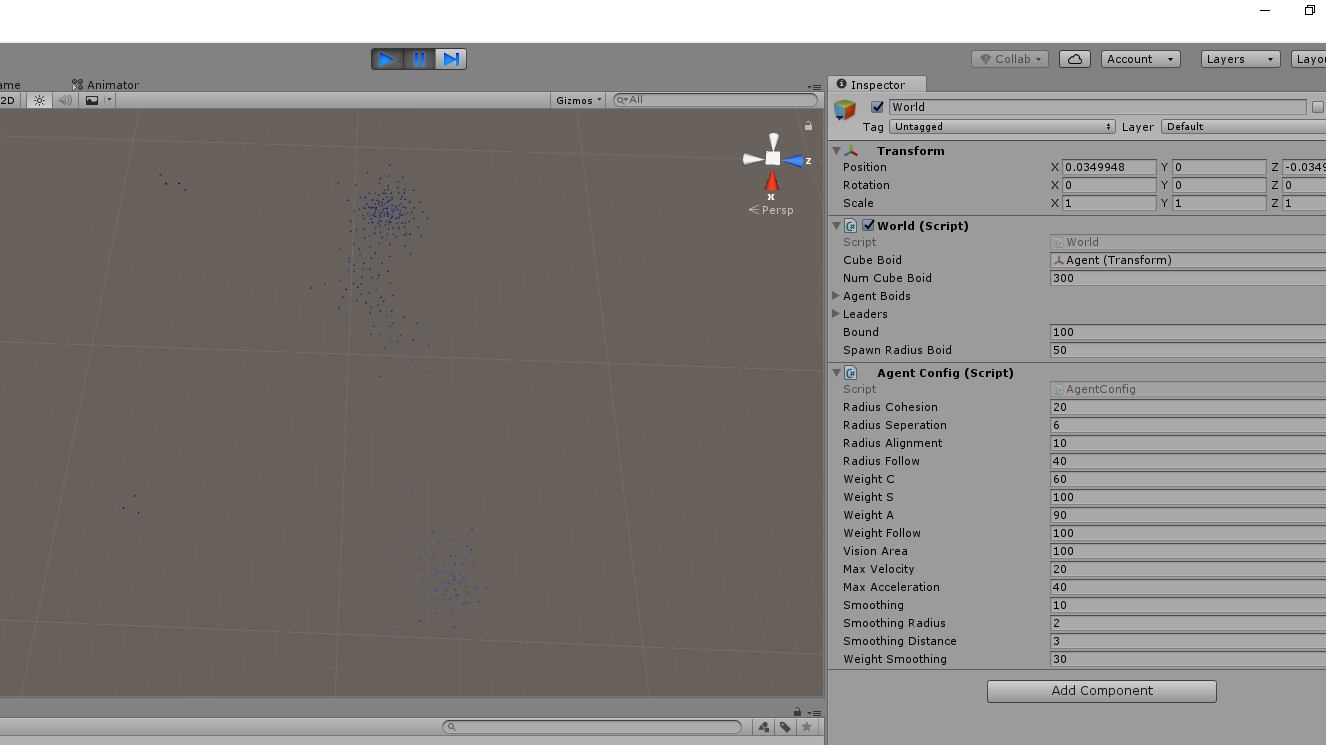
The only screen that occurs in the game scenes is the pause menu to either resume the simulation or return to main menu and select another flocking mode. There really is not HUD in this simulation game.



Figure 8. Pause Menu

### 10.2.2 Maps

The only level in the world is the level called World and it is essentially a world that is 100 by 100 Unity meters in area. It is a little bit hard to see so I have overlaid red dashes through Word’s shapes function to highlight the area in which to voids will be exploring and the blue circle highlights where the CubeBoids are located.

Figure 9. World Level as seen from above.

# 11 Player

## 11.1 Character

The main character of CubeBoids is the humble cube primitive in Unity3D. It is 0.6 by 0.6 by 0.6 in scale. Cubes will Fly!!

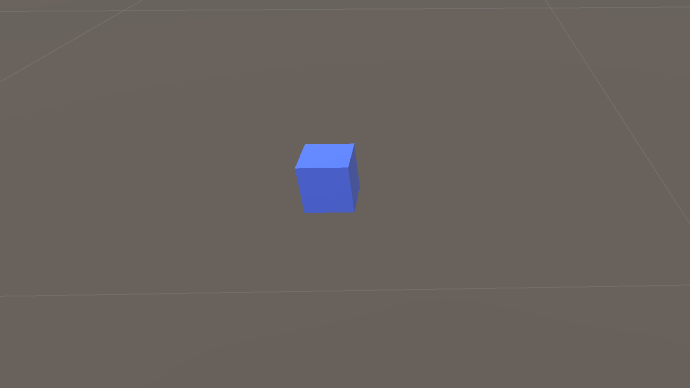


Figure 10. Picture of a Cube Primitive in Unity 3D

## 11.2 Metrics

There are a variety of metrics that sadly I have not yet created a GUI for so it will only be accessible and editable through the Unity3D inspector panel for this submission.

Here are the values that can be changed to manipulate the behavior of the CubeBoids flocking.

1. CubeBoid: <Holds Primitive Unity Cube>
2. Num Cube Boid: <Enter number of CubeBoid birds>
3. Bound: <How large an area will the CubeBoids will travel>
4. Spawn Radius Boid: <How spread out will the CubeBoid will be spawned.>
5. Radius Cohesion: <Radius that tempts bird to do Cohesion>
6. Radius Separation: <Radius that tempts the bird to do Separation>
7. Radius Alignment: <Radius that tempts the bird to do Alignment>
8. Radius Follow: <Radius that tempts the bird to follow the red SphereBoid.
9. Weight C: <How much weight is placed on doing the Cohesion tendency of Boid>
10. Weight S: <How much weight is placed on doing the Separation tendency of Boid>
11. Weight A: <How much weight is placed on doing the Alignment tendency of Boid>
12. Weight Follow: <How much weight is placed on doing the Follow tendency of the Boid>
13. Vision Area: <Degrees Boid could see other creatures in its environment>
14. Max Velocity: <Clamping of Boid’s acceleration so that it doesn’t exceed this velocity>
15. Max Acceleration: <Clamping of Boid’s acceleration so that it doesn’t exceed this acceleration>
16. Smoothing: <How smooth the Boid will fly>
17. Smoothing Radius: <The radius of the circle in front of the Boid that contains the point it follows for smooth movement>
18. Smoothing Distance: <How far the invisible point the Boid will follow is from the Boid.>
19. Weight Smoothing: <How much weight is placed on making the Boid have a smooth flight.>

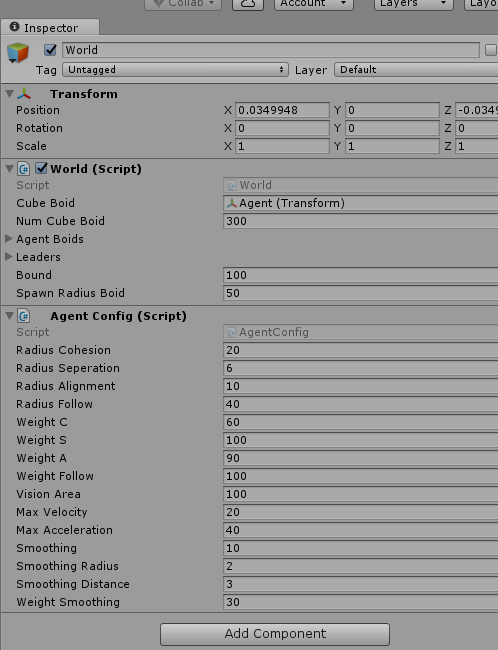


Figure 11. All the variables that control the CubeBoid and Environment Setting.

## 11.3 States

**Cohesion**: This means boids try to maintain a minimum distance with the flocks’ center.

**Alignment**: This means boids in the flock have to move in the same direction as the flock, with the same velocity.

**Separation:** This means boids has to maintain a distance with other neighboring birds in the flock to avoid collision.

**Follow:** This means boids have to follow the red sphereBoid.

# 12 Art

## 12.1 Setting

There are two scenes in this game: Scene1 named Flock1 simulates Follow the Leader and Scene2 named Flock2 simulates Lazy Flight.

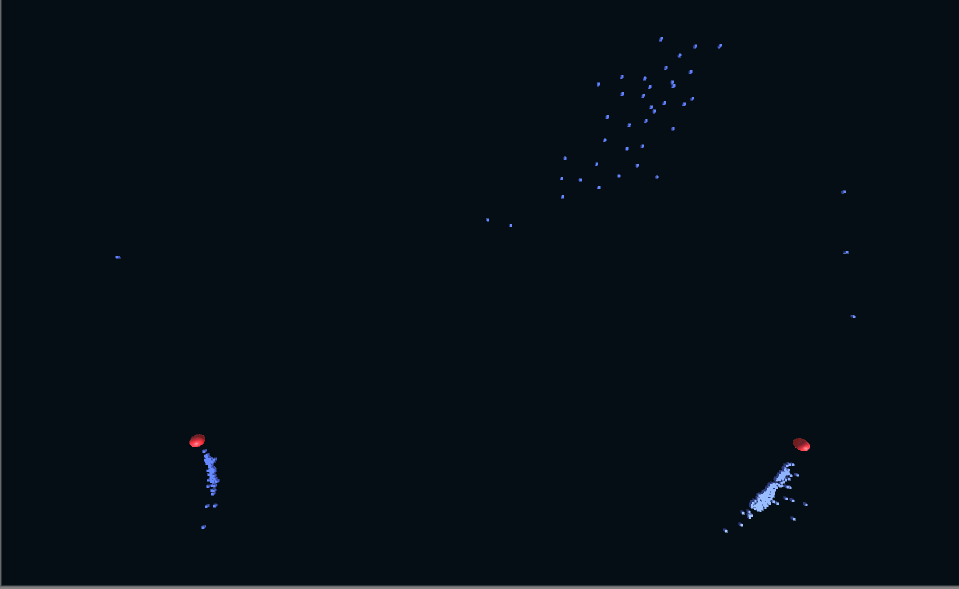


Figure 12. Overview of the Flock2 World.

## 12.2 Level Design

The design and creation of both flocking method simulation levels were done primarily with two functions named: spawnCubeBoid() in World.cs and the function boidWorldBoundaries() in Agent.cs. The function spawnCubeBoid() is in charge of spawning all the CubeBoids in a specified radius, and the boidWorldBoundaries() function is in charge of corralling all the CubeBoids in a specified area on the x-z plane. The CubeBoids can travel all around this area as they wish, but once they get to the boundary they are “warped” to the other side.

Both simulation run forever, and the player must exit out via the pause menu to get back to the main menu.

## 12.3 Audio

|  |  |  |
| --- | --- | --- |
| Name | Category | Description |
| BirdSoundsForKids.mp3 | Flock1,2,Main Menu, Credits Scene Music | Plays and loops. |

The background music of different birds chirping helps immerse the player in the simulation. I found the perfect sound using a YouTube video sound clip called BirdSoundsForKids.mp3.

# 13 Procedurally Generated Content

## 13.1 Environment

As was stated in the art section of this document, the only thing procedurally generated here is the instantiation and “random” placement of the CubeBoids on the x-z plane.

## 13.2 Levels

There are two levels excluding the credit scene and the main menu scene. The player can sit back and relax and watch either watch Flocking Mode 1: Lazy Flight or Flocking Mode 2: Follow the Leader.

# 14 Wish List

Add Another Flocking Mode

If I had given myself more time I would have tried to add and implement the Circle a Tree Mode.

Added a graphical means of changing the weights of cohesion, separation, and alignment.

If I had given myself more time I would have tried to add a GUI system so that the player can choose the importance of which rule to follow.