**BioRubeBot Sandbox Simulator**

**Statement of Work**

For

Dr. Sarah Cline Ph. D. (Athens State University)

Rev 2.5

7/15/2015

Prepared by CS452 - Senior Software Engineering Project

Instructor: Katia Mayfield

Summer 2015

# Statement of Work

|  |  |
| --- | --- |
| Date | May 29, 2015 |
| Client | Athens State University |
| Job Name | BioRubeBot Sandbox Simulator |
| Requested by | Dr. Sarah Cline Ph.D. |
| From | Senior Software Engineering Project Team – Summer 2015 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revised** | **Description** | **Author** | **Latest Version** |
| 1/29/2015 |  | CS 452 Spring 2015 Senior Project Team | Initial Release – Version 1.0 |
| 5/29/2015 | Update to reflect goals for Summer 2015 | CS 452 Summer 2015 Senior Project Team | Rev 2.0 |
| 7/15/2015 | Final revision for Summer 2015. Added Acceptance Criteria | CS 452 Summer 2015 Senior Project Team | Rev 2.5 |

**Table of Contents**

Purpose 4

Scope 4

Applicable Documents 4

Requirements 5

General Description 5

User Stories 5

Product Functions 6

User Characteristics 6

General Constraints 6

Assumptions and Dependencies 6

Non-Functional Requirements 7

Performance 7

Reliability 7

Security 7

Maintainability 7

Tentative Iteration Plan 8

Appendix A 9

Appendix B 11

Appendix C 14

Appendix D – Acceptance Criteria 17

**Purpose**

This document is designed to specify the project plan needed for developing the BioRubeBot Sandbox Simulator for Dr. Sarah Cline. This is an assignment as the Senior Software Engineering Project for CS452 at Athens State University (ASU). This document's intended audience includes the developers of the BioRubeBot, as well as the stakeholders and other interested parties.

**Scope**

The BioRubeBot will be an easy-to-use cross-platform sandbox simulator use for protein interactions within biology. The system is based on a subset of biological rules based on these interactions. We hope to provide a tool for education as well as a possible method to perform experiments in a controlled, virtual environment.

**Applicable Documents**

|  |  |
| --- | --- |
| **Source** | **Location** |
| Bio Machine Sandbox Description | Appendix A |
| Desired Proteins List | Appendix B |
| Design Mock-ups for Cross-platform | Appendix C |

**Requirements**

**General Description**

The current model of education pertaining to intracellular interactions requires much lecture and example, but little interactivity. The goal of the BioRubeBot project is the development of an educational and experiment-based tool, to be used by teachers and student, designed to promote interaction and exploration of concepts. This application will allow its users to take their hypotheses and test them in a virtual, sandbox environment that can be controlled explicitly by it's user. With the inclusion of template cellular structures and template cells, the creation of a reactive, experimental environment can be achieved. The user will be able to create a fully developed cell, with included structures already present, a blank cell with no structures present, inactive cellular structures, and active cellular structures. These can all be accessed through a series of intuitive, pop-out menu boxes which will allow the user to create any of these objects or insert any pre-made templates. These objects are all designed to follow the natural rules, from cellular biology, of their real-world equivalents. From this point, the user can manipulate these objects into any position inside or outside any of the cellular structures they would be present in, thus generation their own experimental environment. Once the user is satisfied with the experimental environment, they may start the simulation. Once started, object within the simulation will begin to interact as per the rules of cellular biology. At any point in time, the simulation can be paused, manipulated, rewound, or fast-forwarded.

**User Stories**

* User wants a structure
* User wants a cell
* User wants a template cell
* User wants to insert an object into a cell
* User wants to delete an object from a cell
* User wants to start a simulation
* User wants to pause a simulation
* User wants to rewind a simulation
* User wants to fast-forward a simulation
* User wants to display a premade cellular interaction

**Product Functions**

* Main Menu
  + Allows user to start a simulation.
  + Allows user to pause a simulation
  + Allows user to quit a simulation
  + Allows user to rewind a simulation
  + Allows user to fast-forward a simulation
* Container Menu
  + Allows user to insert a template cell into the sandbox
  + Allows user to insert a blank cell into the sandbox
  + Allows user to insert a cellular structure into the sandbox
  + Allows user to insert a cellular object into the sandbox

**User Characteristics**

Any potential user should be able to successfully use this application. Basic knowledge of current technology preferred.

**General Constraints**

This application must be simple to use, yet be able to exhibit complex interactions within cells. To this end, the user interface must be intuitive, and the rules of the objects and their interactions must be well documented. With these two goals in mind, this application should be simplistic enough to learn 'on-the-fly', yet complex enough to display more intricate experimental interactions.

**Assumptions and Dependencies**

The software functions described herein are dependent upon only the power of the device trying to execute them. It is assumed that this device will be able to run cross -platform, on most devices. It is also assumed that this application will not require any access to any external data, support, tools, or the internet. This application is assumed to be self-sufficient and self-contained. In future implementations, updates may occur that can include: saving of simulations, sharing of simulations, updates from online sources, imports of user-made templates, game-based development, competition-based learning modules, leader-boards, and statistics/data tracking.

**Non-Functional Requirements**

**Performance**

The cellular biology rules for all represented objects must be exact in implementation.

**Reliability**

The same simulation must yield the same results consistently.

**Security**

There are no security concerns at this time.

**Maintainability**

Updates to the application must be made upon any change in the rules of the interaction of any represented cellular structures.

**Tentative Iteration Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Review Date** | **First Priority** | **Second Priority** | **Third Priority** |
| Feb 26, 2015 | 8 types of proteins with rules programmed | Images of proteins/molecules (Reference Pdb.org and google image search) |  |
| April 2, 2015 | Drag and drop menu (list) implemented | Start button | Proteins functioning in sandbox |
| April 23, 2015 | Phase 1 Completion | Environment/setting/obstacles: cell membrane and nuclear envelope (ideally an arc) | Two more proteins |
| May 18, 2015 | Initiate Phase 2: Familiarize new team members with project | Revise Statement of Work to reflect new requirements and goals | Begin Unity tutorial and coding review |
| June 11, 2015 | Establish functionality of remaining protein objects (limited functionality) | Evaluation of object functionality and code modification |  |
| July 9, 2015 | Improved functionality of game objects | Basic tutorial development |  |
| July 23, 2015 | Final presentation with improved game play and interactive tutorial implementation (limited to game progression) |  |  |

**Appendix A**

**The BioRubeBot Project:** We would like to create a **sandbox application** that can be used in anundergraduate to create two-dimensional movies of protein signaling models found in current literature. The app will be structured upon previously successful educational physics puzzle games, as evidenced by the continued success of The Incredible Machine (<http://www.youtube.com/watch? v=kl7hT2GiO5E>) and similar Rube-Goldburg style puzzle games, but it will be novel in that it will convert known examples of protein-protein interactions into game rules, much in the way educational physics games use the laws of physics to influence gameplay.

In the sandbox, users will be given a ‘blank slate’ wherein they will be able to arrange any of the proteins that they wish to work with in any order that they choose to place them. This will allow the teacher or student to generate his or her own, content specific movies, making it easier to use the game in teaching of the highly dynamic, swiftly changing field of Cell Biology.

|  |
| --- |
| **Table 1 - Example Rules** |
|  |
| *“A kinase adds a phosphate to a phosphorylation domain on a protein”* |
| *“You can add a phosphate to either an activating or inhibiting phosphorylation domain”* |
| *“A channel protein creates a tunnel through a*  *membrane”* |
| *“An ABC domain on a protein binds ATP”* |
| *“Proteins can have one or more domains”* |

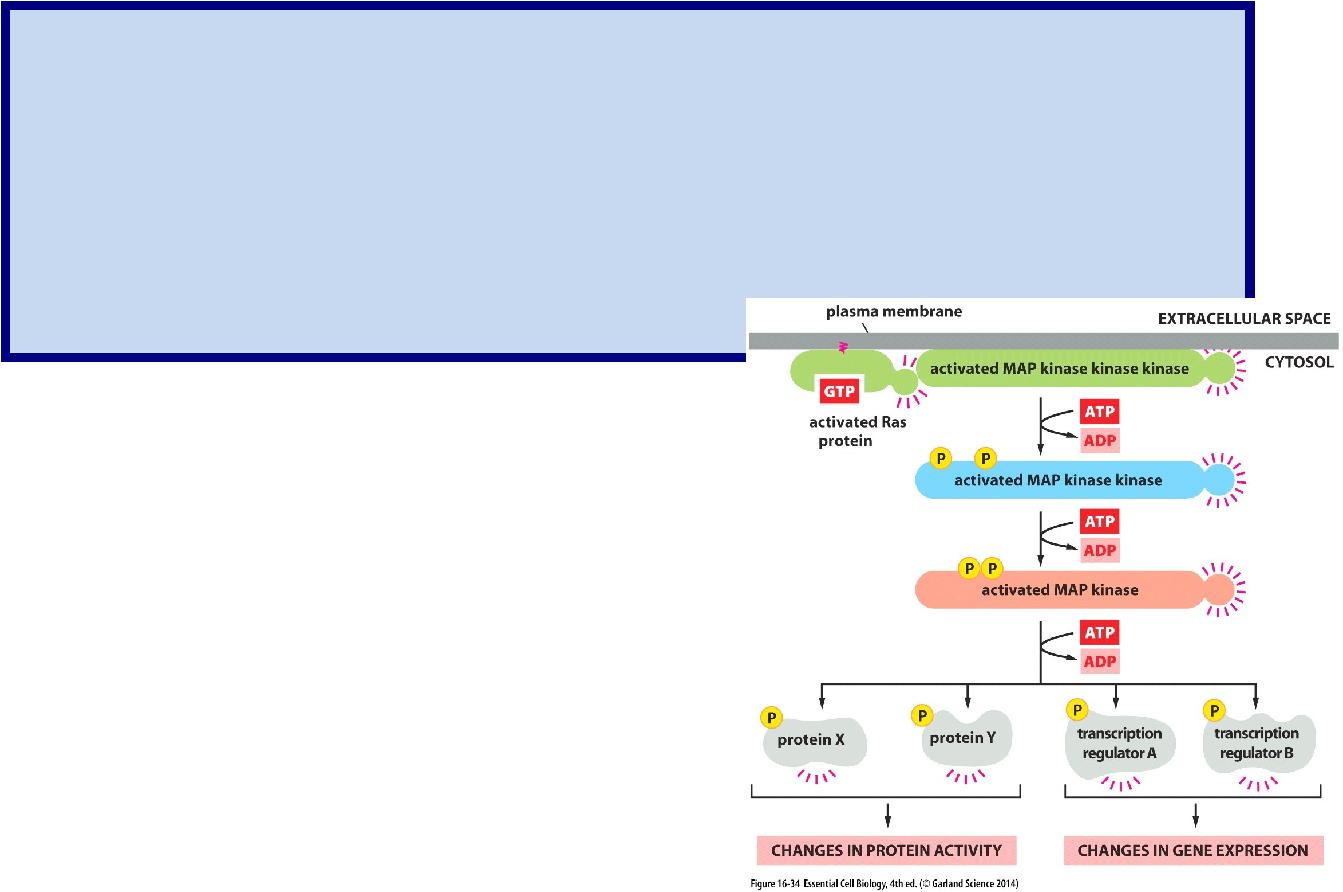
The target audience for the game itself includes public users of all genders and ethnicities, particularly focused on informal users aged 8-15. In consideration of this target audience, product will be but designed will an appropriate gender-neutral color scheme. In addition, since the sandbox level of the game will be promoted among college students and educators as a tool for demonstrating cellular processes, it is desirable that the game be relatively age neutral. Therefore, a graphics scheme will be employed that mimics demonstratively age neutral games, such as Tetris, Bejeweled and Angry Birds. To ensure that the project design is appropriate for the target audience, the gameplay itself will be modeled after an award-winning educational physics game, The Incredible Machine, which has already demonstrated success since. Amazon.com rates this game as appropriate for Early Childhood, but as a puzzle game, it can be designed to be appropriate for a broad range of age groups. This makes it easily adaptable as a content learning game for Advanced Prep Biology high school classrooms and college level General and Cell Biology.

The gameplay will be analogous to the set up seen in Figure 1, where the user is instructed to “Put Curie Cat in the cage”. In the example above, players are given intriguing, physics-based props with which to carry out this in-game objective. Props, which include pipes, treadmills, light switches, a mouse, and dynamite, interact using physics-based principles once the user has set up the objects and presses “Start”. In the protein interactions game created by this project, the props used to solve the puzzles will be proteins with specific functions assigned to them by various protein domains. Notably, there would be no end game objective in the sandbox.



**Figure 1**: Screen shot ofThe Incredible Machine 2.

The game-based rule set for these domains will be based upon simplifications of empirically demonstrated functions, such as those exemplified in Table 1. The rules will be used to complete puzzles based on known protein pathways. The user will be provided with the protein components of, for instance, a MAPK signal transduction cascade (Figure 2), and will be required to place them in the correct locations inside and outside of the cell in order to convey an extracellular signal from the outside of the cell all the way to the nucleus, using only the proteins that they are given.

**After six months, we would like to see the following objectives achieved:**

* *Allow users to arrange objects (proteins/small molecules) as desired.*
* *At least 10 different types of proteins/small molecules.*
* *Proteins/small molecules would have specific limitations, activations and functions. These would be provided by the content knowledge experts.*
* *An ability to “activate” the puzzle so proteins move and interact.*
* *A reward screen if all of the components on the screen interact.*

**Figure 2:** A generic MAPK signaling pathway from a popular textbook (Alberts et al., 2013)

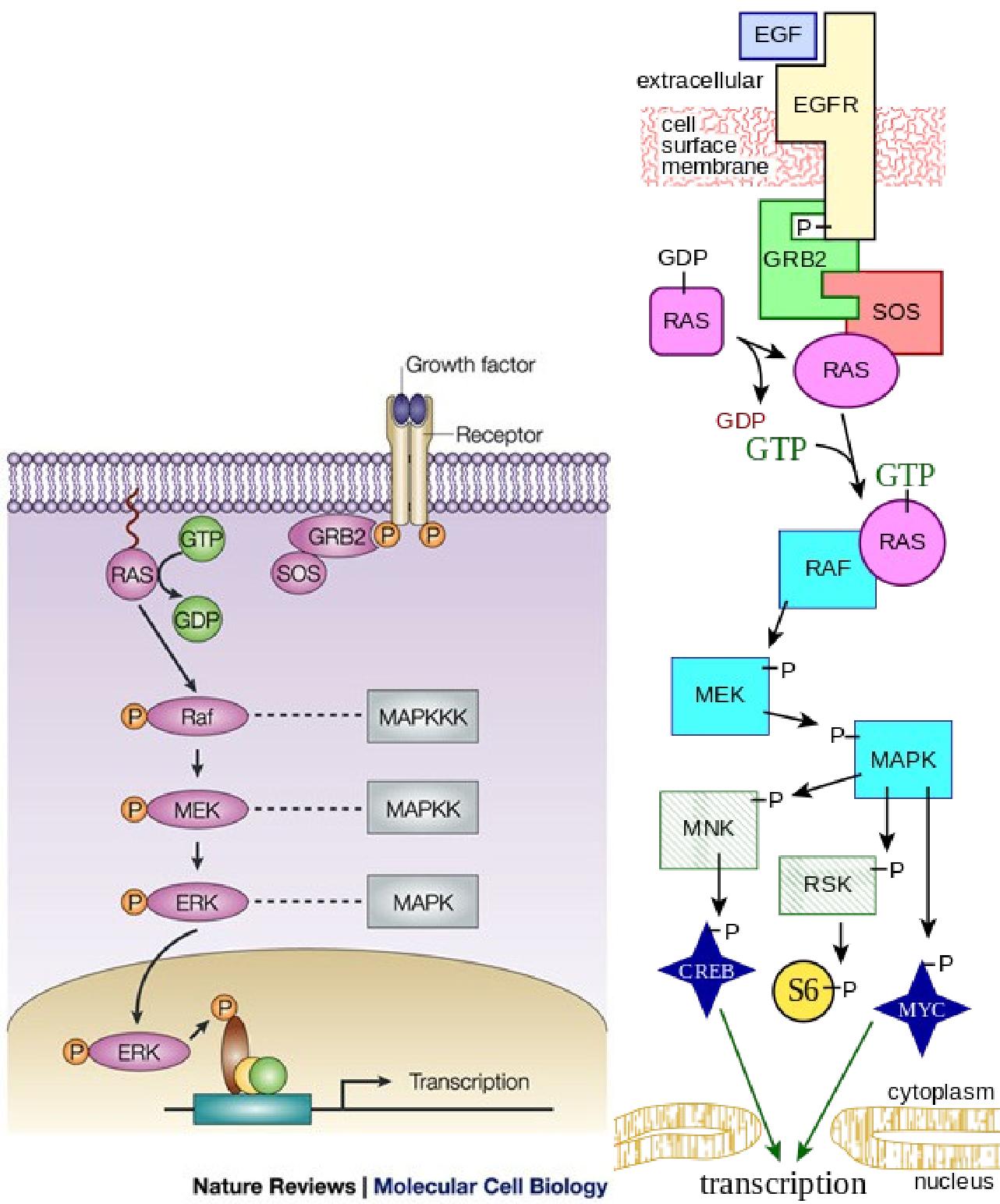
**Appendix B**

Flow chart: Extracellular signaling molecule -> receptor protein -> G-protein uses GTP/GDP/phosphate -> Kinase uses ATP/ADP/ phosphate -> transcription regulator binds to DNA and recruits RNA polymerase

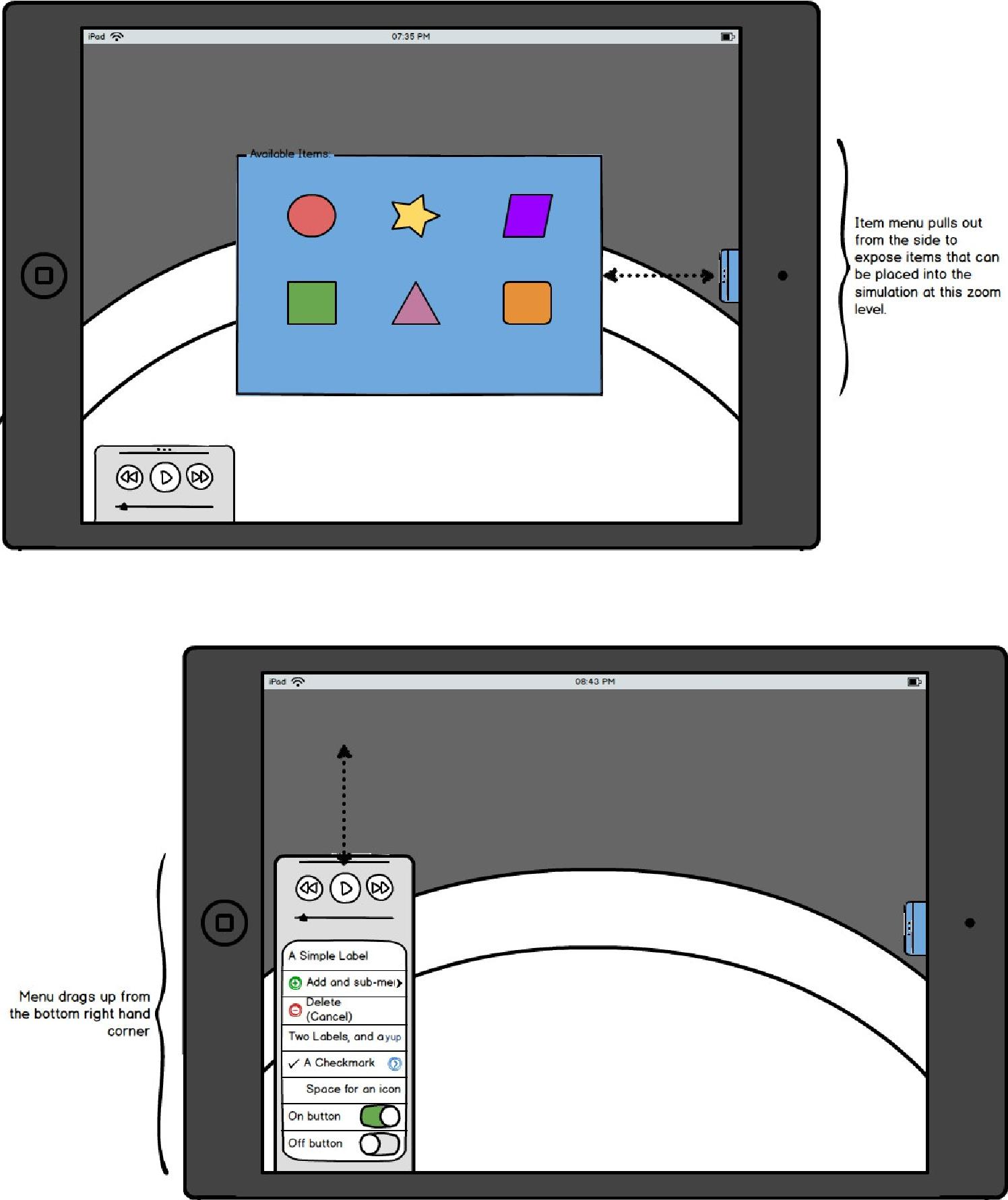
1. **Extracellular signaling molecule** (Technical name = EGF for epidermal growth factor):protein, free floating in liquid, does not change shape, binds to receptor protein
2. **Receptor protein** (Technical name = receptor tyrosine kinase, EGFR): protein, floats inmembrane, one side binds extracellular signaling molecule, the other side binds an intracellular protein. Exists as two proteins which then come together upon binding the signaling molecule. Upon dimer formation (the two proteins coming together), it will autoselfphosphorylate. That is, it will add phosphates to itself. Upon phosphorylation, it will bind to the intracellular protein. Technically Grb2 (see the diagram below), for the first stages of the program, it is OK for it to bind to the G-Protein.
3. **G-protein** (Technical name = RAS): the G-protein is so named because it binds GTP/GDP. Inthe OFF state, it is bound to GDP. Upon binding to a protein that activates it (SOS in the diagram below, but it is OK if it is the phosphorylated receptor protein for the first rendition), it will swap out a GDP for a GTP. It is then active and will activate a kinase.
4. **GTP/GDP/ phosphate**: GTP is a nucleotide, not a protein. It has 3 phosphates on it. GDP is thesame nucleotide with two phosphates instead of three. When GTP is converted (hydrolyzed) into GDP, the nucleotide stays the same, just the phosphate is lost. If a protein is bound to GDP, it cannot add the phosphate back. Instead it has to “spit out” the GDP and grab hold of a different GTP molecule. If you want to think about it as GTP being the pistachio nut, and you eat the nut (phosphate) and throw away the shell (GDP), this is a decent analogy.
5. **Kinase:** Kinases add phosphates to things. The phosphates come from ATP. The situation is likethe one above with GTP. A kinase takes a phosphate from ATP and then adds that phosphate to a protein (in this case, the transcription regulator). You are left with ADP, the “waste”. (In the diagram below, there are three kinases in a row.)
6. **ATP/ADP/ phosphate:** ATP, like GTP, is a nucleotide. It acts basically the same as GTP. Thedifference is the way that they interact with proteins.
7. **transcription regulator:** After being phosphorylated by the kinase, the transcription regulatorwill affect transcription, or the production of an mRNA transcript from DNA. In our case, it will bind to the DNA and increase transcription. It will increase transcription by recruiting, or allowing to bind, an RNA polymerase.
8. **DNA:** The classic double helix. This guy just sits in the nucleus until the transcription regulatorand the RNA polymerase bind to it.
9. **RNA polymerase:** the RNA polymerase reads the DNA and makes a copy of the DNA out ofRNA. It will bind to the DNA after the transcription regulator does.

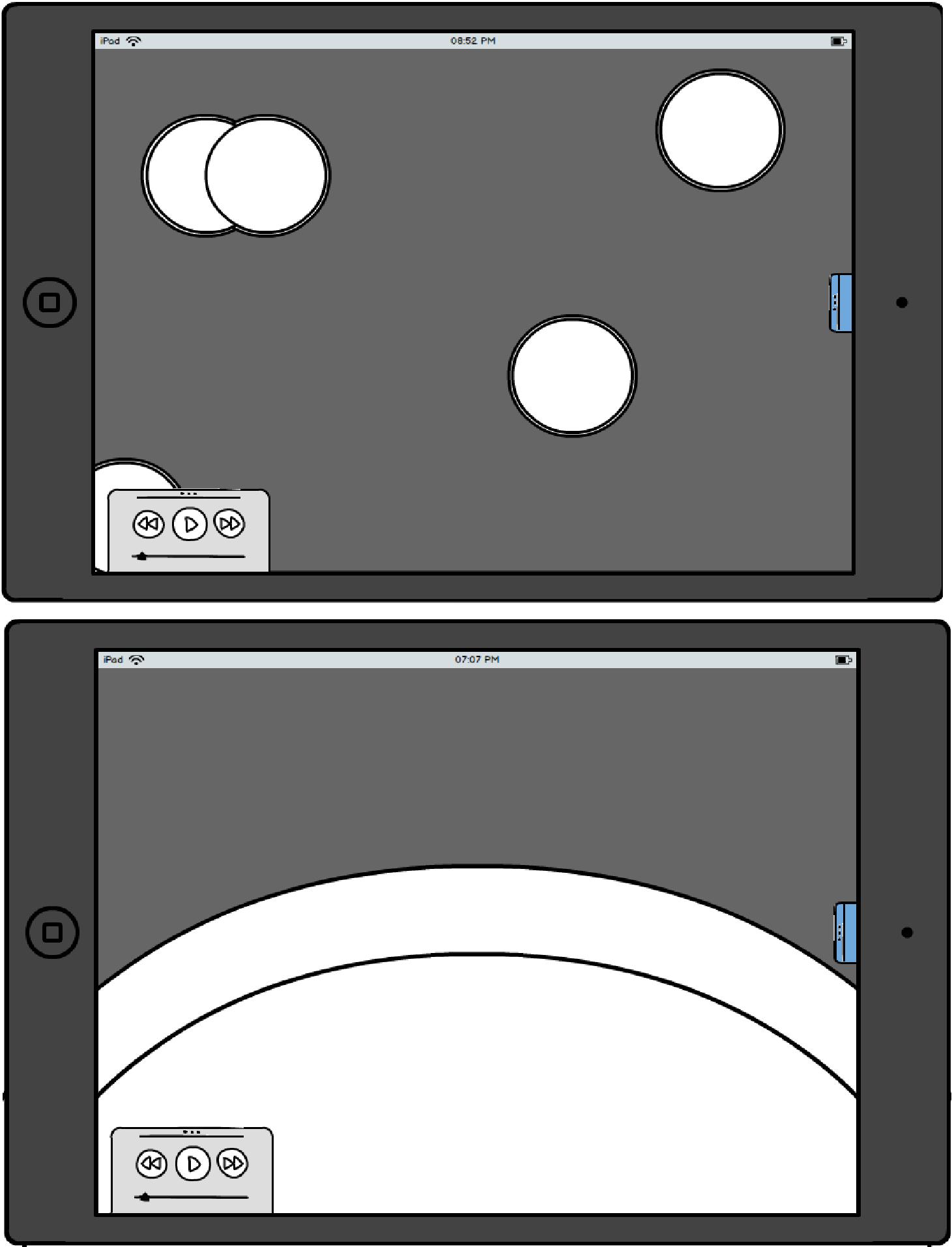
**Reference Videos:**

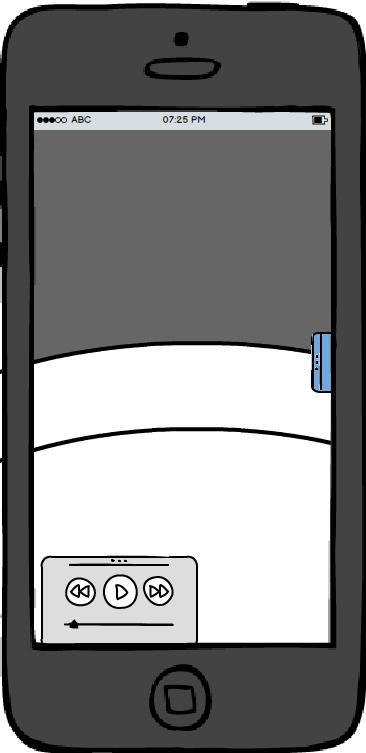
https://www.youtube.com/watch?v=pSw5NLao7IQ (up to 1:10) https://www.youtube.com/watch?v=oDjDUUhGVsI https://www.youtube.com/watch?v=JTY6\_LRtUEE https://www.youtube.com/watch?v=npnLnzsWYFg https://www.youtube.com/watch?v=Ia6OjvBazGE https://www.youtube.com/watch?v=kiHAdan2AOY https://www.youtube.com/watch?v=\_Zq5qrVbN2A



**Appendix C**







Appendix D – Acceptance Criteria

1. Control Functions
   1. Does the Start button start the simulation?
   2. Does the menu button display the menu options correctly?
   3. Does the Play button cause the objects to begin to move?
   4. Does the pause button cause the simulation to temporarily stop?
   5. Does the Fast Forward button increase the simulation’s speed?
2. Object Spawn and Movement
   1. Does the signaler spawn/move correctly?
      1. Signaler should only spawn outside the cell.
   2. Does the Receptor spawn correctly?
      1. Receptor should snap to a position perpendicular to the cell wall.
   3. Does the ATP spawn/move correctly?
   4. Does the G-Protein spawn/move correctly?
      1. The G-Protein should spawn with a GDP attached.
   5. Does the GTP spawn/move correctly?
   6. Does the Kinase spawn/move correctly?
   7. Does the Transcription Regulator spawn/move correctly?
   8. Does the Cell structure spawn correctly?
      1. This includes both the membrane and the nucleus.
3. Object Interactions
   1. Signaler
      1. Does it link properly with the receptor?
   2. Receptor
      1. Does it receive the signaler?
      2. Does it follow its transform protocol to prepare to receive the phosphate from the ATP?
      3. Does it receive the phosphate from the ATP?
   3. ATP
      1. Does it drop a phosphate on the receptor’s leg?
   4. G-Protein
      1. Does it link to the receptor’s phosphate properly?
      2. Does it drop its GDP after it picks the phosphate up? (GDP is fades out at this point.)
      3. Does it pick up the GTP after it drops the GDP?
      4. Does it target the Kinase then it’s in this “fully activated” state?
   5. GDP
      1. Does it release properly from the G-Protein?
   6. GTP (combination of GDP and Phosphate)
      1. Does it link with the G-Protein correctly?
   7. Kinase
      1. Does it link with the activated G-Protein correctly?
   8. Transcription Regulator (T-Reg)
      1. To Be Determined