2015 Summer BioRubeBot Project Progression

TEAM GOALS AND ACCOMPLISHMENTS

Our primary project development goals for the 2015 summer term included:

1. Generate random movement for the following objects:
   - G-protein
   - Kinase
   - GDP
   - GTP (formerly the phosphate object)
   - Transcription Regulator
2. Establish G-protein interaction with its interactive signaling objects:
   - The Receptor Phosphate (populated by an ATP at the legs of an activated receptor)
   - GDP
   - GTP
3. Development of an interactive tutorial
4. Investigate and implement a new pathing algorithm

Having successfully completed our goals we were also able to include an animated home screen (main menu) and pan and zoom functionality.

Accomplishing these goals required slight modifications to the 'ReceptorLegScript' and 'Spawner' scripts, and the addition of the following scripts:

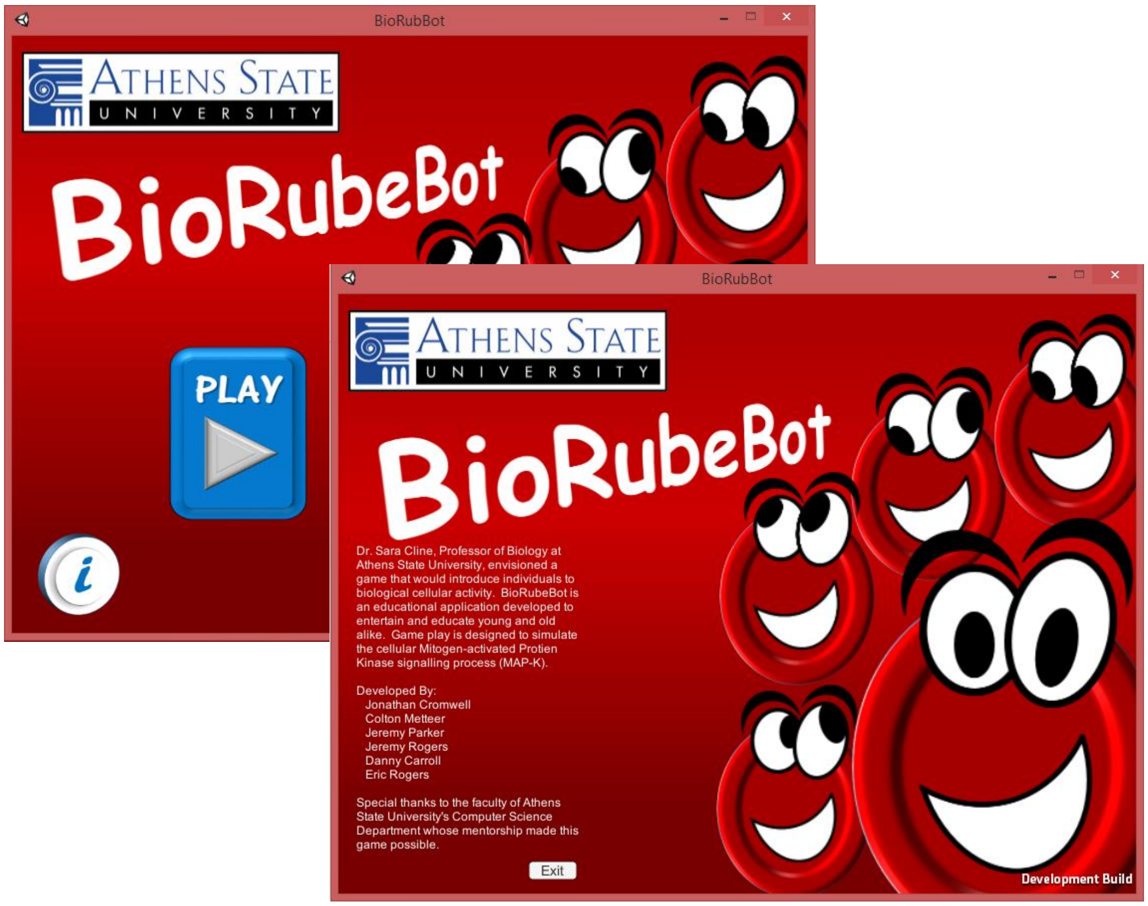
- HomeScreen.cs
- movePhosphate.cs
- moveKinase.cs
- moveT_Reg.cs
- G_ProteinCmdCtrl.cs (G-Protein Command and Control)
- GTP_CmdCtrl.cs (GTP Command and Control)
- PanAndZoom.cs

The following information will provide detailed information regarding the project's progression over the 2015 summer semester.

GRAPHICAL USER INTERFACE (GUI) MODIFICATIONS

Home Screen (HomeScreen.cs)

The home screen received a complete makeover now including cartoonish cell proliferation. Also added was an information icon that when selected displays a brief summary of the games development along with a list of contributing developers. Future developers may research and add vertical scrolling functionality, and further legalese may be added as necessary.



Dr. Sara Cline, Professor of Biology at Athens State University, envisioned a game that would introduce individuals to biological cellular activity. BioRubeBot is an educational application developed to entertain and educate young and old alike. Game play is designed to simulate the cellular Mitogen-activated Protien Kinase signalling process (MAP-K).

Developed By:
Jonathan Cromwell
Colton Metteer
Jeremy Parker
Jeremy Rogers
Danny Carroll
Eric Rogers

Special thanks to the faculty of Athens State University's Computer Science Department whose mentorship made this game possible.

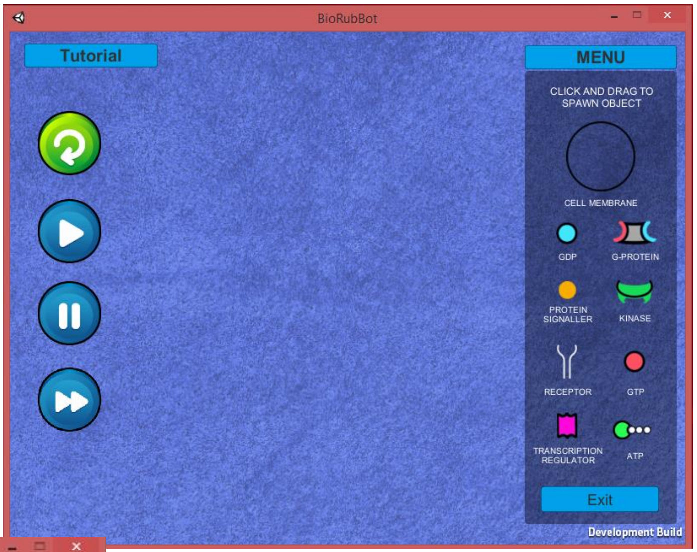
Home Screen

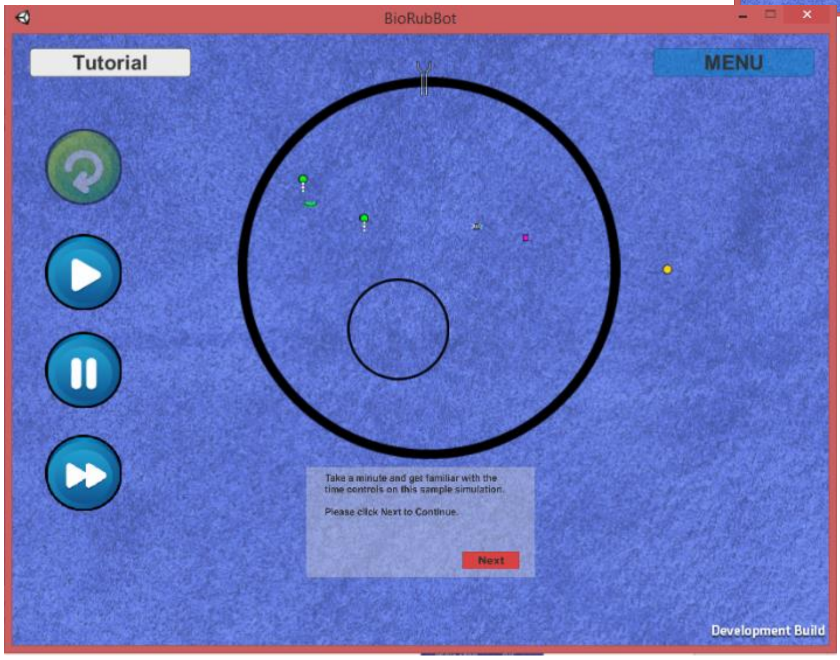
A redesign of the start screen now includes animated cell proliferation and an information pane.

2015 Summer BioRubeBot Project Progression

"The Sandbox"

The sandbox retains its original design, with only minor alterations to its drop down menu and the addition of a tutorial. The objects in the drop down menu have been reorganized to not reflect the actual order of interactions in the signaling process.

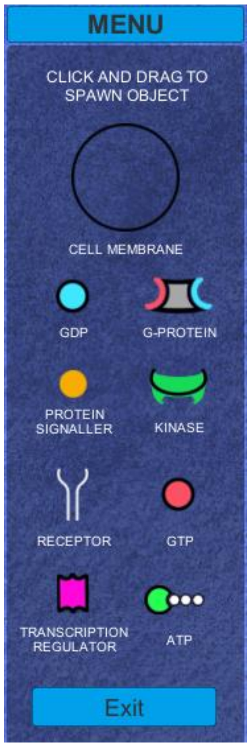




Take a minute and get familiar with the time controls on this sample simulation.

Please click Next to Continue.

The addition of an interactive tutorial provides new users with an overview of game controls and game play simulation

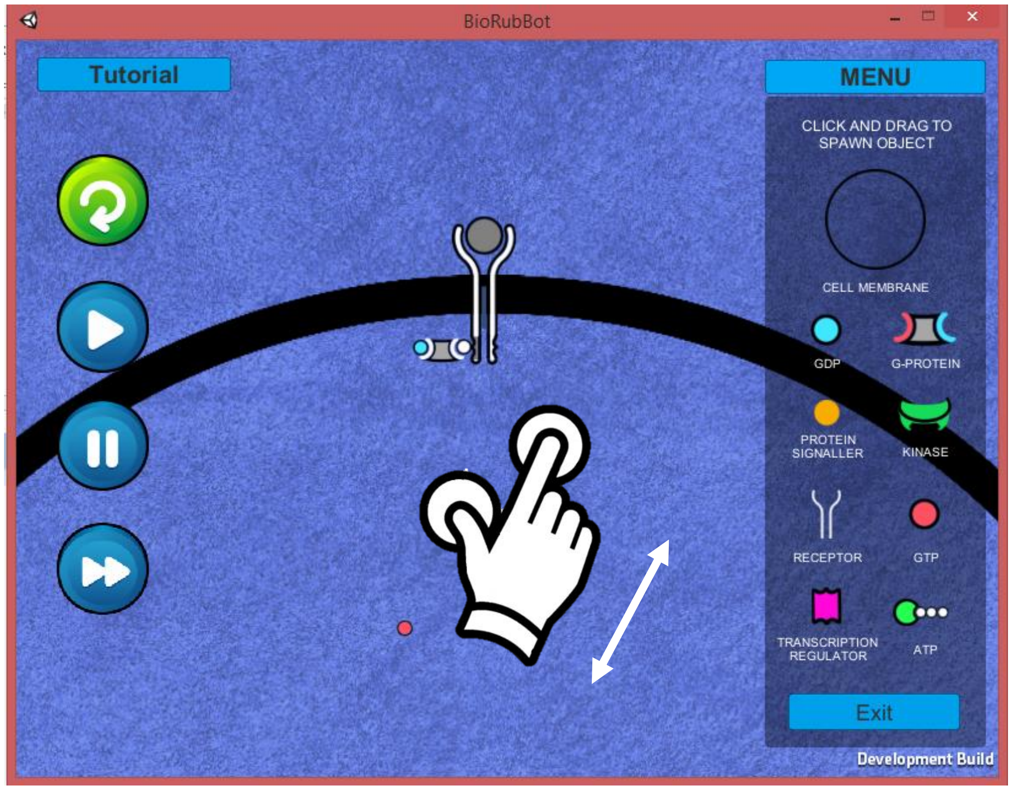


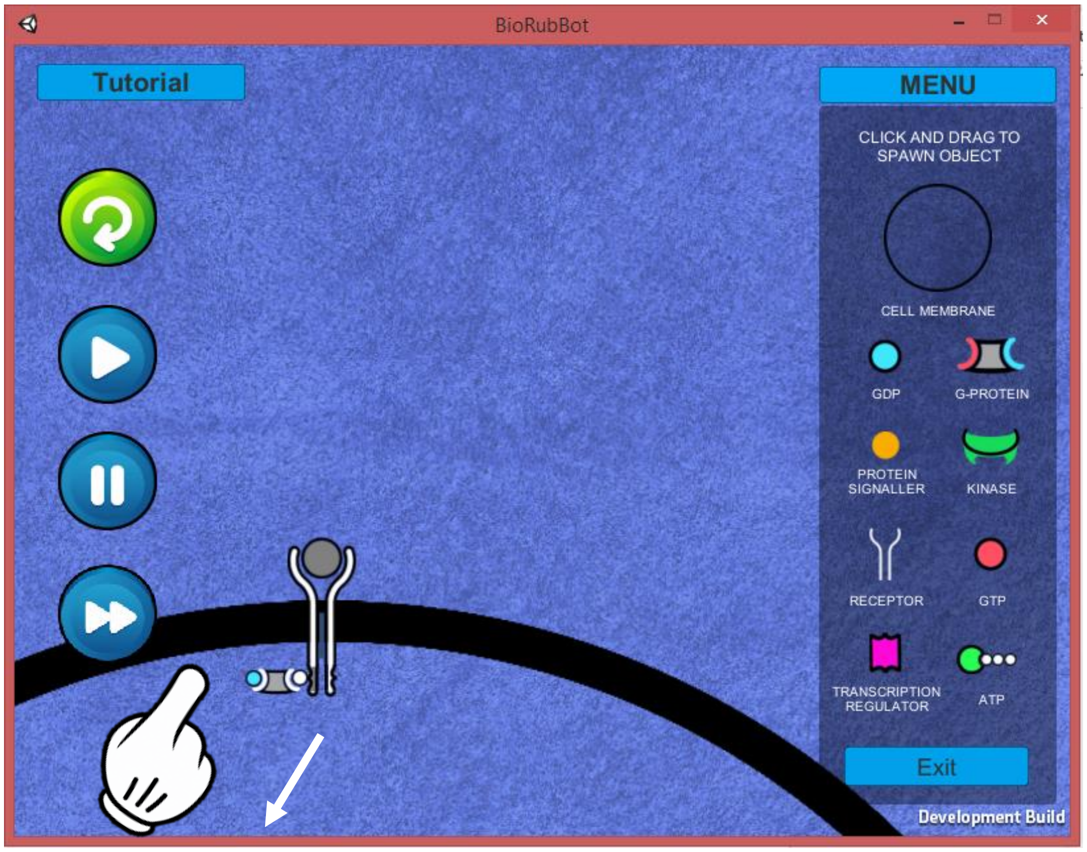
The drop down menu objects are now arranged in a manner non-representative of the actual MAP-K pathway signaling process

Pan and Zoom Functionality (PanAndZoom.cs)

A new pan and zoom script was developed to provide users the ability to zoom in on individual objects through the familiar 'pinch to zoom' method. Current zoom functionality ranges from x1 to x3.

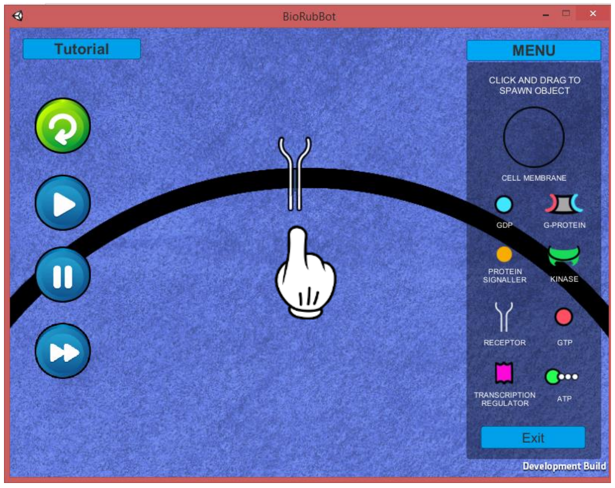
Panning is performed with the usual touch and drag/push commonly used with touch applications allowing users to move about to view all areas of the simulation.





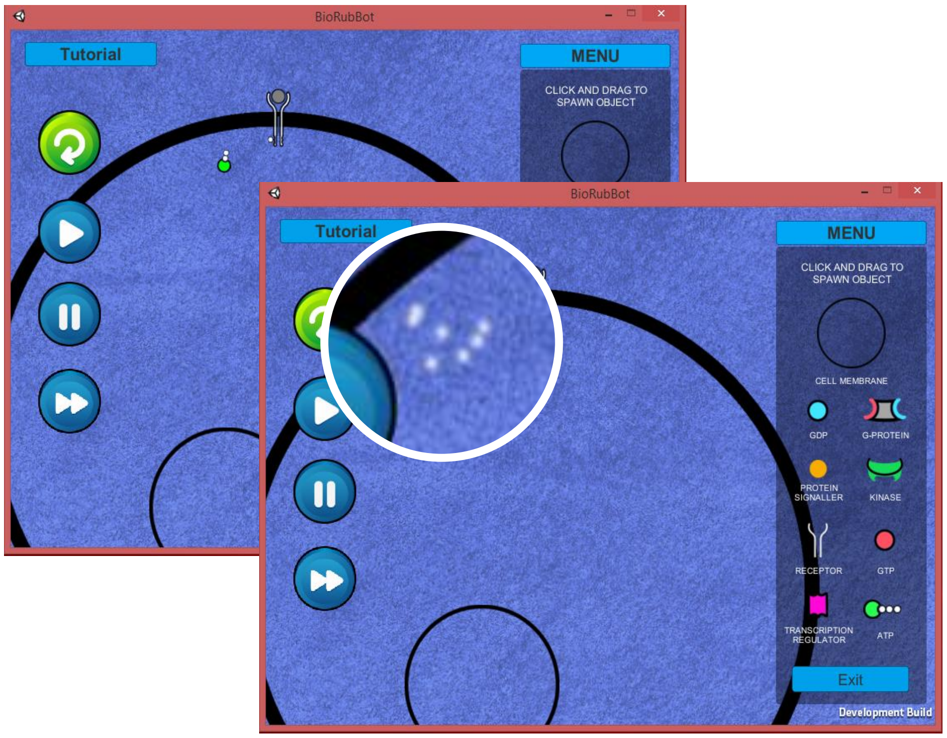
Object Spawning (Spawner.cs)

Also updated was the positioning of game objects, relative to the players finger, while placing them on the canvas. A y-axis offset is now included such that when the objects are being dragged to the canvas, they appear above the players finger and can be placed with higher precision than was previously available. This is particularly useful when attempting to position the receptor object onto the cell membrane with sufficient space for unobstructed interaction within the cell:



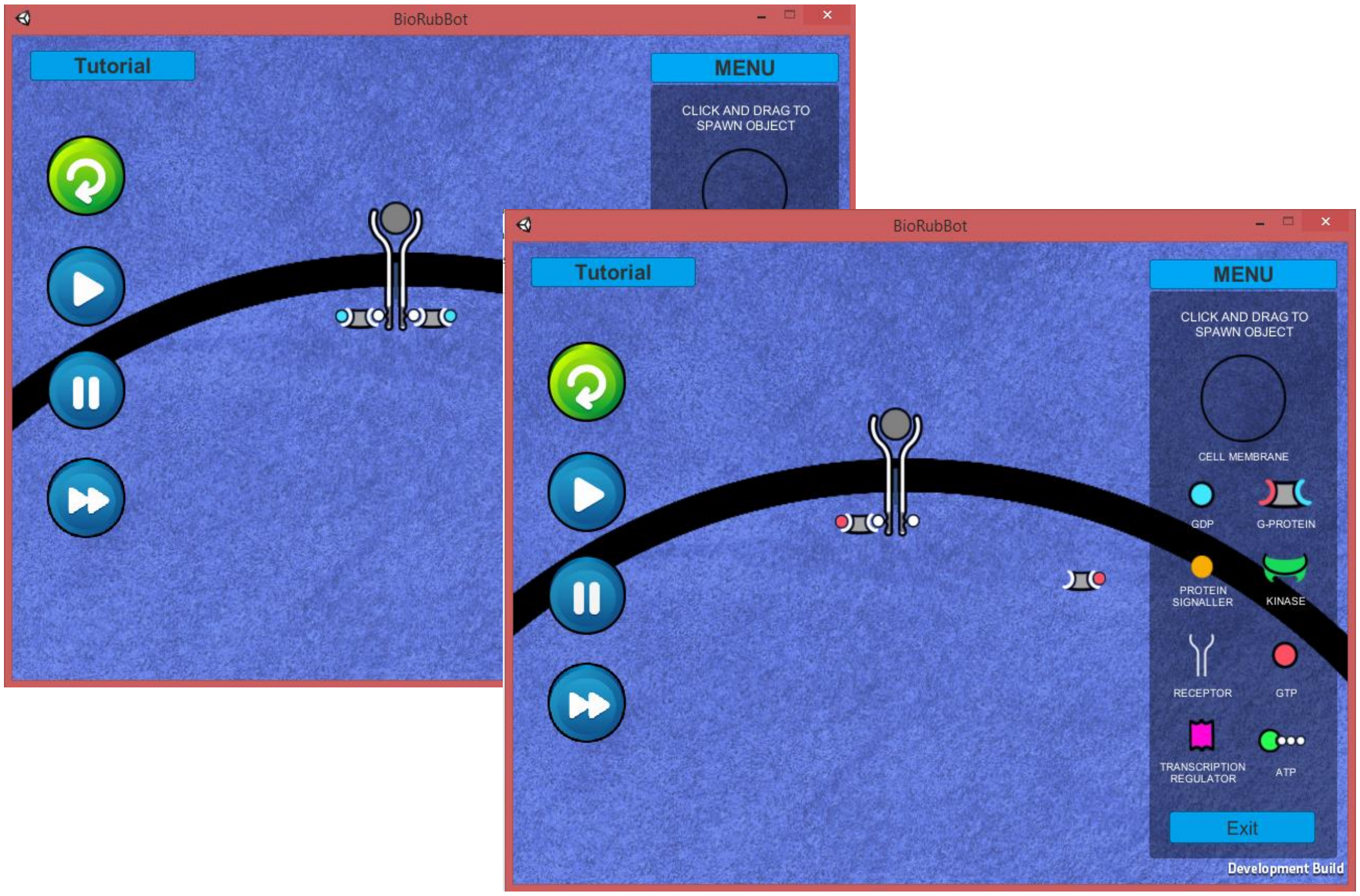
The Expending of ATP and GDP (Modified ReceptorLegScript.cs and G_ProteinCmdCtrl.cs)

One of our objectives was to figure a way to rid the sandbox of expended molecules. A particle effect was added to simulate the idea of an objects energy being expended:



G-Protien Interaction (G_ProteinCmdCtrl.cs)

The most challenging of goals this semester was the coding of the G-protein interactions through the signaling process. The colors of the g-protein 'sockets' were changed to white. The colored 'sockets' were thought to make the game less challenging. The g-protein now roams about until a receptor phosphate target is detected. Once detected, one g-protein object will target that phosphate. It will then bind with the receptor phosphate and release the GDP object. A roaming GTP object will target a G-protein that has released its GDP. Once a GTP has bound with a G-protein, the G-protein will 'undock' from the receptor phosphate and is free to roam. At this point we met our requirements for the semester and no other object interaction was developed.



Methods and Scripts:

The 'Roam()' Method

One of the most widely used methods throughout the project, the Roam() method generates random movement for each of the following objects:

- G-Protein
- GTP
- GDP
- Kinase
- Phosphate
- Transcription Regulator (T-Reg)

Roam takes advantage of Unity's Rigidbody2D method 'addForce' applying force to an object in a defined direction:

- public void AddForce(Vector2 force, ForceMode2D mode = ForceMode2D.Force);

By assigning a random (x,y) vector coordinate, we are able to apply a force in random directions at a constant interval through FixedUpdate(). To further randomize the movement, developers can modify the object's rigidbody parameters to give each object a more unique movement.

Script Modifications:

<u>ReceptorLegScript.cs</u>

Lines 24-25: Disable ATP collider while dropping off a phosphate

Lines 34-35: Enable the ATP collider once phosphate dropped off

Lines 38-43: Change receptor leg tags (referenced in G_ProteinCmdCtrl.cs)

Line 44: Added call to IEnumerator co-routine 'Explode'

Lines 47-65: Added 'Explode' to destroy ATP after dropping phosphate at the receptor

<u>Spawner.cs</u>

Line 18: Added a static bool varible to flag when player is spawning an object and is not panning

Lines 37 & 51: Set panning to false when spawning and true once object has spawned

G-Protein and GTP Command and Control Overview:

Algorithm:

```
Look for a target
    If found
        Is it the closest target
        Is another targeting object closer
        If not – call dibs
        else – roam about
    if not found – roam about
```

The algorithm and the code associated with it work just fine, but there is a more efficient way to accomplish this task. It is not necessary for a targeting object to be the closest to a target. If there is a target, and this targeting object is the first to detect it, let it call dibs.

The way the algorithm is currently coded, an array of targets is stored and the targeting object traverses the array looking for the closest target. Once it has found its closest target, the target then checks if it has a closer targeting object.

If we wish to find the closest targeting object to a target, have the call made by the target and eliminate the need for the second traversing of the array, or better yet, just let the first targeting object to spot a target, target that object.

With the semester coming to a close, we have decided to leave the script as is and leave it for future teams to decide how to proceed.

Suggested Future Modifications and Functionality Updates:

- Game Paused – Allow user to maneuver objects (that were improperly placed) to their proper place

- Receptor Transmembrane – add a color indication of the receptor passing through the cell membrane to provide the user with a visual indication that 'something is supposed to happen here.'

- Change the GDP and GTP nucleotides to resemble ADP and ATP respectively only with a square body followed by a series of round phosphates currently used with ATP.

- Address cell membrane permeation by the protein signaler molecule (occurs when a receptor is spawned inside the cell membrane)

Suggested Future Modifications and Functionality Updates (cont):

- Randomize menu. Current menu layout closely represents the actual signaling process. Randomize the menu to make game play a bit more challenging.

- ATP tracking, receptor placement, and prevent objects from focusing on interactive objects outside of their reach. (i.e. an ATP targeting an activated receptor outside the cell membrane. This will have to be addressed on all objects as the current algorithm does not account for objects spawned outside of their reach.

- Address receptor angle relative to the cell membrane (the receptor should always spawn/attach perpendicular to the membrane)