Rene Gajardo

Thong Tran

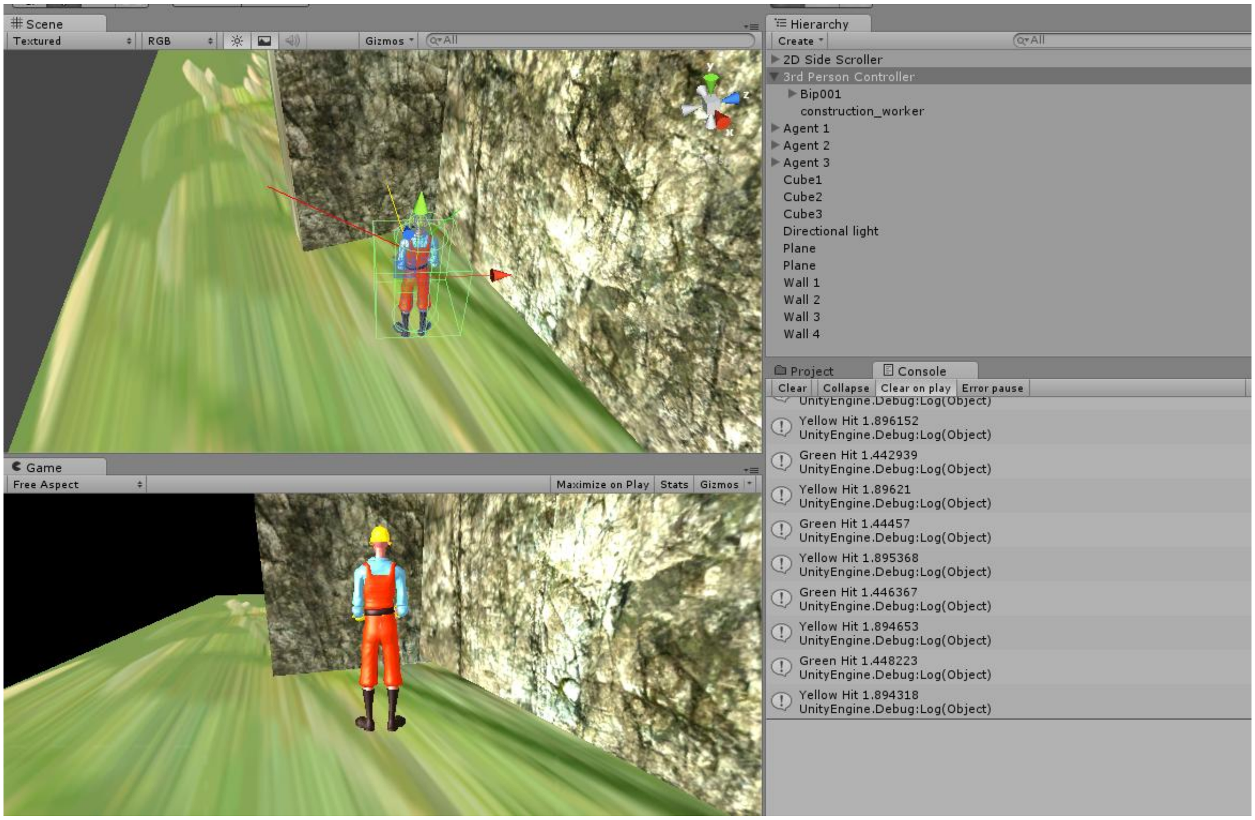
2/11/13

CAP4053

Assignment 1

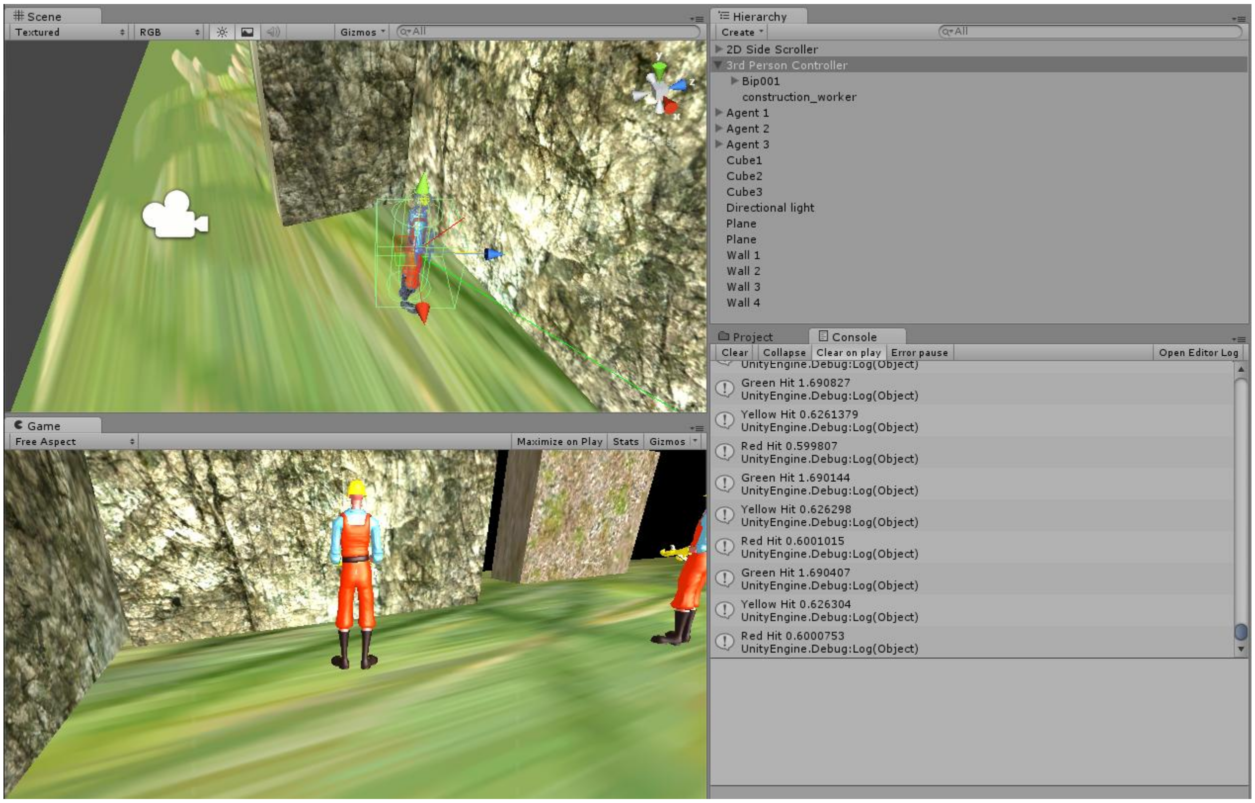
We decided on the Unity3D engine for our class project. The Unity3D engine runs off both JavaScript and C# languages as to accommodate for different programmers. The free to download engine serves the purposes of the course and allows one to develop game in 3D, and for most gaming and mobile platforms available today. There are also a lot tutorials and documentation available online. Additionally setting up the game world is simple and straightforward.

We programmed the rangefinders in Unity by implementing feelers of certain length. When the feeler would collide with an object we defied as a wall it would return the distance from its origin to where it collided with the wall. If the feeler does not hit a wall it remains silent.



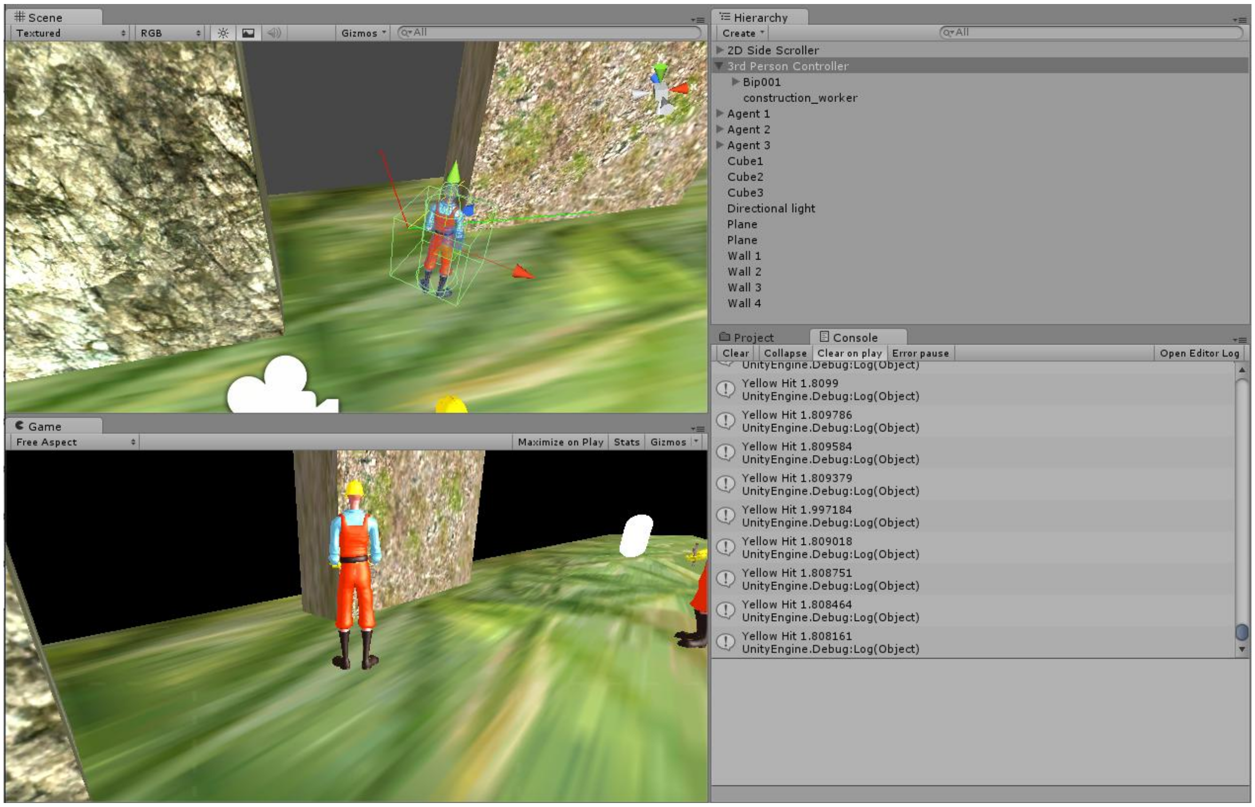
```
UnityEngine.Debug:Log(Object)
Yellow Hit 1.896152
UnityEngine.Debug:Log(Object)
Green Hit 1.442939
UnityEngine.Debug:Log(Object)
Yellow Hit 1.89621
UnityEngine.Debug:Log(Object)
Green Hit 1.44457
UnityEngine.Debug:Log(Object)
Yellow Hit 1.895368
UnityEngine.Debug:Log(Object)
Green Hit 1.446367
UnityEngine.Debug:Log(Object)
Yellow Hit 1.894653
UnityEngine.Debug:Log(Object)
Green Hit 1.448223
UnityEngine.Debug:Log(Object)
Yellow Hit 1.894318
UnityEngine.Debug:Log(Object)
```

Figure 1 – Central and right feeler connecting with wall and returning the distance from the wall. Left feeler does not interact with any wall.



```
UnityEngine.Debug:Log(Object)
Green Hit 1.690827
UnityEngine.Debug:Log(Object)
Yellow Hit 0.6261379
UnityEngine.Debug:Log(Object)
Red Hit 0.599807
UnityEngine.Debug:Log(Object)
Green Hit 1.690144
UnityEngine.Debug:Log(Object)
Yellow Hit 0.626298
UnityEngine.Debug:Log(Object)
Red Hit 0.6001015
UnityEngine.Debug:Log(Object)
Green Hit 1.690407
UnityEngine.Debug:Log(Object)
Yellow Hit 0.626304
UnityEngine.Debug:Log(Object)
Red Hit 0.6000753
UnityEngine.Debug:Log(Object)
```

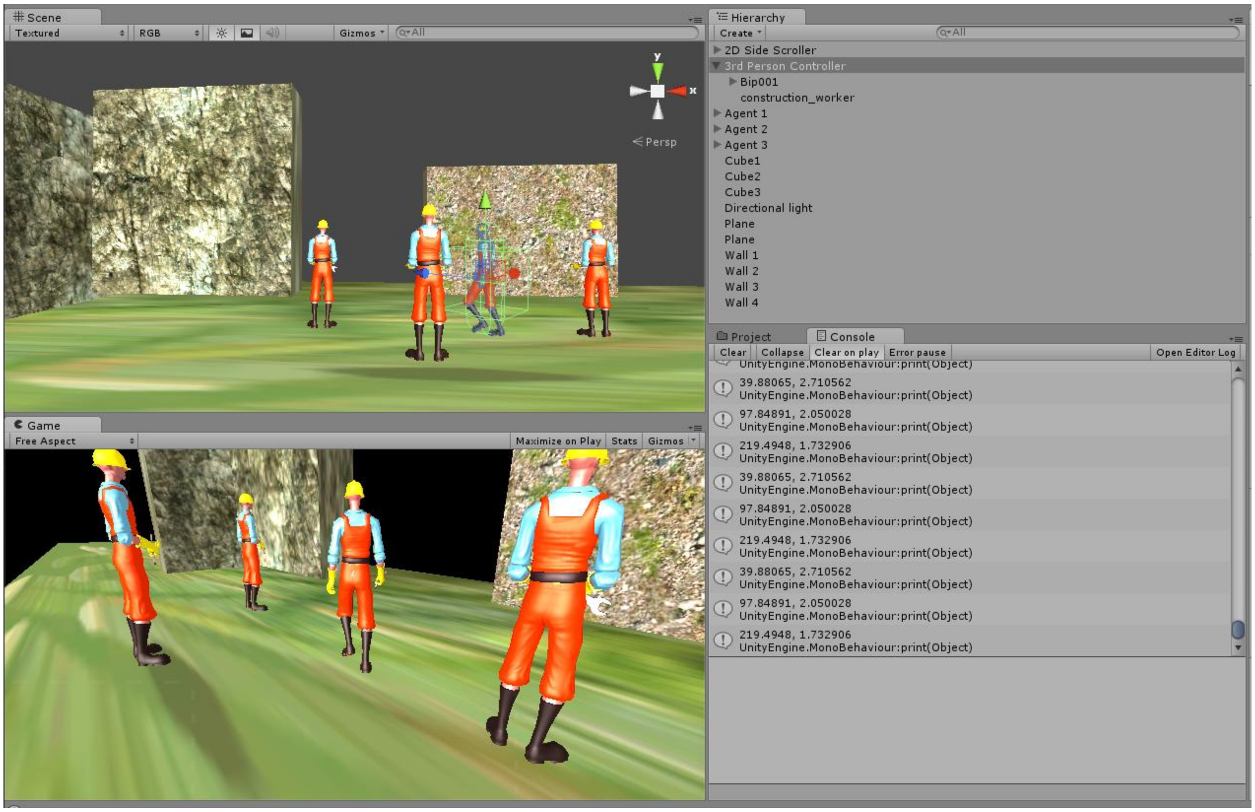
Figure 2 – All three feelers colliding with a wall and returning the distance from it.



```
UnityEngine.Debug:Log(Object)
Yellow Hit 1.8099
UnityEngine.Debug:Log(Object)
Yellow Hit 1.809786
UnityEngine.Debug:Log(Object)
Yellow Hit 1.809584
UnityEngine.Debug:Log(Object)
Yellow Hit 1.809379
UnityEngine.Debug:Log(Object)
Yellow Hit 1.997184
UnityEngine.Debug:Log(Object)
Yellow Hit 1.809018
UnityEngine.Debug:Log(Object)
Yellow Hit 1.808751
UnityEngine.Debug:Log(Object)
Yellow Hit 1.808464
UnityEngine.Debug:Log(Object)
Yellow Hit 1.808161
UnityEngine.Debug:Log(Object)
```

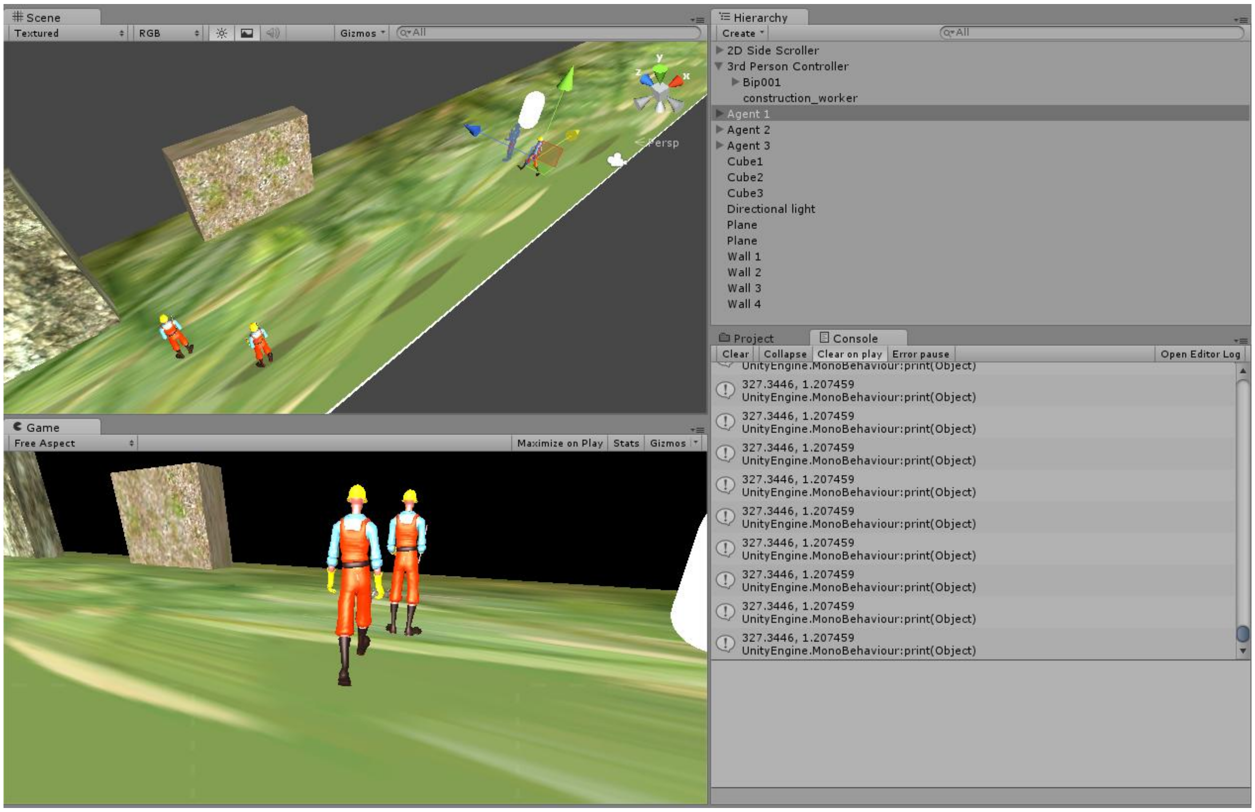
Figure 3 – Only the central feeler interacts with a wall and returns the distance from the wall.

We programmed the adjacent agent sensor by counting all agents in the world and then checking to see if any were in range. Then we calculate the distance from the main agent and the angular location of the agent with respect to the direction the main agent is facing by calculating the inverse tangent of the X and Z difference from the main agent and the agent in range. Then subtracting the Euler angle of Y, then modding the answer by 360, and finally find its absolute value.



```
UnityEngine.MonoBehaviour:print(Object)
39.88065, 2.710562
UnityEngine.MonoBehaviour:print(Object)
97.84891, 2.050028
UnityEngine.MonoBehaviour:print(Object)
219.4948, 1.732906
UnityEngine.MonoBehaviour:print(Object)
39.88065, 2.710562
UnityEngine.MonoBehaviour:print(Object)
97.84891, 2.050028
UnityEngine.MonoBehaviour:print(Object)
219.4948, 1.732906
UnityEngine.MonoBehaviour:print(Object)
39.88065, 2.710562
UnityEngine.MonoBehaviour:print(Object)
97.84891, 2.050028
UnityEngine.MonoBehaviour:print(Object)
219.4948, 1.732906
UnityEngine.MonoBehaviour:print(Object)
```

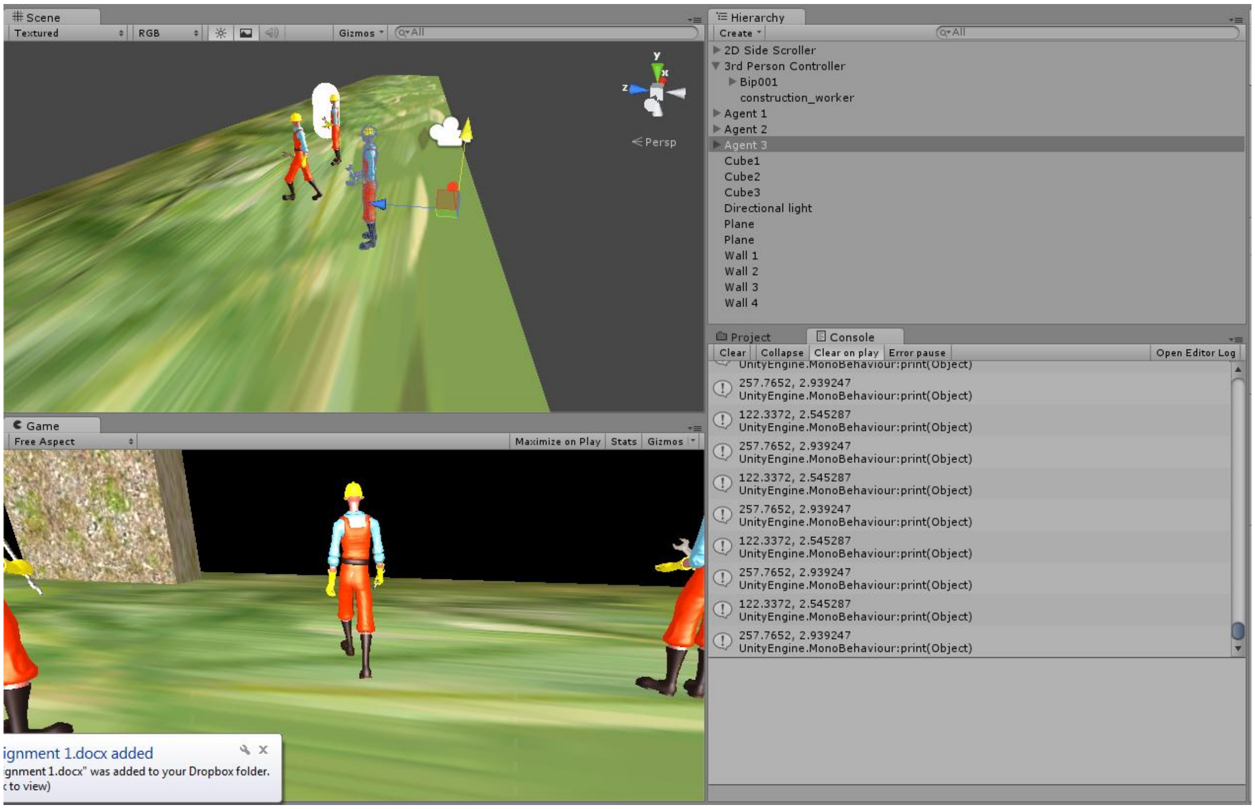
Figure 4 – Three agents surround the running main agent.



2D Side Scroller
3rd Person Controller
Bip001
construction_worker
Agent 1
Agent 2
Agent 3
Cube1
Cube2
Cube3
Directional light
Plane
Plane
Wall 1
Wall 2
Wall 3
Wall 4

```
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
327.3446, 1.207459
UnityEngine.MonoBehaviour:print(Object)
```

Figure 5 – An agent is slightly to the right of the main agent.



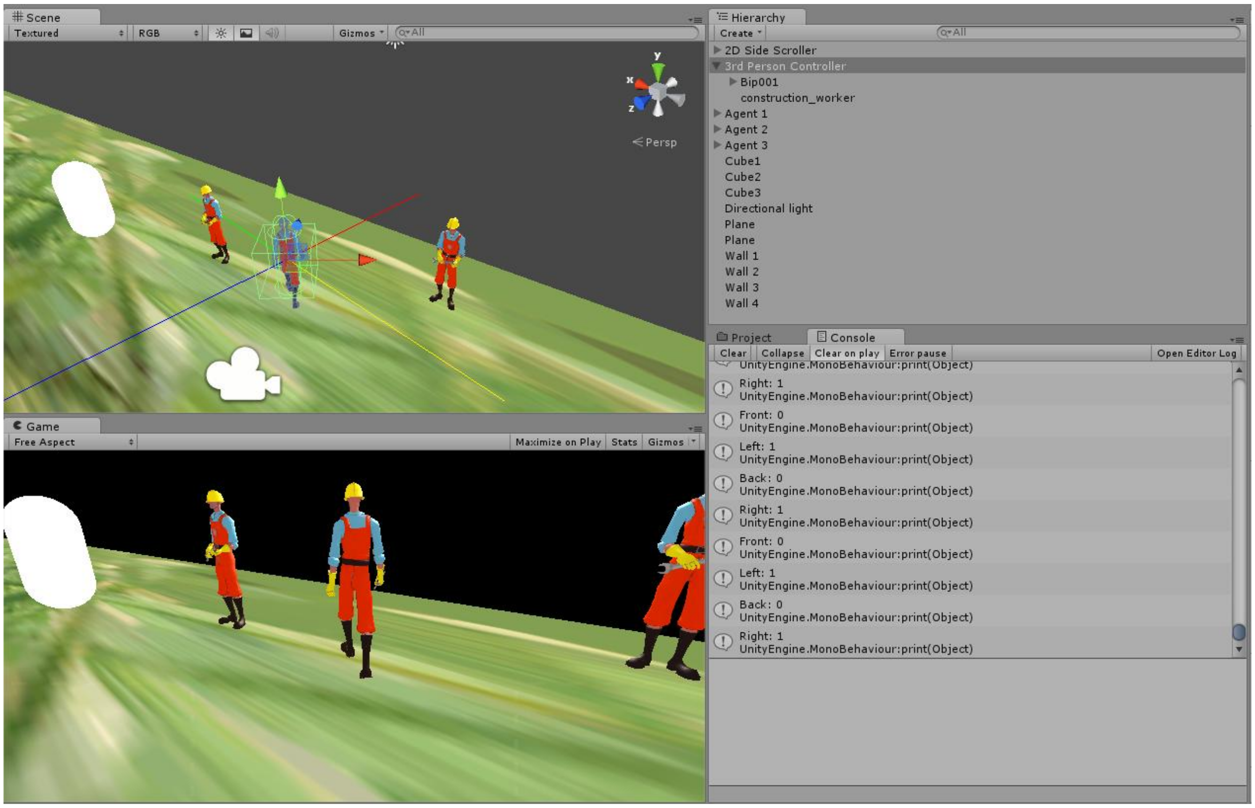
2D Side Scroller
3rd Person Controller
Bip001
construction_worker
Agent 1
Agent 2
Agent 3
Cube1
Cube2
Cube3
Directional light
Plane
Plane
Wall 1
Wall 2
Wall 3
Wall 4

```
UnityEngine.MonoBehaviour:print(Object)
257.7652, 2.939247
UnityEngine.MonoBehaviour:print(Object)
122.3372, 2.545287
UnityEngine.MonoBehaviour:print(Object)
257.7652, 2.939247
UnityEngine.MonoBehaviour:print(Object)
122.3372, 2.545287
UnityEngine.MonoBehaviour:print(Object)
257.7652, 2.939247
UnityEngine.MonoBehaviour:print(Object)
122.3372, 2.545287
UnityEngine.MonoBehaviour:print(Object)
257.7652, 2.939247
UnityEngine.MonoBehaviour:print(Object)
122.3372, 2.545287
UnityEngine.MonoBehaviour:print(Object)
257.7652, 2.939247
UnityEngine.MonoBehaviour:print(Object)
```

ignment 1.docx added
ignment 1.docx" was added to your Dropbox folder.
k to view)

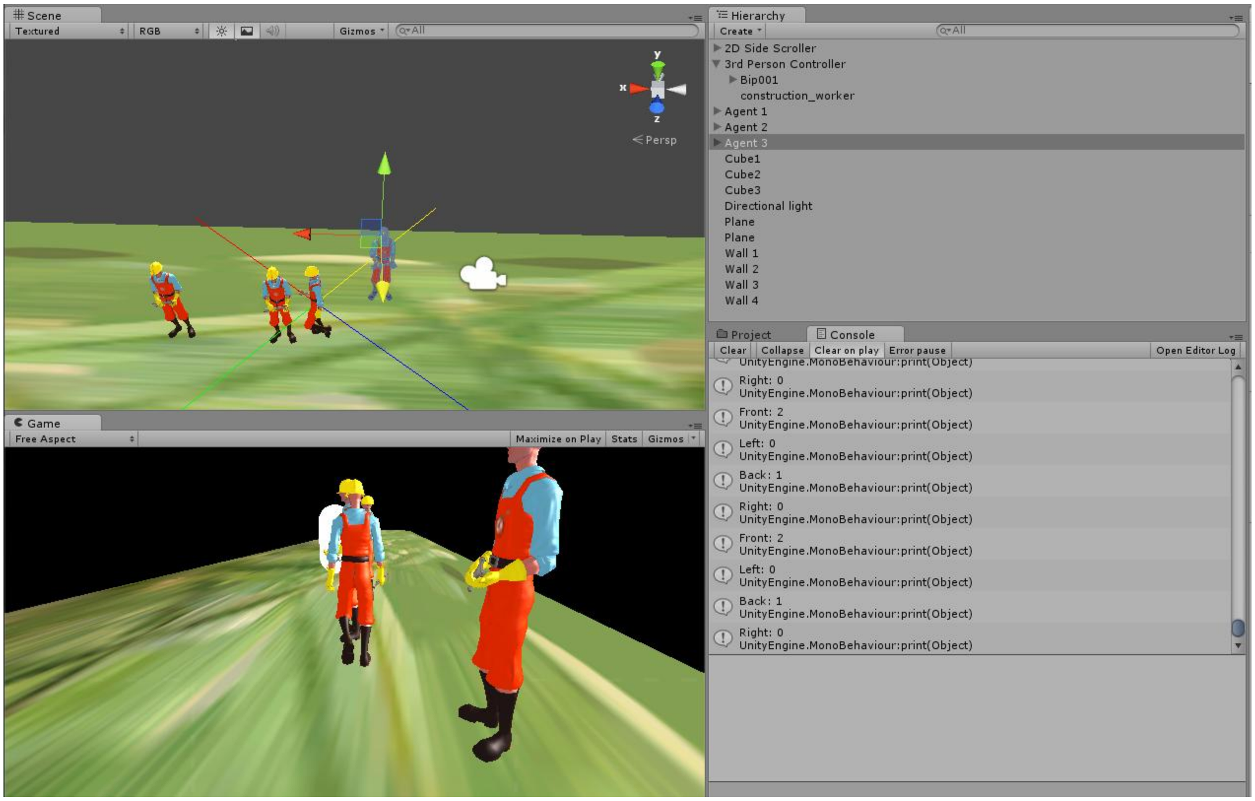
Figure 6 – One agent is to the left of the main agent while another is to its right.

We programmed the pie-slice radar by incorporating four rays that defined the edges of the four zones. We then reused the agent location code from the radar to determine if agents were inside one of the pie-slices. If they are inside the pie, we then calculate in quick quadrant, front, back, left, or right, they are in. Then we would increase a counter for that quadrant. After all agents are accounted for we would print out all four counters.



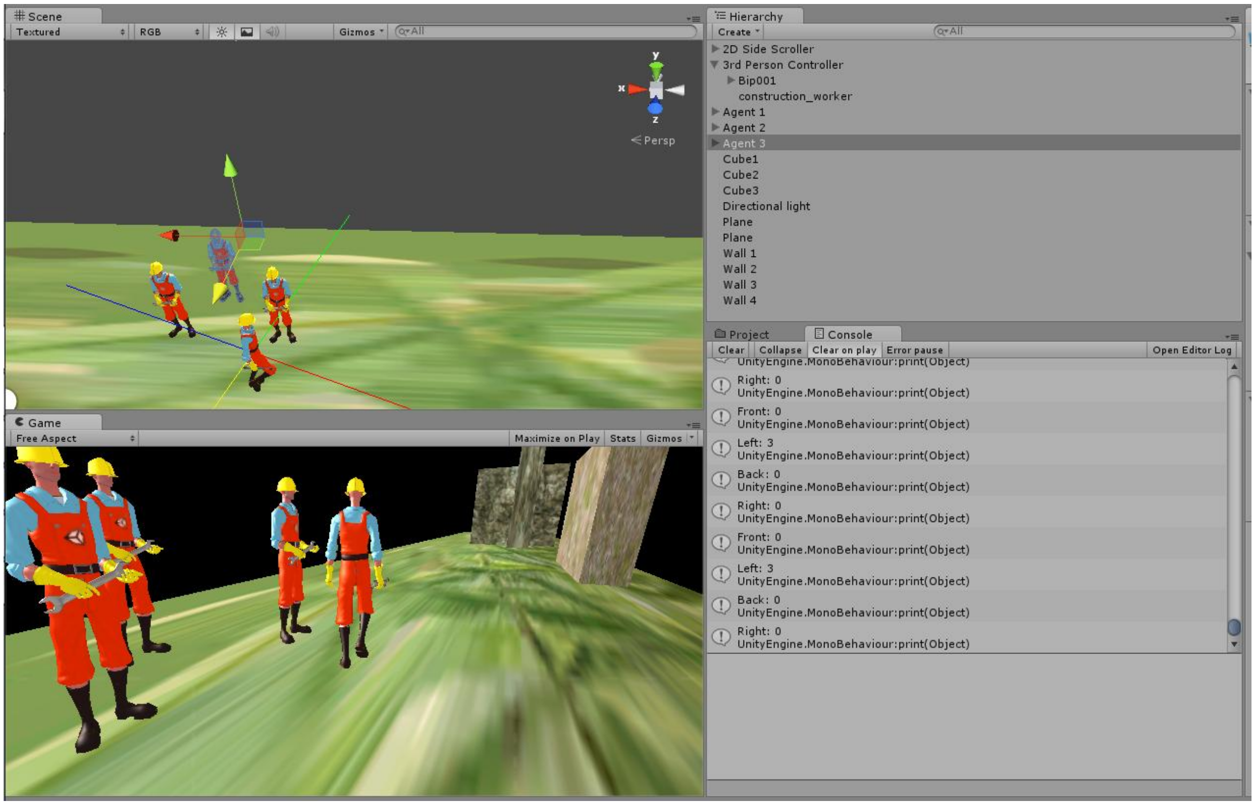
```
UnityEngine.MonoBehaviour:print(Object)
Right: 1
UnityEngine.MonoBehaviour:print(Object)
Front: 0
UnityEngine.MonoBehaviour:print(Object)
Left: 1
UnityEngine.MonoBehaviour:print(Object)
Back: 0
UnityEngine.MonoBehaviour:print(Object)
Right: 1
UnityEngine.MonoBehaviour:print(Object)
Front: 0
UnityEngine.MonoBehaviour:print(Object)
Left: 1
UnityEngine.MonoBehaviour:print(Object)
Back: 0
UnityEngine.MonoBehaviour:print(Object)
Right: 1
UnityEngine.MonoBehaviour:print(Object)
```

Figure 7 – One in the right pie-slice and another on the left pie-slice.



```
UnityEngine.MonoBehaviour:print(Object)
Right: 0
UnityEngine.MonoBehaviour:print(Object)
Front: 2
UnityEngine.MonoBehaviour:print(Object)
Left: 0
UnityEngine.MonoBehaviour:print(Object)
Back: 1
UnityEngine.MonoBehaviour:print(Object)
Right: 0
UnityEngine.MonoBehaviour:print(Object)
Front: 2
UnityEngine.MonoBehaviour:print(Object)
Left: 0
UnityEngine.MonoBehaviour:print(Object)
Back: 1
UnityEngine.MonoBehaviour:print(Object)
Right: 0
UnityEngine.MonoBehaviour:print(Object)
```

Figure 7 – Two agents in front of the main agent and a third behind it.



```
UnityEngine.MonoBehaviour:print(Object)
Right: 0
UnityEngine.MonoBehaviour:print(Object)
Front: 0
UnityEngine.MonoBehaviour:print(Object)
Left: 3
UnityEngine.MonoBehaviour:print(Object)
Back: 0
UnityEngine.MonoBehaviour:print(Object)
Right: 0
UnityEngine.MonoBehaviour:print(Object)
Front: 0
UnityEngine.MonoBehaviour:print(Object)
Left: 3
UnityEngine.MonoBehaviour:print(Object)
Back: 0
UnityEngine.MonoBehaviour:print(Object)
Right: 0
UnityEngine.MonoBehaviour:print(Object)
```

Figure 9 – Three agents to the left of the main agent.

We have learned to code slightly in Javascript. We have also learned that calculations of angles can be very tricky and confusing at times. We should have meet more regularly to facilitate the learning of the engine and Javascript. We realized that invisible walls can be made by using planes and rotating them vertically instead of creating large cubes and rendering them invisible.