Possible Update Documentation

By: Fall 2022 BioRube Team

Table Of Contents

Generalization Updates **3**
1. Roamer.cs 3
1.1. Roaming(GameObject Obj) 3
1.2. MoveToDock(GameObject Obj, GameObject dock) 3
2. GTP_CmdCtrl.cs 3
2.1. Fixed update() 3
3. Spawner.cs 3
3.1. ThisIsARotatableObject() 3
4. UIControl.cs 3

Efficiency Updates **4**
1. Roamer.cs 4
1.1. Roaming(GameObject Obj) 4
2. ATPPathfinding.cs 4
2.1. FixedUpdate() 4
3. PKABMovement.cs/ PKAMovement.cs 4
3.1. Kinase movement bug 4

Readability updates **5**
1. GTP_CmdCtrl.cs 5
1.1. FixedUpdate() 5

Generalization Updates

1. Roamer.cs
    1.1. Roaming(GameObject Obj)
        1.1.1. Make it so roaming will account for more collisions than just the cell membrane. (Use the collider type to define the movement or path)
    1.2. MoveToDock(GameObject Obj, GameObject dock)
        1.2.1. Generalize pathing around objects so that it accounts for more than the inner cell wall. (Use the collider type to define the movement or path)
2. GTP_CmdCtrl.cs
    2.1. Fixed update()
        2.1.1. use the pathfinding algorithm in the roamer class.
3. Spawner.cs
    3.1. ThisIsARotatableObject()
        3.1.1. Generalize this function so that snaps can occur to other objects than cell wall and nucleus wall. (You could do this by checking what the object is colliding with.)
        3.1.2. Attach a GameObject Reference to each snapped object so that any movements along that object can be generalized. (The solution above makes this change easier.)
        3.1.3. Generalize snap and rotate (View each object's collision as a collection of nodes and not as a whole object and this becomes much easier.) (For Rotation getting a line perpendicular to the tangent of the location will solve this issue)
        3.1.4. Once you finish this Generalize object traversal on an object.(Will not be in spawner)
        3.1.5. After that Remove lines:117, 120-124, 136, 27,163,171,178 in Spawner.cs to re allow dropping snappable objects anywhere. (Per Client Requests)
4. UIControl.cs
    4.1. Updating the screen resolution behind the scene
        4.1.1. Look at how to ask for the specs on a monitor from the hardware through the OS and use that to create a generalized beginning resolution.
        4.1.2. Use Unity Documentation with Screen.SetResolution class for width height and screen modes preferred.
        4.1.3. With RectTransform you can change the location of the anchors so that the camera adequately has a view of the game area (Use the size of the png in hardware to determine the size of the play area… Use the prefab size of the background.)
        4.1.4. With use of the Display class, the user is able to tell the resolution of the particular displays with systemHeight, systemWidth and rendering as well

with renderingHeight and renderingWidth. SetParams and SetRenderingResolution are also there for rendering the sizes and positions of users' displays.

Efficiency Updates

1. Roamer.cs
    1.1. Roaming(GameObject Obj)
        1.1.1. Find a way to get the center without brute force.
2. ATPPathfinding.cs
    2.1. FixedUpdate()
        2.1.1. Needs to drop off of a target after a certain amount of time
        2.1.2. MAKE SURE that the target is reallowed to be targeted by other ATP after the time limit is up.
3. PKABMovement.cs/ PKAMovement.cs
    3.1. Kinase movement bug
        3.1.1. Not sure how to solve the kinase bugging around a cell wall
        3.1.2. May have something to do with the bug that allows infinite backwards momentum.
            3.1.2.1. Link to a glitch that is basically the same as an example
            Alberto's Beginner Manual Superswim Tutorial!

Readability updates

1. GTP_CmdCtrl.cs
    1.1. FixedUpdate()
        1.1.1. This needs to be fully rewritten; it is too many if statements and needs to be broken into parts.