

KinectAGraph

Software Requirements Specification

0.1

April 21, 2016

Callum Brill
Lauren Donner
Lily Evans
Will Jamieson

Prepared for
Sebastian van Delden
CSCI 462 Software Engineering Practicum

Revision History

Date	Description	Author	Comments
4/14/16	<Version 1>	Will Jamieson	Finished document

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Will Jamieson	Lead Software Eng.	4/14/16

Table of Contents

Revision History.....	ii
Document Approval.....	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Definitions, Acronyms, and Abbreviations.....	1
1.4 References.....	1

1.5 Overview.....	1
2. General Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions.....	2
2.3 User Characteristics.....	2
2.4 General Constraints.....	2
2.5 Assumptions and Dependencies.....	2
3. Specific Requirements.....	2
3.1 External Interface Requirements.....	3
3.1.1 User Interfaces...	3
3.1.2 Hardware Interfaces...	3
3.1.3 Software Interfaces...	3
3.1.4 Communications Interfaces...	3
3.2 Functional Requirements.....	3
3.2.1 <Functional Requirement or Feature #1>...	3
3.2.2 <Functional Requirement or Feature #2>...	3
3.3 Use Cases.....	3
3.3.1 Use Case #1...	3
3.3.2 Use Case #2...	3
3.4 Classes / Objects.....	3
3.4.1 <Class / Object #1>...	3
3.4.2 <Class / Object #2>...	3
3.5 Non-Functional Requirements.....	4
3.5.1 Performance...	4
3.5.2 Reliability...	4
3.5.3 Availability...	4
3.5.4 Security...	4
3.5.5 Maintainability...	4
3.5.6 Portability...	4
3.6 Inverse Requirements.....	4
3.7 Design Constraints.....	4
3.8 Logical Database Requirements.....	4
3.9 Other Requirements.....	4
4. Analysis Models.....	4
4.1 Sequence Diagrams.....	5
4.3 Data Flow Diagrams (DFD).....	5
4.2 State-Transition Diagrams (STD).....	5
5. Change Management Process.....	5
A. Appendices.....	5
A.1 Appendix 1.....	5
A.2 Appendix 2.....	5

1. Introduction

1.1 Purpose

Through the use of an augmented reality controller such as the Xbox Kinect, a user will be able to choose a type of graph (polynomial, conic section, etc.) and be able to manipulate that graph with their bodies. The equation for the graph object displayed would be displayed to the user.

1.2 Scope

Our project, KinectAGraph, uses Microsoft's Kinect V2 to capture human movements and make adjustments to a graph allowing the end user to become more familiar with basic geometry. This software will allow the user to stand in front of an Xbox Kinect connected to a local server via an open socket. When the Kinect and laptop are powered on the the end user is then able to select a graph type from the menu. Once the graph is selected the user is able to interact with the graph by adjusting it an updated graph formula will appear on the laptop's display. The end user will be able to manipulate the graph as much as possible with their hands along the X and Y coordinates and see the updated graph formula adjust in real time. The reason for developing such software is to allow our client, Hawkes Learning Systems, to use this at conferences and expos. Their reasoning for wanting this to be developed is to educated students in elementary and middle school as well as excite them about the possibilities of what can be done with STEM. If the end user of this application has an enjoyable experience while using the app and is eager to learn more about technology our finished product will have reached its goal.

1.3 Definitions, Acronyms, and Abbreviations

1. STEM: Acronym for Science, Technology, Engineering, and Mathematics. This is used when describing an education in one of those specific disciplines
2. SRS: Acronym for Software requirements specification, which is this document highlighting all specifics around the software we build for Hawkes Learning
3. Microsoft Kinect: Is a motion sensing input device developed by Microsoft for Xbox 360, Xbox One, and Windows PC. Our team used it specifically for a Windows PC implementation

1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5 Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter. The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. General Description

2.1 Product Perspective

The project KinectAGraph is unique to the market in since there is no other software out there that performs our functionality. There are many only Windows PC apps out there that leverage the Kinect's motion and depth recognition functionality as well as other application that use the Kinect for education. What these apps fail to do is provide an avenue for education through graph manipulation. There are few independent developer applications for the Kinect mostly do to the higher barrier to entry a Kinect and an adapter. This high barrier to entry helps eliminate competition in the space and lack of diversity in the app market.

2.2 Product Functions

1. The user will choose a graph type (perhaps polynomial or conic section) from a main screen using movement.
2. Once a graph type is chosen, a default graph type is placed onto a Cartesian coordinate grid along with its equation.
3. The user can manipulate this graph through the use of the handles provided in the default graph type or by some other means.
4. The equation of the graph (as manipulated) will be displayed to the user, along with other fundamental parameters (such as vertex for a parabola, or center for a circle, etc.).

2.3 User Characteristics

The user of our end product will be as young as eight years old and have at least a little understanding of technology and how to interact with a camera and computer. A older and professional user group will also be using this application, just in a different environment. The younger will be using it to learn about basic graph types and gain an understanding of the power of technology. Whereas the older user group will most likely see this product demoed off at fairs to give them a visual understanding of the type of products Hawkes is developing while recruiting prospective hires.

2.4 General Constraints

The application will not allow more than one user to use it at one time, since the Kinect's sensors are looking exclusively for two hands. Having more than one user in the field of view will cause the application to function improperly. The application will not respond to vocal commands only physical hand motions.

2.5 Assumptions and Dependencies

This software assumes that the end user has a both a windows machine, an Xbox Kinect V2, and an adapter for the Xbox Kinect V2. Without these dependencies the software will fail to function due to the applications need to access specific elements of the hardware.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

This application is built using the Kinect SDK as well as javascript and HTML 5. The design is based off the core design guidelines for HTML 5 web applications.

3.1.2 Hardware Interfaces

The hardware interface is the Xbox Kinect and its adapter, no additional hardware interfaces or modifications to the original hardware are included in our application.

3.1.3 Software Interfaces

This application is built using the Kinect SDK as well as javascript and HTML 5. The design is based off the core design guidelines for HTML 5 web applications.

3.1.4 Communications Interfaces

There is no communications interface involved in this system.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 1. The user (as young as 8 years old) will choose a graph type (perhaps polynomial or conic section) from a main screen using movement.

3.2.1.1 Introduction: This requirement on the opening screen of the app and dictates all future screens.

3.2.1.2 Inputs: The input is the users hands selecting the specific graph type from the menu

3.2.1.3 Processing: The application then processes the selection via gesture recognition and handles it accordingly.

3.2.1.4 Outputs: The application then changes screens with respect to the menu option selected. It will display the appropriate graph type in the web browser.

3.2.2 Once a graph type is chosen, a default graph type is placed onto a Cartesian coordinate grid along with its equation.

3.2.1.1 Introduction: This requirement allows the user to begin interacting with a graph and utilize the functionality of the Kinect

3.2.1.2 Inputs: There is no input required to make this requirement occur. The input occurs on the previous screen and the graph appears in response to the previous input.

3.2.1.3 Processing: The selected graph is drawn on a canvas via HTML 5

3.2.1.4 Outputs: The graph chosen on the menu screen.

3.2.3 The user can manipulate this graph through the use of the handles provided in the default graph type or by some other means.

3.2.1.1 Introduction: This allows the users to interact with the graph and receive user feedback

3.2.1.2 Inputs: The hand gestures and movements allow interaction with the graph. The user can narrow the graph or adjust its alignment on the X and Y axis.

3.2.1.3 Processing: The inputs are processed via javascript that recognize the different gestures a user can make. It will identify either two fingers, an open hand, or a closed hand.

3.2.1.4 Outputs: The graph will be redrawn in accordance to the movements and gestures recognized by the Kinect.

3.2.4 The equation of the graph (as manipulated) will be displayed to the user, along with other fundamental parameters

3.2.1.1 Introduction: This requirement gives valuable user feedback to end user allowing them to learn from the graph manipulation

3.2.1.2 Inputs: There are no inputs for this requirement

3.2.1.3 Processing: The application calculates the location graph on

3.2.1.4 Outputs: The output is the updated graph formula

3.3 Use Cases

3.3.1 Use Case #1

Use Case: Manipulate parabola

Primary Actor: Member (*Registered User*)

Scope: Xbox Kinect Application

Brief:

The user selects the parabola option from the menu and is then taken to a screen that shows a parabola on a canvas. From there the user can interact with it by moving their hands in front of the Kinect.

Postconditions

Success Guarantees:

- The member moves the parabola with hand motions
- The parabolas formula is successfully updated

Preconditions:

The laptop and Kinect are both powered on.

Triggers:

The member selects the parabola menu button to open the correct graph.

Basic flow:

1. The software provides a menu screen upon opening the application with options to select and info, parabola, hyperbola, or ellipse button.
2. The member then selects parabola.
3. A new screen opens with a parabola appearing on a canvas
4. The member then interacts with the parabola by closing their hands and moving them in certain directions.
5. The graph then changes and the formula for the graph updates on the application.

3.3.2 Use Case #2

Use Case: Manipulate ellipse

Primary Actor: Member (*Registered User*)

Scope: Xbox Kinect Application

Brief:

The user selects the ellipse option from the menu and is then taken to a screen that shows a ellipse on a canvas. From there the user can interact with it by moving their hands in front of the Kinect.

Postconditions

Success Guarantees:

- The member moves the ellipse with hand motions
- The ellipse formula is successfully updated

Preconditions:

The laptop and Kinect are both powered on.

Triggers:

The member selects the ellipse menu button to open the correct graph.

Basic flow:

1. The software provides a menu screen upon opening the application with options to select and info, parabola, hyperbola, or ellipse button.
2. The member then selects ellipse.
3. A new screen opens with a ellipse appearing on a canvas
4. The member then interacts with the ellipse by closing their hands and moving them in certain directions.
5. The graph then changes and the formula for the graph updates on the application.

3.3.3 Use Case #3

Use Case: Manipulate hyperbola

Primary Actor: Member (*Registered User*)

Scope: Xbox Kinect Application

Brief:

The user selects the hyperbola option from the menu and is then taken to a screen that shows a hyperbola on a canvas. From there the user can interact with it by moving their hands in front of the Kinect.

Postconditions

Success Guarantees:

- The member moves the hyperbola with hand motions
- The ellipse formula is successfully updated

Preconditions:

The laptop and Kinect are both powered on.

Triggers:

The member selects the hyperbola menu button to open the correct graph.

Basic flow:

1. The software provides a menu screen upon opening the application with options to select and info, parabola, hyperbola, or ellipse button.
2. The member then selects hyperbola.
3. A new screen opens with a hyperbola appearing on a canvas
4. The member then interacts with the hyperbola by closing their hands and moving them in certain directions.
5. The graph then changes and the formula for the graph updates on the application.

3.3.4 Use Case #4

Use Case: Open menu screen

Primary Actor: Member (*Registered User*)

Scope: Xbox Kinect Application

Brief:

The user selects the menu option from the menu and is then taken to a screen that shows a hyperbola on a canvas. From there the user can interact with it by moving their hands in front of the Kinect.

Postconditions

Success Guarantees:

- The member moves the hyperbola with hand motions
- The ellipse formula is successfully updated

Preconditions:

The laptop and Kinect are both powered on.

Triggers:

The member selects the hyperbola menu button to open the correct graph.

Basic flow:

1. The software provides a menu screen upon opening the application with options to select and info, parabola, hyperbola, or ellipse button.
2. The member then selects hyperbola.
3. A new screen opens with a hyperbola appearing on a canvas
4. The member then interacts with the hyperbola by closing their hands and moving them in certain directions.
5. The graph then changes and the formula for the graph updates on the application.

3.4 Classes / Objects

To be completed when code is finished and can make accurate comparisons to the code.

3.4.1 Parabola

3.4.1.1 Attributes

- The Parabola is drawn on an HTML 5 canvas and changes colors based on its current state.
- It is interacted with via the hand motions from the Xbox Kinect.
- Displays a formula for the parabola on screen.

3.4.1.2 Functions

- Allows the users to either expand or collapse the graph with their hands
- Allows the users to translate the graph across the X or Y axis by utilizing the lasso hand gesture

3.4.2 Line

3.4.2.1 Attributes

- The line is drawn on an HTML 5 canvas and changes colors based on its current state.
- It is interacted with via the hand motions from the Xbox Kinect.
- Displays a formula for the line on screen.

3.4.2.2 Functions

- Allows the users to adjust the slope of the line with their hands
- When the slope changes the formula updates on the screen in real time

3.5 Non-Functional Requirements

1. HTML 5 Web Application Preferred (see [Kinect for Windows SDK Javascript API](#))
2. Bandwidth limitations of real-time image streams
3. Instructional Documentation
4. Usability
5. At least 4/5 students who have not yet learned to love math think this application is fun to use

3.5.1 Performance

The application will compile and execute without constraint on the host machine's CPU allowing the program to run smoothly with minimal lag.

3.5.2 Reliability

The application will work properly as long as there is sufficient power in on the host machine and Xbox Kinect V2

3.5.3 Availability

The application will be available as long as there is sufficient power in on the host machine and Xbox Kinect V2

3.5.4 Security

Security is not a concern for this application because it runs only on a local host and has not open connections to third party application

3.5.5 Maintainability

The system will not be maintained by the current team going forward Hawkes will do the projects maintenance if they wish to continue with the project.

3.5.6 Portability

The system is portable to any windows machine as long as there is a power source to power the Xbox Kinect. The power enables the Kinect to capture user's interaction with the screen.

3.6 Inverse Requirements

This application does not have any inverse requirements.

3.7 Design Constraints

This application is built using the Kinect SDK as well as javascript and HTML 5. The design is based off the core design guidelines for HTML 5 web applications.

3.8 Logical Database Requirements

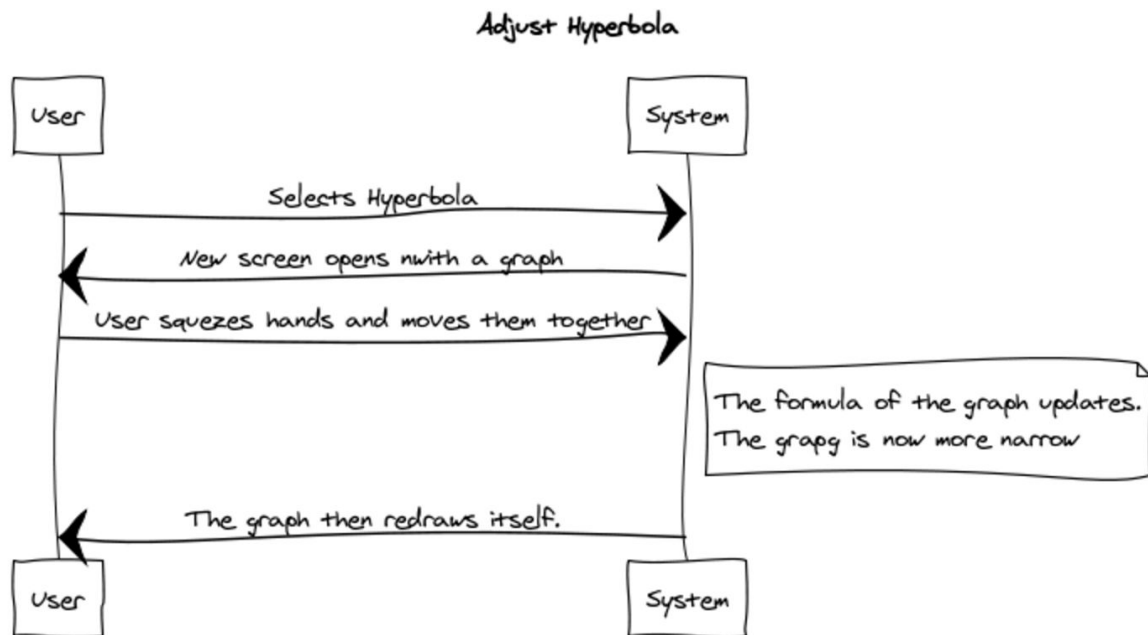
This application does not maintain a database all system specs are restored to default each time the application is executed.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

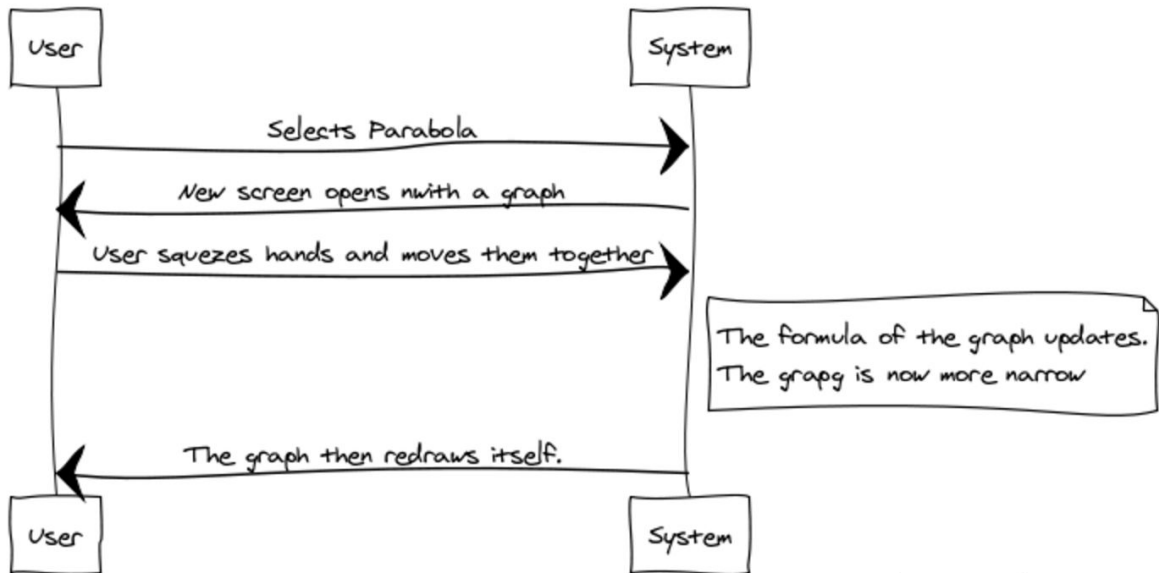
4.1 Sequence Diagrams

4.1.1



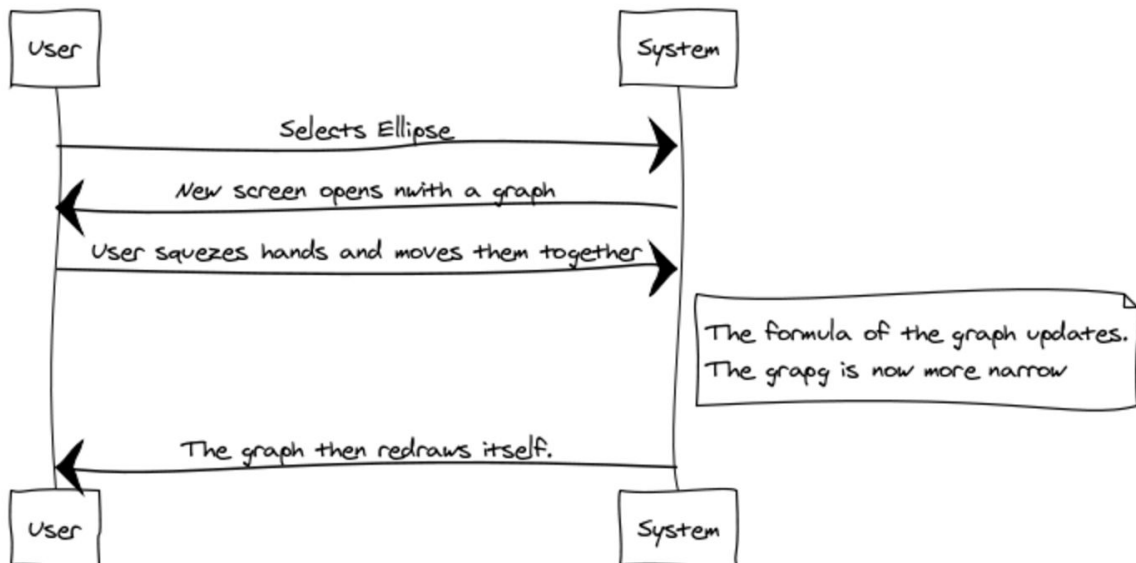
4.1.2

Adjust Parabola

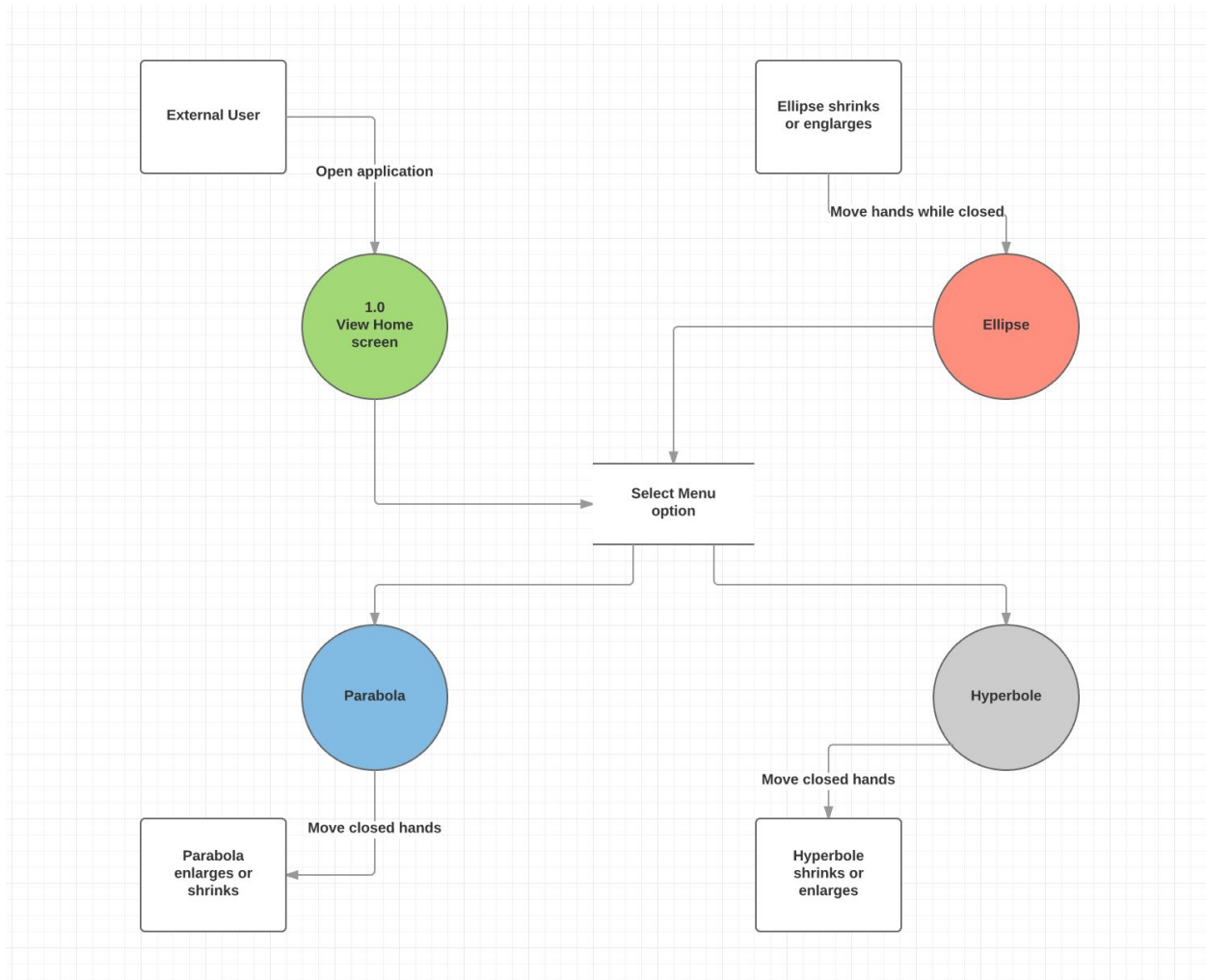


4.1.3

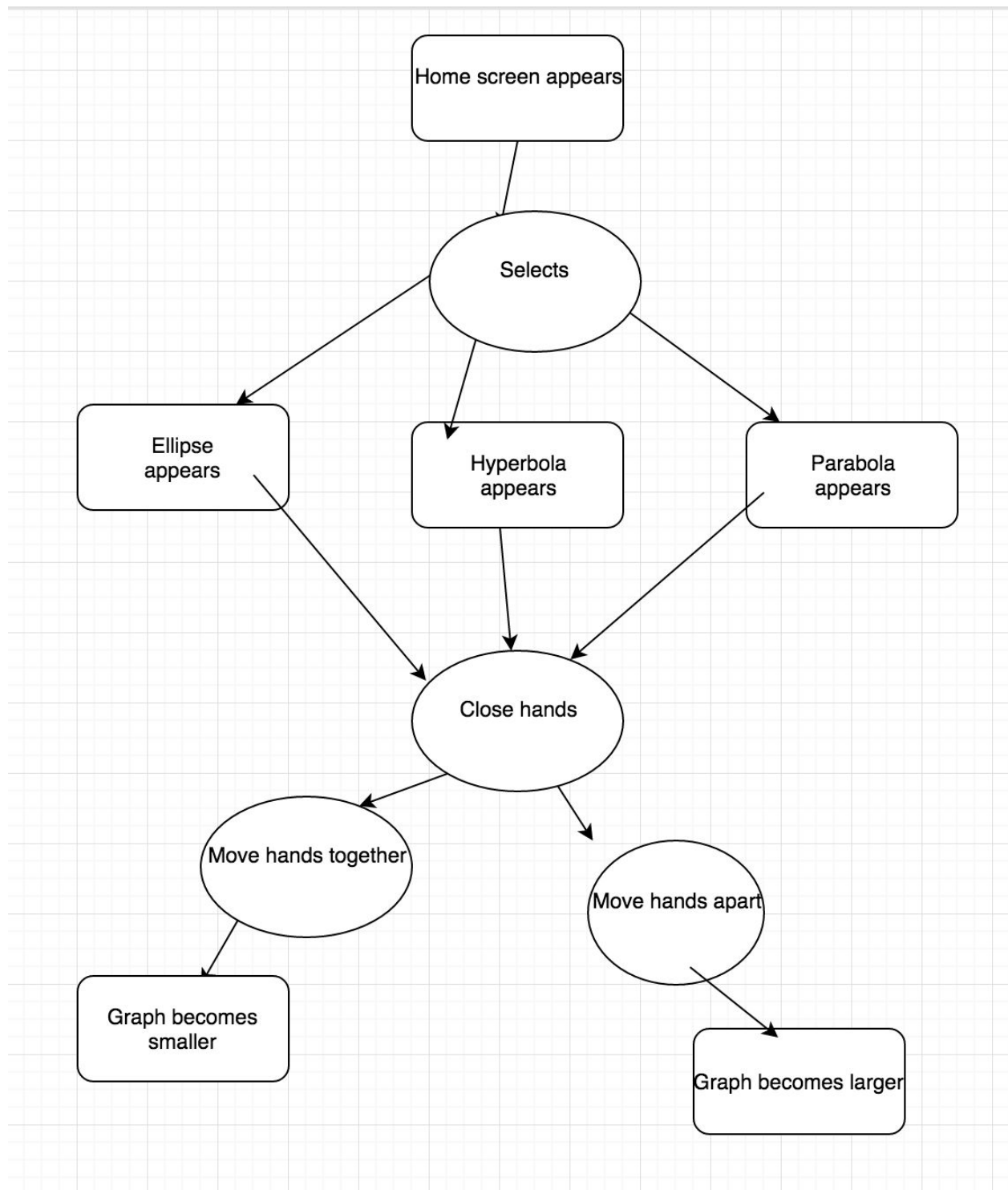
Adjust Ellipse



4.3 Data Flow Diagrams (DFD)



4.2 State-Transition Diagrams (STD)



5. Change Management Process

Any team member, Will, Callum, Lauren, or Lilly can submit changes to this document at any time. If a change is submitted, it was because project instructions were changed by either Marcel or Alex collectively known as the ("**Project Owner's**"). The project owners will also have the ability to make direct changes to this document if they deem fit.

6. SOFTWARE IMPACT

6.1 Describe the professional, ethical, legal, security and social issues related to this project.

This project serves no legal, ethical, security, or social issues. However it does relate to a professional issue that Hawkes Learning Systems may be dealing with. Tech recruiting is becoming more and more difficult with companies in the California offering luxurious salaries as well as many hard to turn down office perks. This project allows Hawkes to demo and intriguing new software product that allows them demo exciting new technology they are creating at Hawkes. This project serves more as a recruiting tool than anything else to Hawkes Learning Systems. This project allows Hawkes to demo and intriguing new software product and recruit top tech talent to their company. This product serves more as as recruiting tool than anything else.

6.2 Analysis of the local and global impact of this project on individuals, organizations, and society

This project has little global impact on individuals, organizations, and society. However this project does impact local organizations and individuals by educating them on the emerging fields of math and science. It will provide as a useful tool to get individuals excited about STEM related fields as well as create new jobs.

6.3 What types of continuing professional development training/opportunities/etc are needed to maintain and evolve this project.

There are no continued professional developments required to maintain this project. It was developed with the intention that it would be used as a prototype and prototype only used for demos. The training needed to for this product is one walkthrough demo.

The product was created with an intuitive UI to allow the end user to understand the product workflow with little if any explanation at all.