## Details of the Scene's Behavior

There are three entities in the scene:

1- **Spheres**

These are “Exhibit Objects” in the scene which are being visited by agents.

2- **Standing Positions**

Number of standing positions besides spheres for agents, so they can see if there is not much crowd and line of sight is not blocked. Availability of places are controlled by them. They are invisible in the scene.

3- **Capsules**

These are “Agents” which visit spheres. They are controlled by behavior tree technique to move around exhibition rooms intelligently.

## Explanation

All the capsules have a movement AI script which is implemented using behavior tree agents technique. Before creating behavior tree, two blackboards are created:

1- **Agent’s own blackboard**

This blackboard is used to trigger nodes in the tree. First of all it checks if the agent is reached to destination or not, if agent is not reached to destination then it triggers node that sets destination of the agent. Combined with the shared blackboard’s variable that checks if the destination is not already occupied it triggers nodes according to current state. After that another node in sequence is played to check the remaining distance from destination.

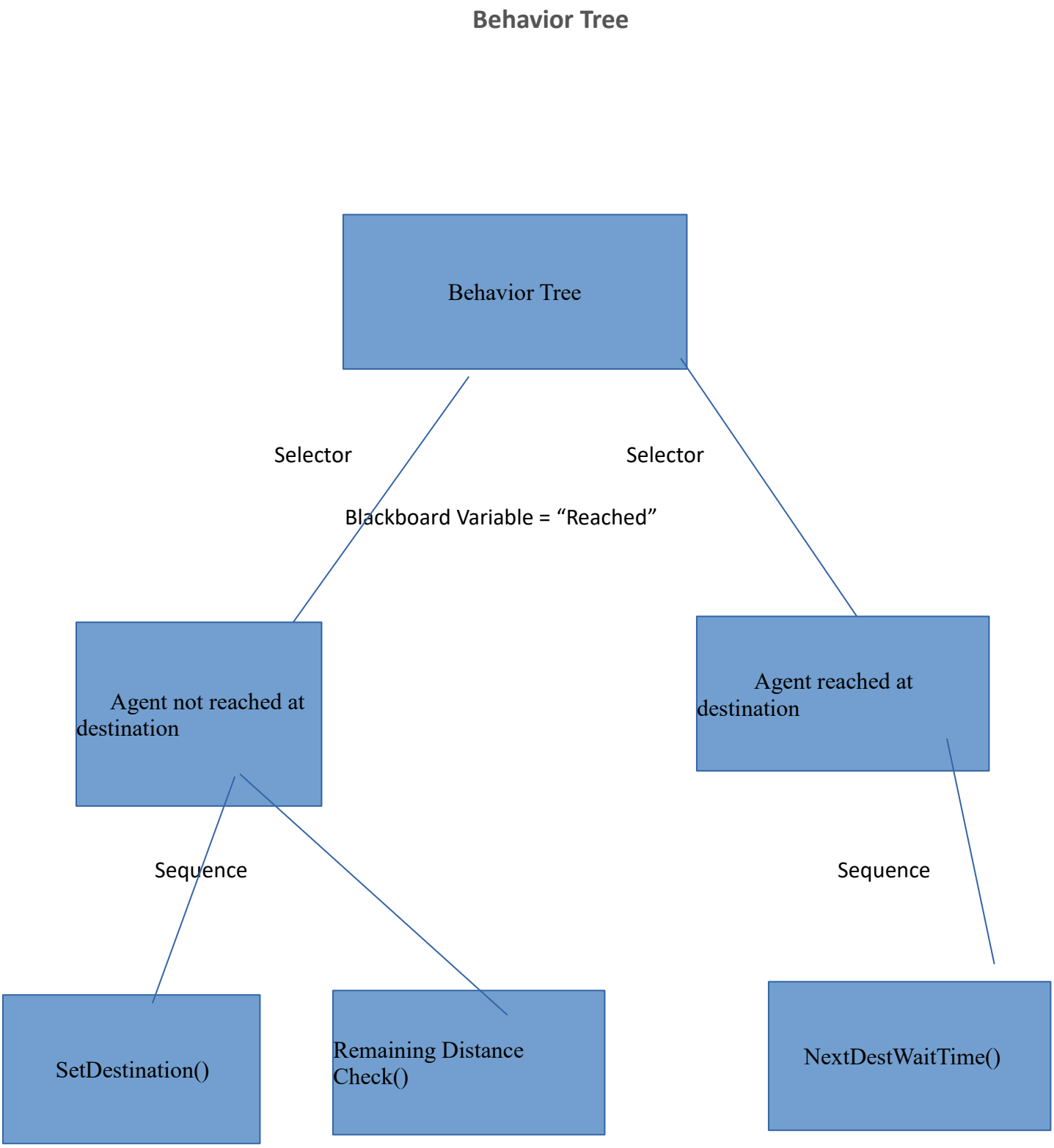
If the destination is set then it waits for the agent to reach destination. Where it starts another node that let it wait for random amount of time. After time is complete the agent checks if there is another exhibit in current room or another room to be visited. If yes, then the cycle starts again from triggering first node that checks the availability of place in next exhibit.

When all the exhibits are visited in all of the rooms. It goes to exit location and deactivates itself.

2- **Shared Blackboard**

This is used as a communication between agents to check which agent is at what position. If an exhibit is full then through this blackboard agent knows that it must wait until there is a free space to go. This blackboard is used with the combination of an agent’s self blackboard to make it intelligent enough to move through the places.

For agents to move through the rooms and not collide with static and dynamic objects Unity's NavMesh Agent component is used. All the floor and walls are baked for navigation of agents. This AI keeps an agent move wisely and make it's own path dynamically.



## Code Explanation

-A behavior tree is created in the following function call. While creating

→ behaviorTree = CreateBehaviourTree();

- While creating behavior tree we pass it a blackboard which is parented to shared blackboard.

→ sharedBlackboard = UnityContext.GetSharedBlackboard("SharedData");
ownBlackboard = new Blackboard(sharedBlackboard, UnityContext.GetClock());
return new Root(ownBlackboard, ……

-This CreateBehaviourTree function has two **Selectors**. They are conditioned with Own Blackboard's "**Reached**" variable. First Selector has two actions in sequence. And second selector has one action in sequence.

→ new Selector(
new BlackboardCondition("Reached", Operator.IS_EQUAL, false,
Stops.IMMEDIATE_RESTART,

new Sequence(

new Action(() => SetDestination()) { Label = "Setting Destination" },
new Action(() => RemainingDistanceCheck()) { Label = "Checking Remaining Distance" }

→ new Selector(

new BlackboardCondition("Reached", Operator.IS_EQUAL, true,
Stops.IMMEDIATE_RESTART,

new Sequence(
new Action(() => NextDestWaitTime()) { Label = "Wait for random time" }

-These **Actions** are the main nodes of trees that actually do everything in the game.

-There is a **co-routine** called in the **NextDestWaitTime** action.

→ IEnumerator waiter()
{
float waitTime = UnityEngine.Random.Range(5f, 15f);
yield return new WaitForSeconds(waitTime);

```
        print("Waited for " + waitTime);
        ClearUsedData();
    }
```

It waits for random time (between 5-10 seconds) allowing agent to wait at the exhibit and look at it for that period of time. After wait it allows agent to reset it's destination again.

## Functions

**SetDestination:**

Sets destination of an agent according to both blackboards(shared, owned).

**RemainingDistanceCheck:**

Checks the distance remaining from destination.

**NextDestWaitTime:**

When a destination is reached it allows agent to wait for a random time here. For that it calls a co-routine.

**ClearUsedData:**

When an agent starts a new destination and completes its whole course the data is reset so other agents can see if the position is occupied or not and if the exhibition course is completed then the agent is deactivated.


References:

1.https://github.com/meniku/NPBehave
2.https://www.youtube.com/watch?v=JZbesN2_-fE
3.https://www.youtube.com/watch?v=0jKz9WqF24c
4.https://www.youtube.com/watch?v=YCMvUCxzWz8