Algebra Tutor

System Manual

Mujeeb Adelekan

madelekan@mail.stmarytx.edu

StMU

CS6345 eLearning and Gamification

12/07/2021

Table of Contents

**Overview** ……….......................................................................................................................... 3

**Requirements**............................................................................................................................. ..6

Hardware and Software Requirements…………………............................................................ ..6

Player Requirements……………………................................................................................... ...8

Functional Requirements……………………............................................................................ ...8

Language and Operating System……………............................................................................. ..9

**Design & Storyboard** …………….............................................................................................10

Flowchart......................................................................................................................................10

Updated Storyboard.….................................................................................................................11

**Installation**……………………….............................................................................................. 34

User Setup ................................................................................................................................. ..34

Developer Setup……................................................................................................................. ..35

**Sample Session** ....................................................................................................................... …39

**Modifications** ……...,...................................................................................................................52

How to Modify Algebra Tutor’s Code ......................................................................................... 52

Changes to Consider .....................................................................................................................58

**Troubleshooting** ……...,.. ...........................................................................................................59

**References**……..……...,.. .........................................................................................................61

**Appendix**……… ……...,.. .........................................................................................................62

Overview

The following is an overview of the background and nature of Algebra Tutor, as well as an outline of this manual.

Algebra marks a turning point in math education. It is the gatekeeper between elementary math concepts and higher-level math. Yet, despite the myriad of math tutors throughout the years, there doesn’t seem to be enough electronic games that focus on algebra.

There are many resources that assist students in algebra, including Khan Academy and Brilliant. However, Khan Academy is extremely formal and academic, giving off a dry atmosphere. Brilliant offers more involved instruction but is pricey compared to alternative means of self-instruction, like buying a textbook.

There are many math YouTube Channels that cover algebra, but, like Khan Academy, is very cut and dry in content. A YouTube Channel that teaches math through visual means in 3Blue1Brown, but his content focuses on higher-level math.

When it comes to gaming, elementary and middle school math gets a lot of attention, but algebra seems to be a missing piece outside of a few exceptions (i.e., Math Blaster Algebra). My tutor, Algebra Tutor, is designed to help fill the gap.

In designing Algebra Tutor, I was inspired by Brilliant’s instruction of algebra and ability to make the player interact with the concepts in various ways before applying the concepts with formal math notation. I intend to apply Gagne’s Nine Events of Instruction to solidify the topics covered by the tutor.

Algebra Tutor is a single user application, intended for middle or high school students struggling with algebra concepts. The topics are algebra concepts covered in an 8th or 9th grade class following Texas education standards. The prerequisites for using Algebra Tutor includes having a basic English reading ability, familiarity with basic arithmetic (addition, subtraction, multiplication, and division), and the user must be familiar with math concepts up to the 7th grade (negative numbers, fractional representation, coordinate planes, etc.)

Algebra Tutor was developed in Unity using scripts generated in C#. The tutor is designed for Windows 7, 10 and 11 Machines. The game will be developed on an HP Pavilion Laptop 15z-cw100 running the Windows 10 operating system. A mouse and keyboard will be required to play the tutor.

The Algebra Tutor will consist of five lessons, each focusing on a topic with each topic building off the last. The topics are: 1) Introduction to Variables, 2) Fractions, decimals, and percentages review, 3) Constructing equations, 4) Solving Equations, and 5) Graphing lines.

Each lesson will have a Presentation, a Tutorial, and 3 questions regarding the topic. The user gets infinite tries on each question. Once a lesson has been completed, the user will see the amount of attempts it took for them to answer the question correctly. After finishing all five lessons, the user is given an overview of their performance and what they learned.

This manual will outline the hardware, software, and user requirements of Algebra Tutor. In addition, the manual will discuss the tools needed by the developer as well as the functional requirements of the tutor. The tools used by the author to develop Algebra Tutor will be highlighted with justification on why such tools were used.

Afterwards, the flowchart and a storyboard of Algebra Tutor will be displayed. This will be followed by instructions on how to install Algebra Tutor. A sample session of the user going through all five lessons will follow.

After the sample session, the manual will discuss how to make changes to the code of Algebra Tutor, and discuss how to troubleshoot common problems or errors. This manual will end with a references page followed by an appendix containing the tutor’s source code.

Requirements

The following is an outline of the hardware, software, user, and functional requirements of Algebra Tutor.

**Hardware and Software requirements for Algebra Tutor:**

Hardware:

* Windows machine
  + Intel/AMD/Nvidia Processor
  + Minimum Disk Space: 246 MB
* Mouse, Keyboard, and Monitor

Software:

* Operating System: Windows 7+
* Code Editor: Notepad or Visual Studio (2015 or newer)

**Notes for Unity Player and Editor:**

Algebra Tutor is intended to run on an x86-based, 64-bit Windows machine. The following requirements are provided by Unity Technologies; see the links provided in the References section for more information.

How to run the Unity Player on a Windows desktop:

|  |  |
| --- | --- |
| **Windows version** | Windows 7 (SP1+), Windows 10 and Windows 11 |
| **CPU** | x86, x64 architecture with SSE2 instruction set support. |
| **Graphics API** | DX10, DX11, DX12 capable. |
| **Additional requirements** | Hardware vendor officially supported drivers. |

For all operating systems, the Unity Player is supported on workstations, laptop, or tablet form factors, running without emulation, container, or compatibility layer.

How to run the Unity Editor on a Windows desktop:

|  |  |
| --- | --- |
| **Windows version** | Windows 7 (SP1+), Windows 10 and Windows 11, 64-bit versions only. |
| **CPU** | x64 architecture with SSE2 instruction set support |
| **Graphics API** | DX10, DX11, and DX12-capable GPUs |
| **Additional requirements** | Hardware vendor officially supported drivers |

For all operating systems, the Unity Player is supported on workstations, laptop, or tablet form factors, running without emulation, container, or compatibility layer.

**Player Requirements for Algebra Tutor:**

* Basic English reading ability
* Familiarity with basic arithmetic
* Familiarity with math concepts up to the seventh-grade level, including but not limited to:
  + Negative numbers
  + Coordinate planes
  + Parts of a whole

**Functional Requirements for Algebra Tutor:**

* Display a presentation for each of the five lessons
* Display a tutorial for each of the five lessons
* Display 3 questions for each of the five lessons
* Allow the user to type in an answer
  + Give immediate feedback on whether the answer is correct or incorrect
  + Give the user infinite attempts to answer the question
* Allow the user to move to the previous screen when desired
* Allow the user to get a unique hint upon clicking a “Hint” button
* Display the amount of attempts the user took to enter the correct answer at the end of a lesson
* Allow the user to move on to the next lesson after viewing the number of attempts for each question in the current lesson.
* Display an outline of everything the user has learned after completing the fifth and final lesson.
* Allow the user to quit the application while on the title screen
* Allow the user to hit “Play” to start the first lesson while on the title screen

**Language and Operating System**

Language: C#

Due to the game’s development within the Unity 2D Editor, the game is written in the high-level, object-oriented programming language C#. C# is a versatile language that allows for game scripting in Unity. This makes it easy to modify UI elements such as button logic, question generation, and question/hint diagram display logic.

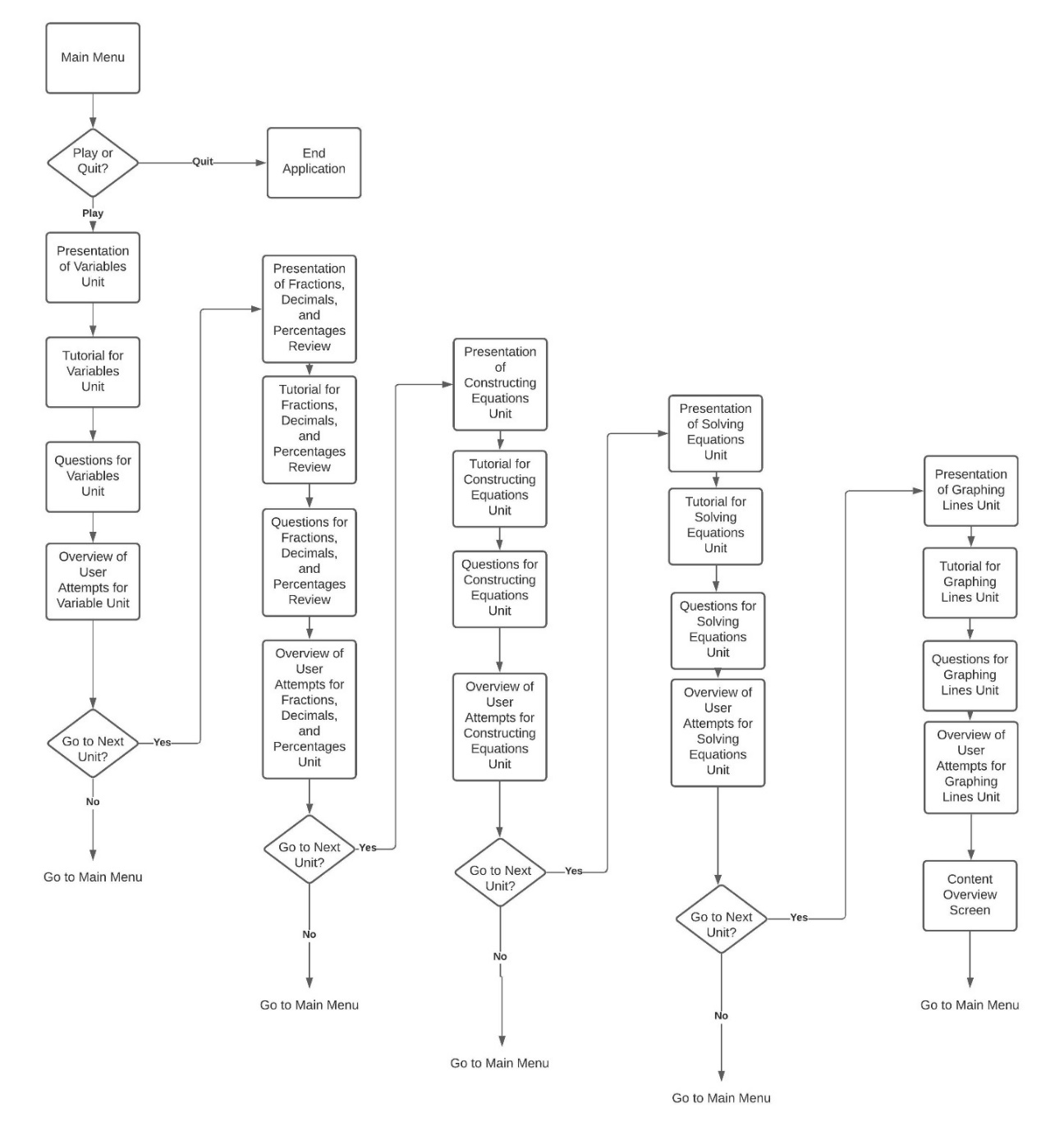
Operating System: Windows 10

Algebra Tutor was developed on and for Windows machines. I chose Windows as this game’s development platform because it is the operating system I am the most familiar with, thus simplifying the development process. Windows is also the most popular desktop operating system in terms of use (see the Statista link in the References section for more information). For similar reasons I intended Windows to be the game’s target platform, particularly due to me having prior knowledge on how to launch programs on Windows machines.

Design & Storyboard

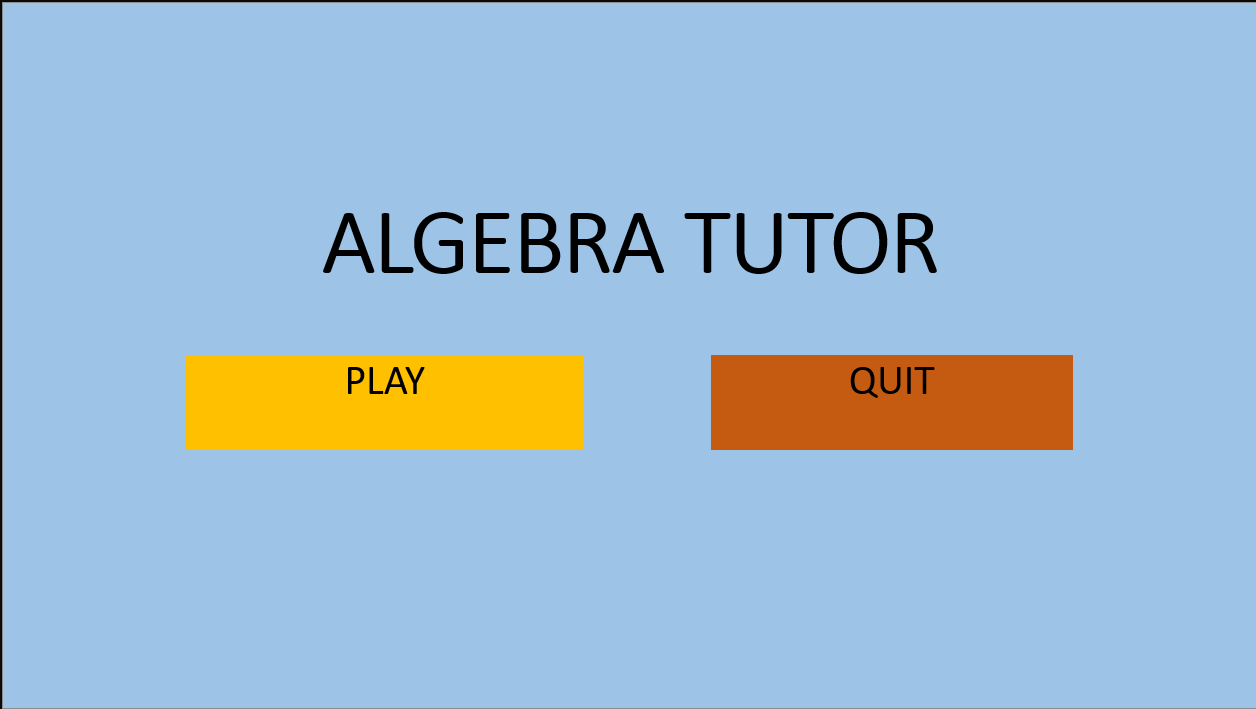
The following section will provide the flowchart for Algebra Tutor, as well as an updated storyboard reflecting the final product.

**Flowchart:** an overview of the flow of control of the tutor. Each of lesson consists of a presentation, a tutorial, and 3 questions. Completing all three questions will take the user to a screen that outlines the attempts it took for the user to answer each question correctly.



**Updated Storyboard**

**Main Menu**

****

This is the first screen the user will see upon launching the program. The user can use the mouse to click the “Play” button or the “Quit” button.

The user can do the following:

* Click Play
  + Goes into the Lesson 1 Presentation if first use
  + Goes into the current lesson if continuing from last use
* Click Quit
  + Close the application

**Lesson 1 Presentation:**

**Chart, box and whisker chart

Description automatically generated**

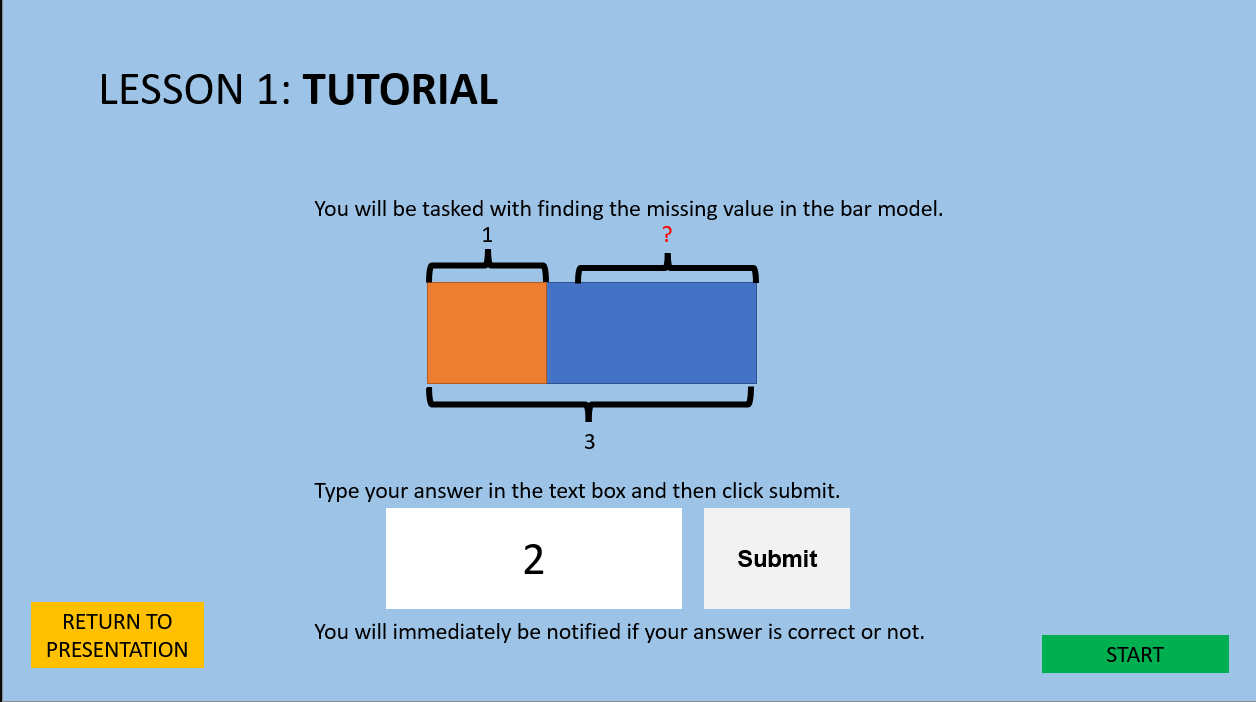
This screen will display a presentation of the content for the topic “Introduction of Variables”.

The user can click “Return to Main Menu” to be taken back to the main menu or can click “Next” to go to Lesson 1 Tutorial.

The user can do the following:

* Click “Return to Main Menu”
  + Be taken back to Main Menu
* Click “Next”
  + Be taken to Lesson 1 Tutorial

**Lesson 1 Tutorial**

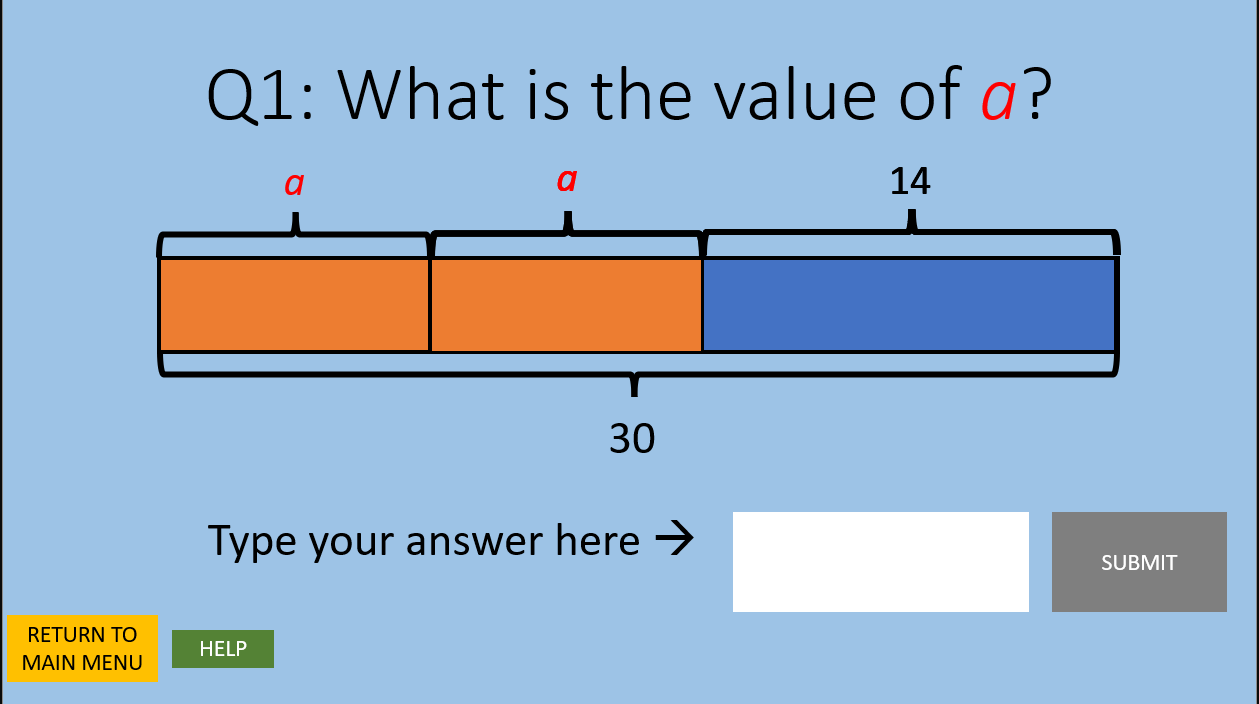


This screen will show the user how to answer questions for this lesson. After reading, the user can go into the questions by clicking “Start”, or return to the presentation by clicking “Return To Presentation”.

The user can do the following:

* Click “Start”
  + Go to Lesson 1 Questions
* Click “Return to Presentation”
  + Go to Lesson 1 Presentation

**Lesson 1 Questions (No Answer Given by the User)**

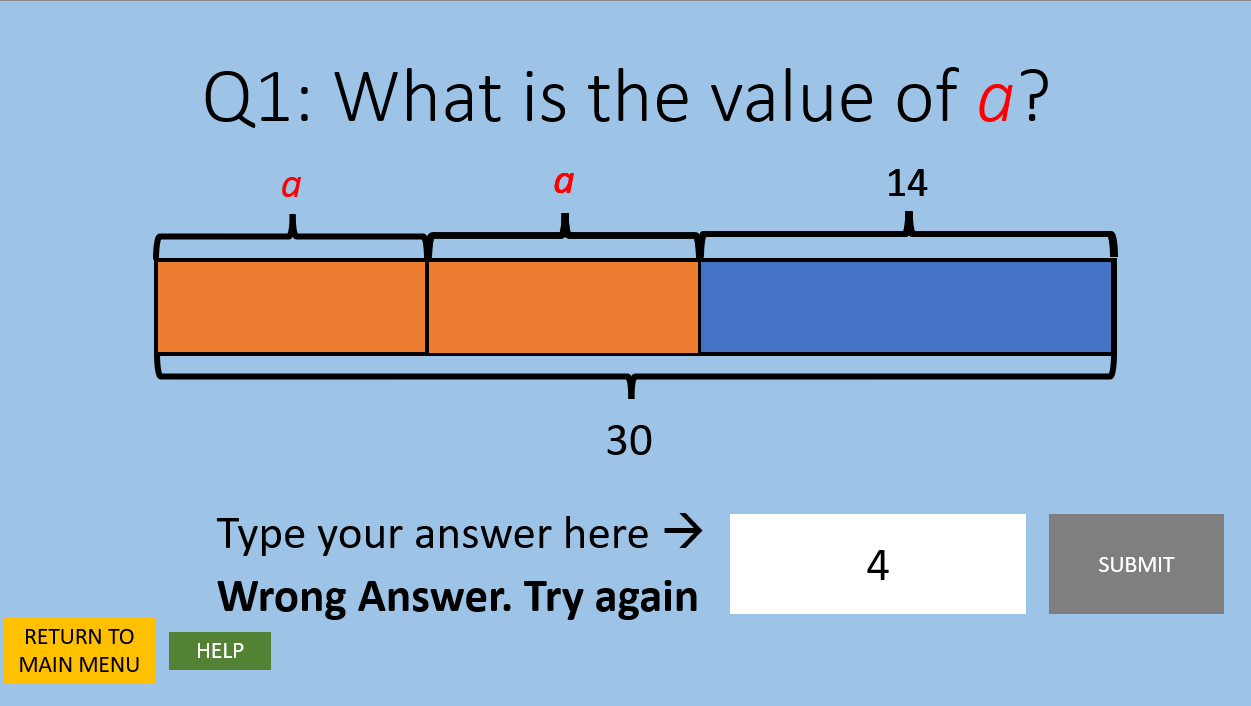
****

The user is tasked with answering three questions of the style above. The user must type the answer in the text box and click “Submit” to submit the answer. If correct, a pop-up will notify the user that the answer is correct. If incorrect, a pop-up will notify the user that the answer is incorrect. The user can click “Return to Main Menu” to return to the Main Menu or click “Help” to go the question’s help screen.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s hint screen
* Type in the text box
  + Show value typed
* Click “Submit”
  + If correct,
    - Show pop-up notifying user the answer is correct
  + If incorrect,
    - Show pop-up notifying user the answer is incorrect

**Lesson 1 Questions (Wrong Answer Submitted)**

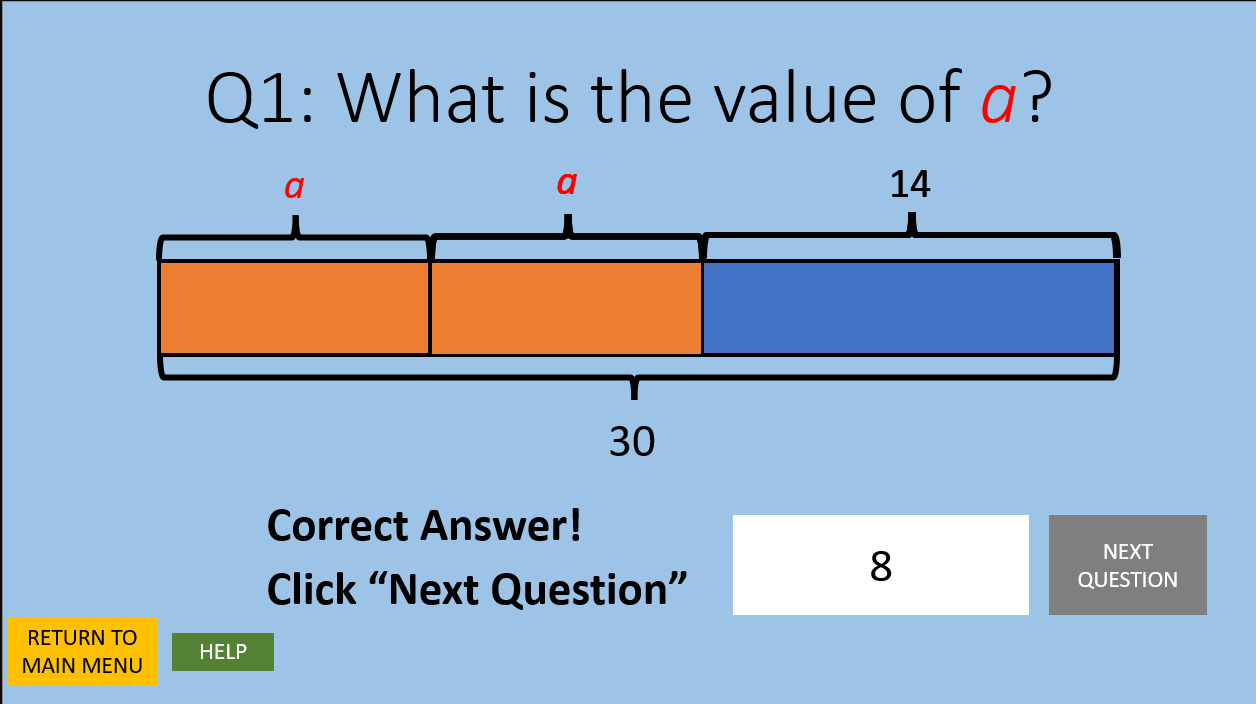
****

In the event that the user submits the wrong answer, the wrong answer prompt will be displayed.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s hint screen
* Type in the text box
  + Show value typed
* Click “Submit”
  + If correct,
    - Show pop-up notifying user the answer is correct
    - Disable “Submit”
    - If on question 1 or 2
      * Enable “Next Question”
    - If on question 3
      * Enable “See Results”
  + If incorrect,
    - Show pop-up notifying user the answer is incorrect

**Lesson 1 Questions (Correct Answer Submitted)**

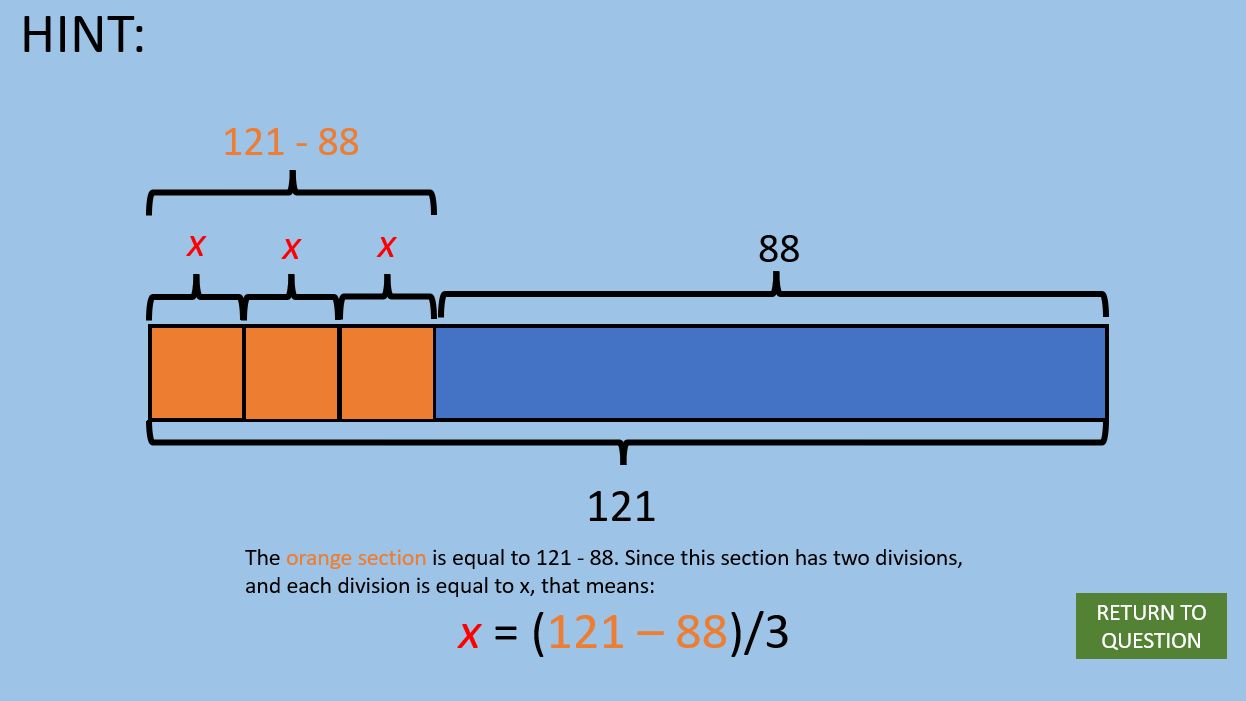
****

In the event that the user submits the right answer, the correct answer prompt will be displayed and the “Next Question” button will replace the “Submit” button. If the user is on the third question of the lesson, the “See Results” button will replace the “Submit” button instead.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s help screen
* Click “Next Question”
  + Go to the next question.
* If on the last question:
  + Click “See Results”
  + Go to Overview of Attempts Screen.

**Sample Hint Screen for Lesson 1**

****

The user will see guidance for the question at hand. The user can click “Return to Question” to return to the current question.

The user can do the following:

* Click “Return to Question”
  + Return user to the question screen for the current question.

**Lesson 2 Presentation (1)**

**Text

Description automatically generated**

This screen will display a presentation of the content for the topic “Fractions, Decimals, and Percentages”. The user can click “Return to Main Menu” to be taken back to the main menu or can click “Next” to go to Lesson 2 Presentation (2).

The user can do the following:

* Click “Return to Main Menu”
  + Be taken back to Main Menu
* Click “Next”
  + Be taken to Lesson 2 Presentation (2).

**Lesson 2 Presentation (2)**

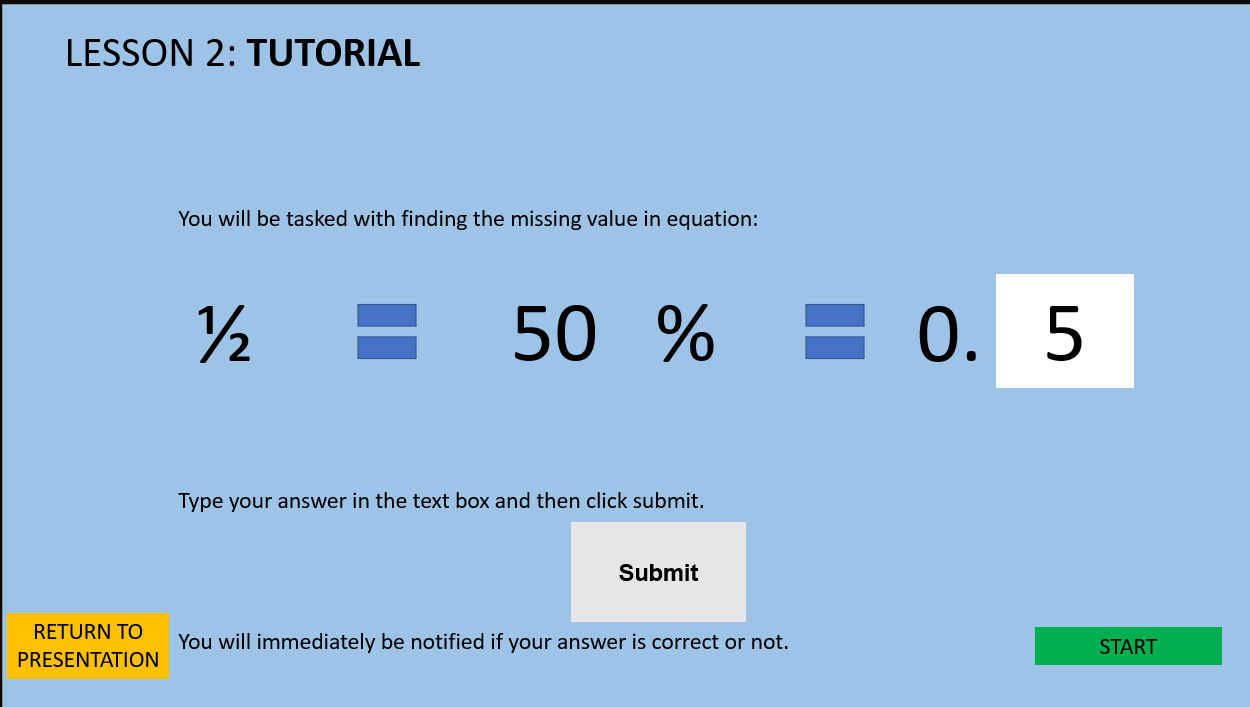
****

This screen will display a presentation of the content for the topic “Fractions, Decimals, and Percentages”. The user can click “Back” to be taken to Lesson 2 Presentation (1) or can click “Next” to go to Lesson 2 Tutorial.

The user can do the following:

* Click “Back”
  + Be taken back to Lesson 2 Presentation (1)
* Click “Next”
  + Be taken to Lesson 2 Tutorial

**Lesson 2 Tutorial**

****

This screen will show the user how to answer questions for this lesson. After reading, the user can go into the questions by clicking “Start”, or return to the presentation by clicking “Return To Presentation”.

The user can do the following:

* Click “Start”
  + Go to Lesson 2 Questions
* Click “Return to Presentation”
  + Go to Lesson 2 Presentation (2).

**Lesson 2 Questions**

Graphical user interface

Description automatically generated

The user is tasked with answering three questions of the style above. The user must type the answer in the text box and click “Submit” to submit the answer. If correct, a pop-up will notify the user that the answer is correct. If incorrect, a pop-up will notify the user that the answer is incorrect. The user can click “Return to Main Menu” to return to the Main Menu or click “Help” to go the question’s hint screen.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s hint screen
* Type in the text box
  + Show value typed
* Click “Submit”
  + If correct,
    - Show pop-up notifying user the answer is correct
    - Disable “Submit”
    - If question 1 or 2
      * Enable “Next Question”
    - If question 2
      * Enable “See Results”
  + If incorrect,
    - Show pop-up notifying user the answer is incorrect

**Lesson 3 Presentation**

**Chart, box and whisker chart

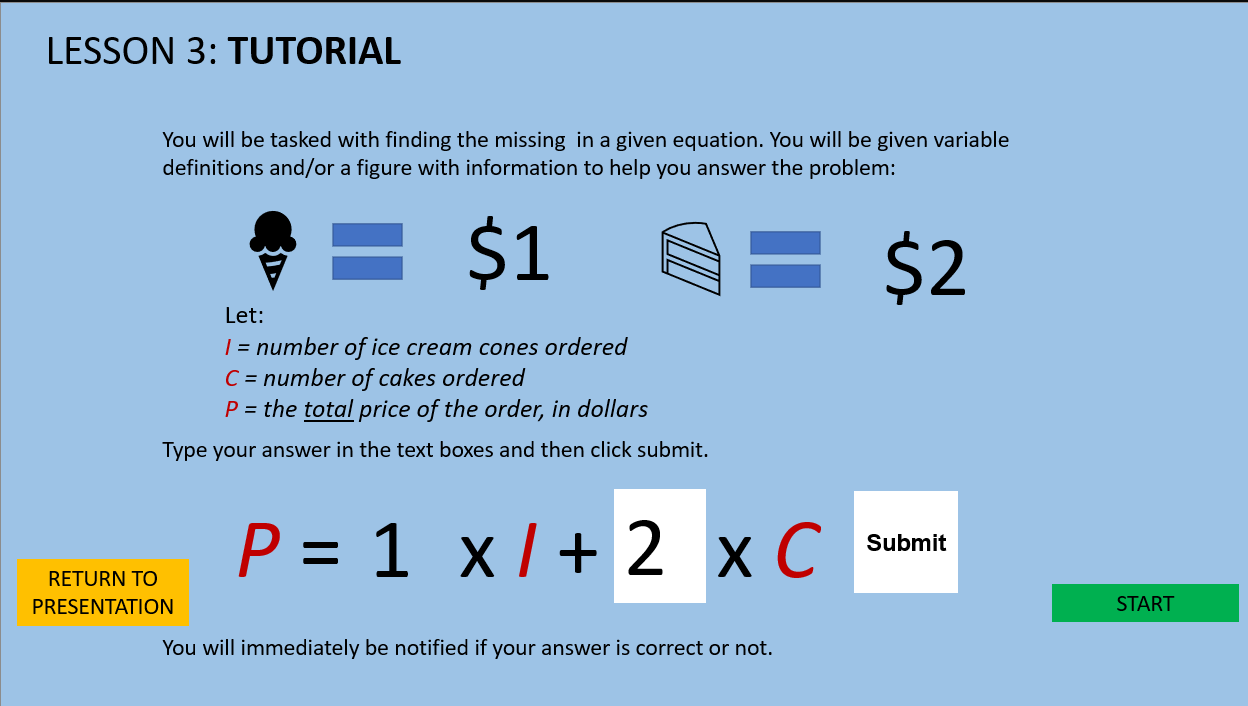
Description automatically generated**

This screen will display a presentation of the content for the topic “Constructing Equations”. The user can click “Return to Main Menu” to be taken back to the main menu or can click “Next” to go to Lesson 3 Tutorial.

The user can do the following:

* Click “Return to Main Menu”
  + Be taken back to Main Menu
* Click “Next”
  + Be taken to Lesson 3 Tutorial

**Lesson 3 Tutorial**

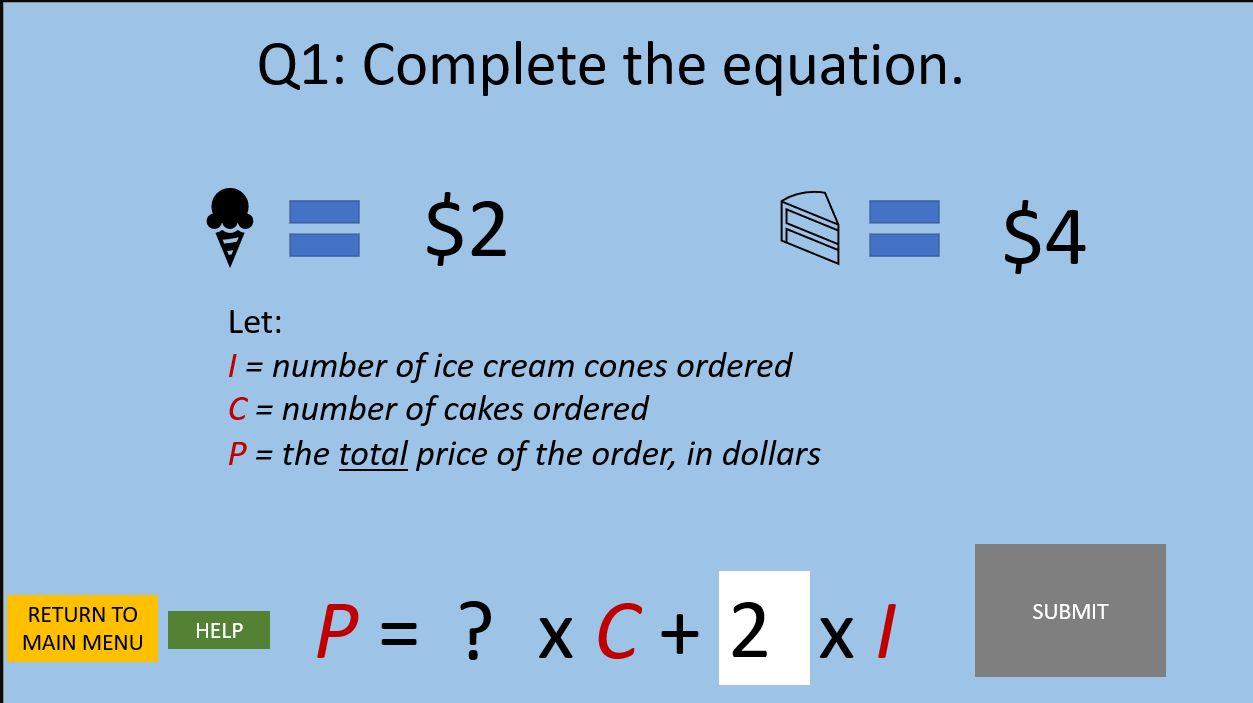
****

This screen will show the user how to answer questions for this lesson. After reading, the user can go into the questions by clicking “Start”, or return to the presentation by clicking “Return To Presentation”.

The user can do the following:

* Click “Start”
  + Go to Lesson 3 Questions
* Click “Return to Presentation”
  + Go to Lesson 3 Presentation.

**Lesson 3 Questions**

****

The user is tasked with answering three questions of the style above. The user must type the answer in the text box and click “Submit” to submit the answer. If correct, a pop-up will notify the user that the answer is correct. If incorrect, a pop-up will notify the user that the answer is incorrect. The user can click “Return to Main Menu” to return to the Main Menu or click “Help” to go the question’s hint screen.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s hint screen
* Type in the text box
  + Show value typed
* Click “Submit”
  + If correct,
    - Show pop-up notifying user the answer is correct
    - Disable “Submit”
    - If question 1 or 2
      * Enable “Next Question”
    - If question 2
      * Enable “See Results”
  + If incorrect,
    - Show pop-up notifying user the answer is incorrect

**Lesson 4 Presentation**

**Text

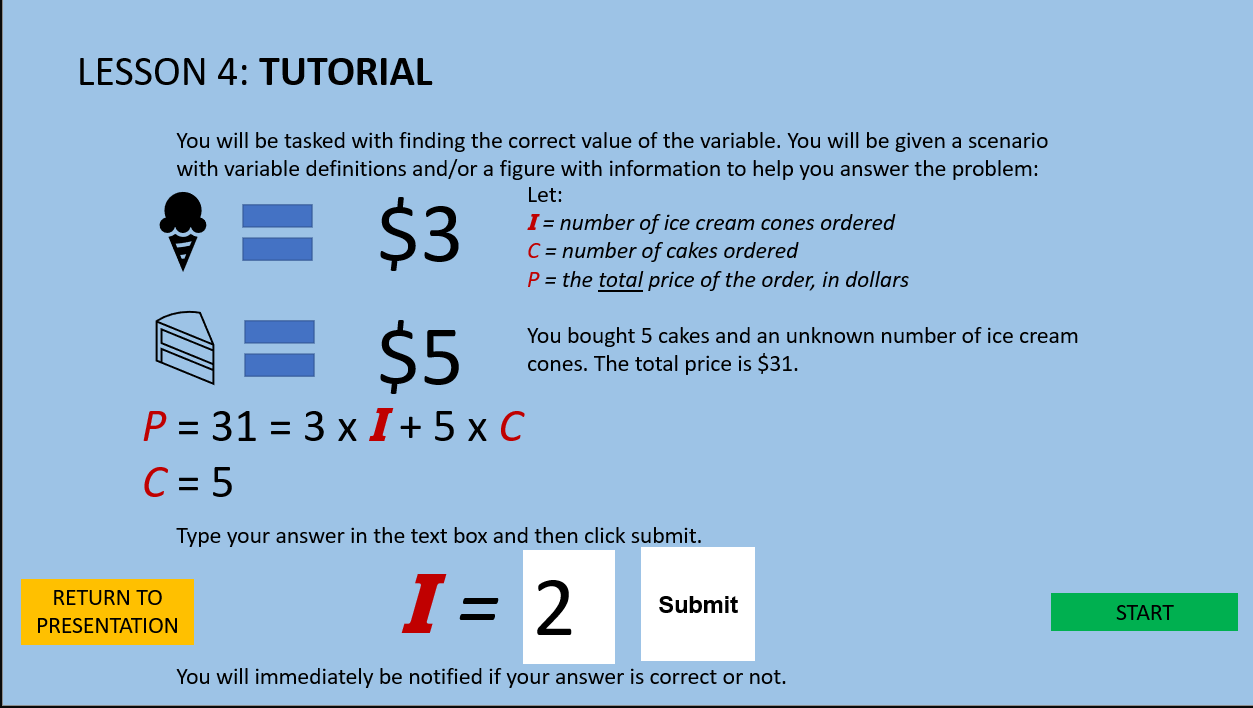
Description automatically generated**

This screen will display a presentation of the content for the topic “Solving Equations”. The user can click “Return to Main Menu” to be taken back to the main menu or can click “Next” to go to Lesson 4 Tutorial.

The user can do the following:

* Click “Return to Main Menu”
  + Be taken back to Main Menu
* Click “Next”
  + Be taken to Lesson 4 Tutorial

**Lesson 4 Tutorial**

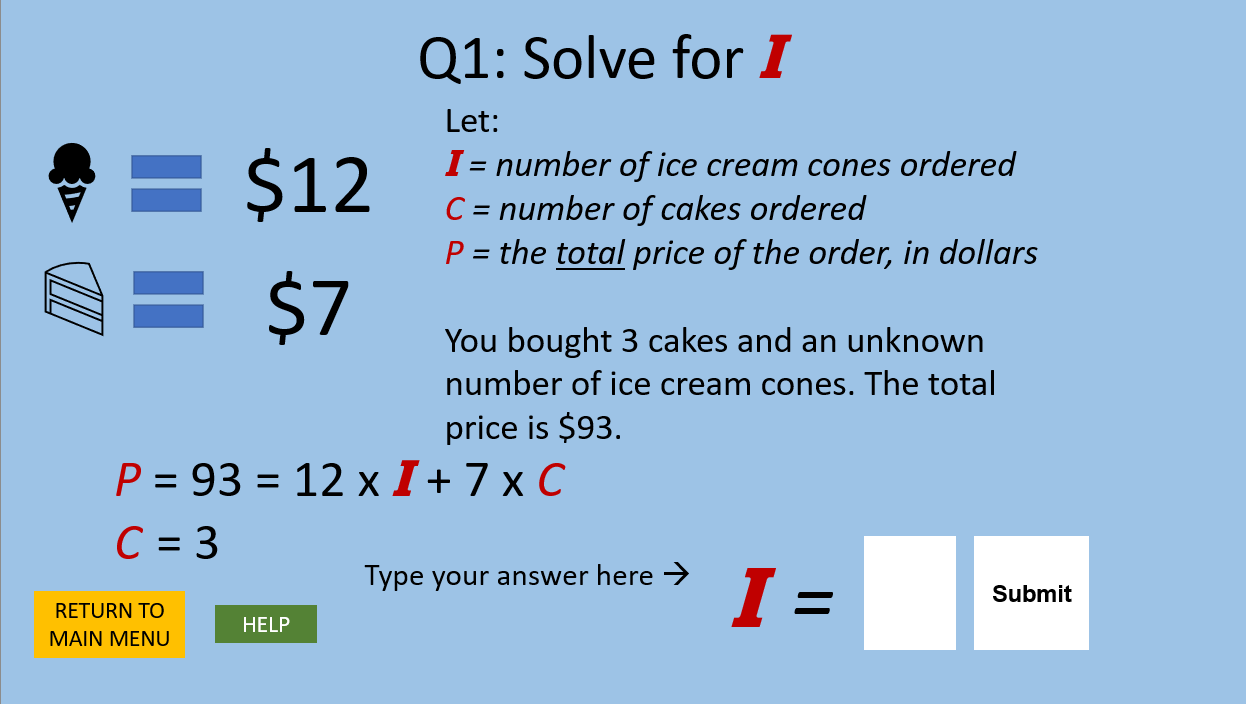
****

This screen will show the user how to answer questions for this lesson. After reading, the user can go into the questions by clicking “Start”, or return to the presentation by clicking “Return To Presentation”.

The user can do the following:

* Click “Start”
  + Go to Lesson 4 Questions
* Click “Return to Presentation”
  + Go to Lesson 4 Presentation.

**Lesson 4 Questions**

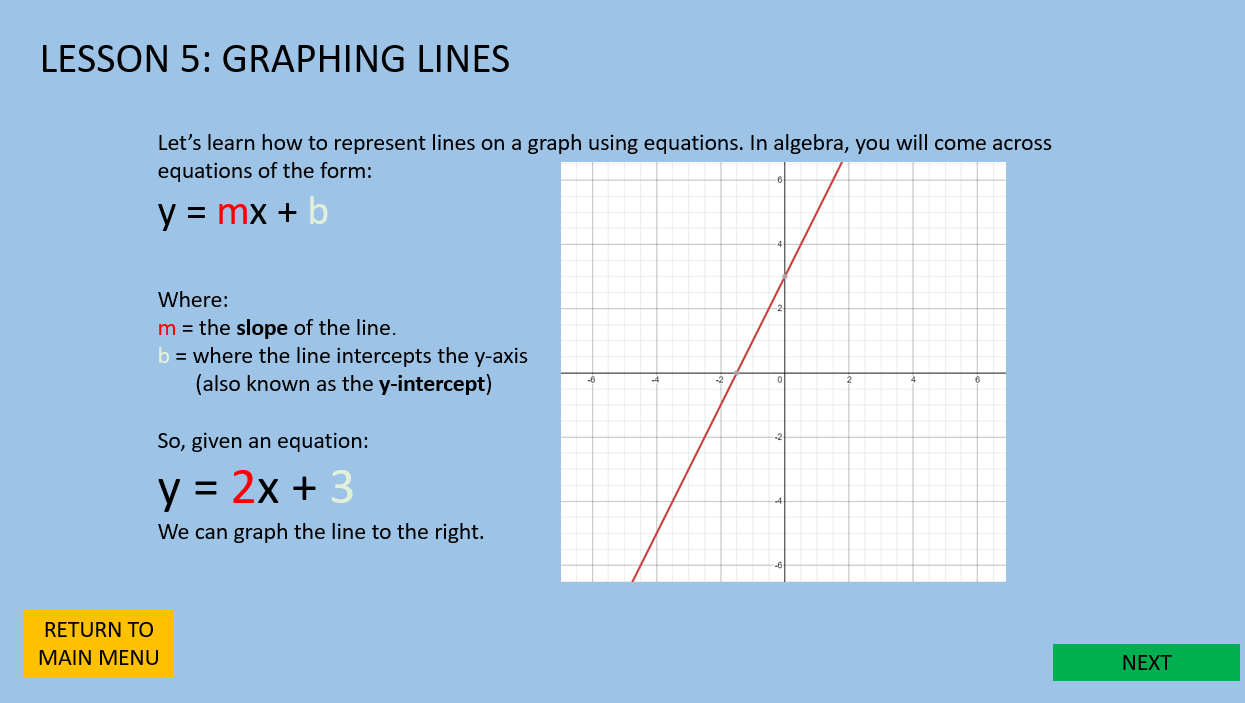
****

The user is tasked with answering three questions of the style above. The user must type the answer in the text box and click “Submit” to submit the answer. If correct, a pop-up will notify the user that the answer is correct. If incorrect, a pop-up will notify the user that the answer is incorrect. The user can click “Return to Main Menu” to return to the Main Menu or click “Help” to go the question’s hint screen.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s hint screen
* Type in the text box
  + Show value typed
* Click “Submit”
  + If correct,
    - Show pop-up notifying user the answer is correct
    - Disable “Submit”
    - If question 1 or 2
      * Enable “Next Question”
    - If question 2
      * Enable “See Results”
  + If incorrect,
    - Show pop-up notifying user the answer is incorrect

**Lesson 5 Presentation (1)**

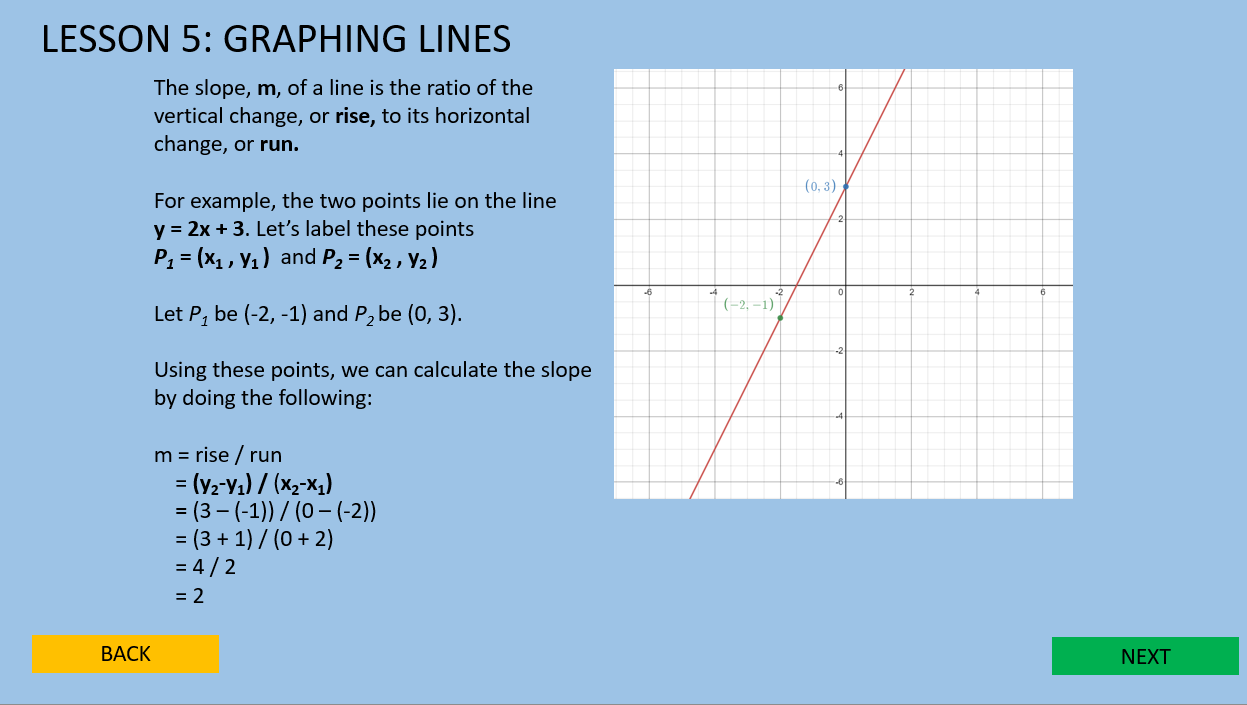
****

This screen will display a presentation of the content for the topic “Graphing Lines”. The user can click “Return to Main Menu” to be taken back to the main menu or can click “Next” to go to Lesson 5 Presentation (2).

The user can do the following:

* Click “Return to Main Menu”
  + Be taken back to Main Menu
* Click “Next”
  + Be taken to Lesson 5 Presentation (2).

**Lesson 5 Presentation (2)**

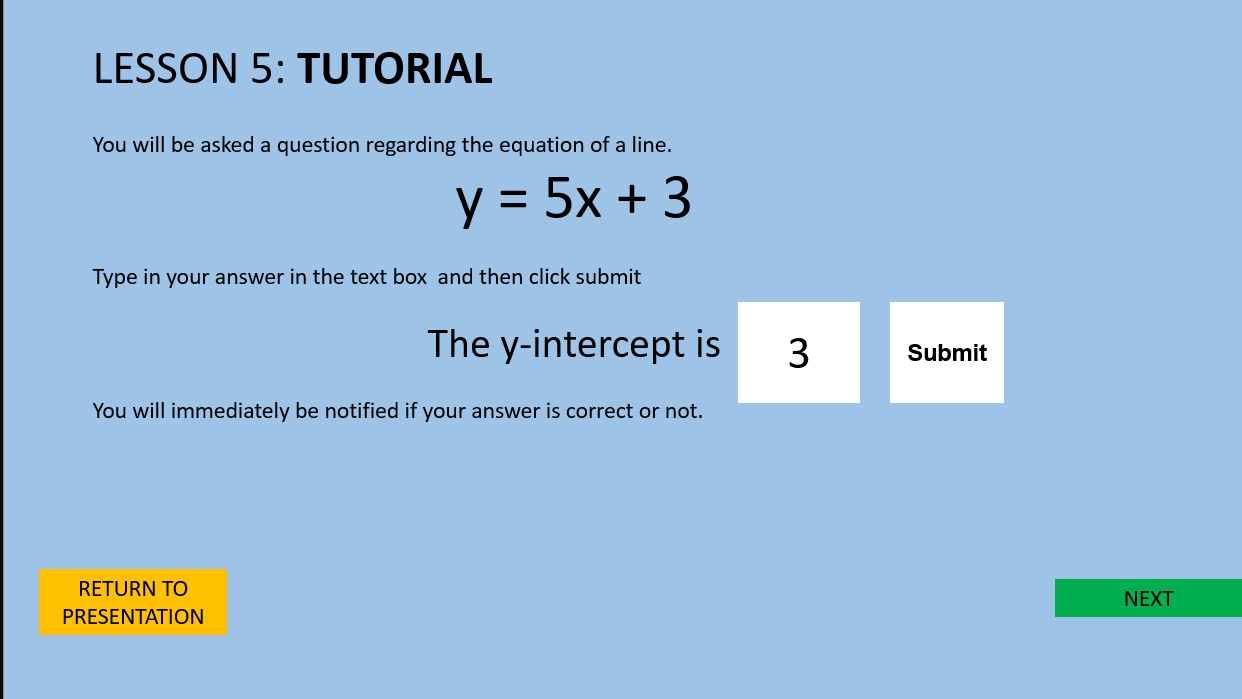
****

This screen will display a presentation of the content for the topic “Graphing Lines”. The user can click “Back” to be taken to Lesson 5 Presentation (1) or can click “Next” to go to Lesson 5 Tutorial.

The user can do the following:

* Click “Back”
  + Be taken back to Lesson 5 Presentation (1)
* Click “Next”
  + Be taken to Lesson 5 Tutorial

**Lesson 5 Tutorial**

****

This screen will show the user how to answer questions for this lesson. After reading, the user can go into the questions by clicking “Start”, or return to the presentation by clicking “Return To Presentation”.

The user can do the following:

* Click “Start”
  + Go to Lesson 5 Questions
* Click “Return to Presentation”
  + Go to Lesson 5 Presentation (2).

**Lesson 5 Questions**

**Text

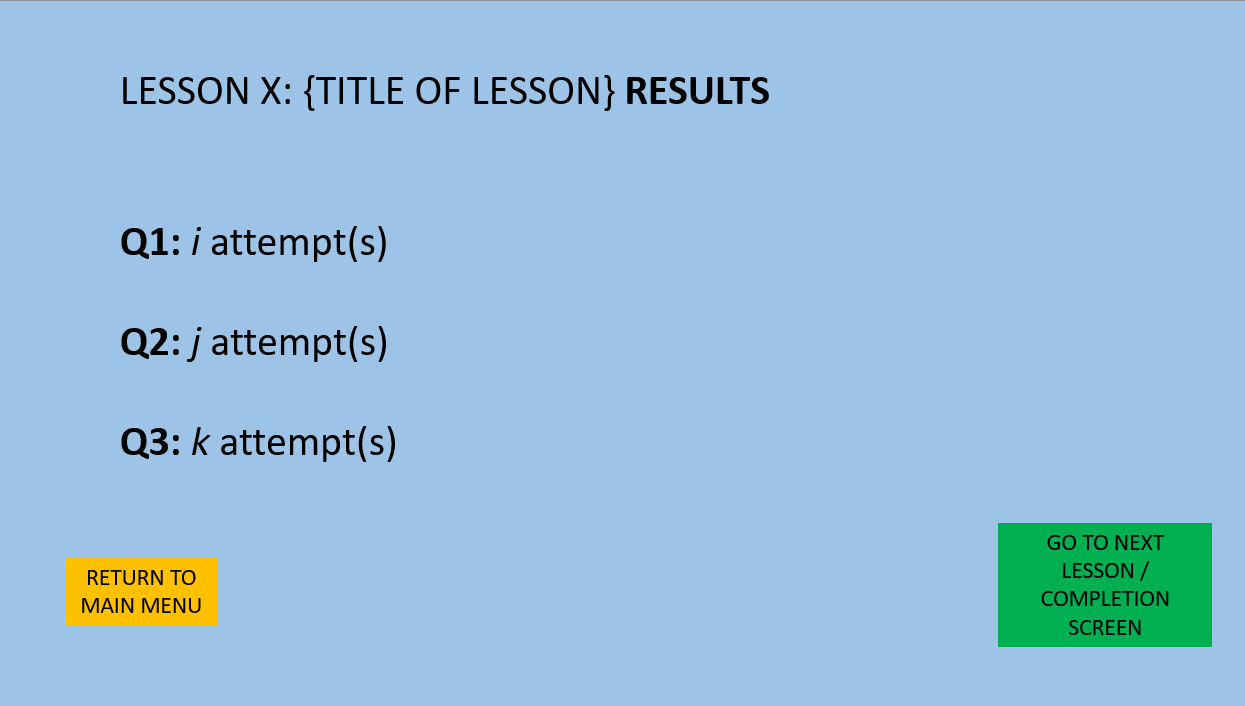
Description automatically generated with medium confidence**

The user is tasked with answering three questions of the style above. The user must type the answer in the text box and click “Submit” to submit the answer. If correct, a pop-up will notify the user that the answer is correct. If incorrect, a pop-up will notify the user that the answer is incorrect. The user can click “Return to Main Menu” to return to the Main Menu or click “Help” to go the question’s hint screen.

The user can do the following:

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Help”
  + Go to the question’s hint screen
* Type in the text box
  + Show value typed
* Click “Submit”
  + If correct,
    - Show pop-up notifying user the answer is correct
    - Disable “Submit”
    - If question 1 or 2
      * Enable “Next Question”
    - If question 2
      * Enable “See Results”
  + If incorrect,
    - Show pop-up notifying user the answer is incorrect

**Overview of Attempts Screen**

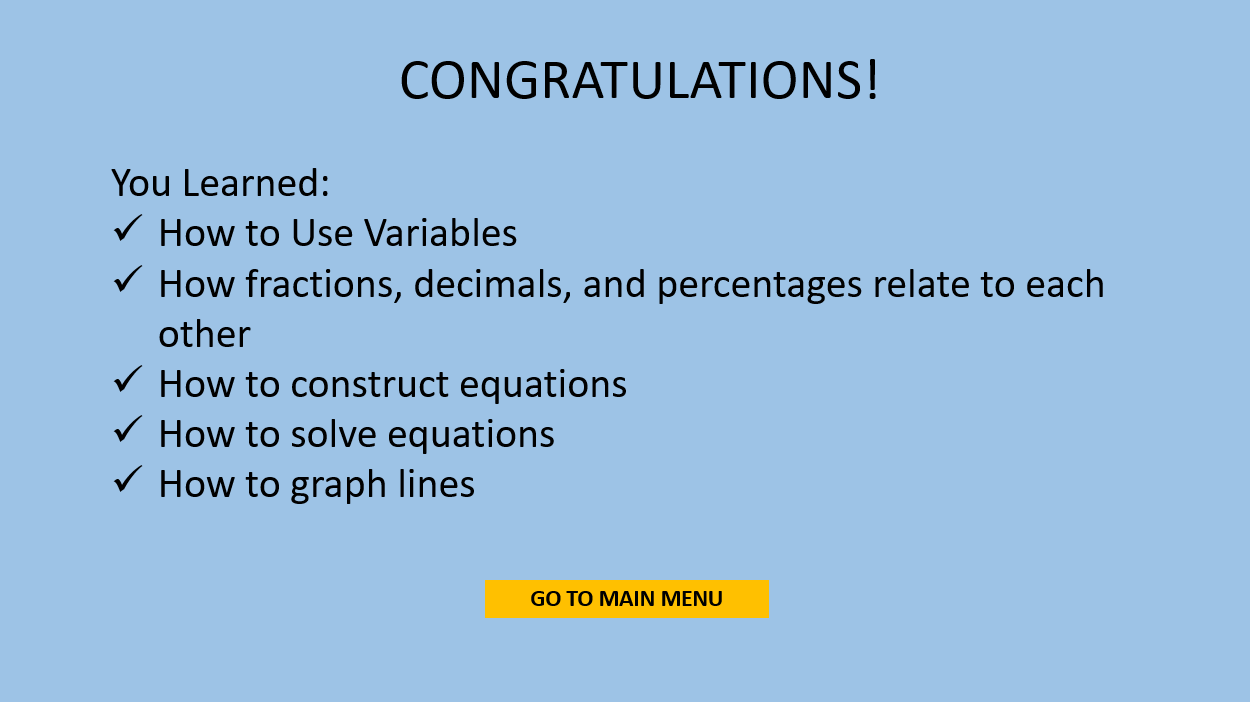
****

Here the user will see an overview of the number of attempts it took the user to answer each question correctly. The “X” will be the number of the current lesson, and i, j, and k will be the amount of attempts the user took on Q1, Q2, and Q3 respectively. The user can return to the main menu by click the “Return to Main Menu” button. If the user is on lesson 5, the button will read “Go to Completion Screen” and will take the user to the Completion Screen. If the user is not on lesson 5, the button will read “Go to Next Lesson” and will take the user to the next lesson.

The user can do the following.

* Click “Return to Main Menu”
  + Go to Main Menu
* Click “Go to Next Lesson” if not on lesson 5 and “Go the Completion Screen” if on lesson 5.
  + Go to the next lesson if the user is not on lesson 5.
  + Go to Completion Screen if on Lesson 5

**Completion Screen**

****

The user will see an overview of the topics they have practiced. The user can click “Go to Main Menu” to go to the Main Menu.

The user can do the following:

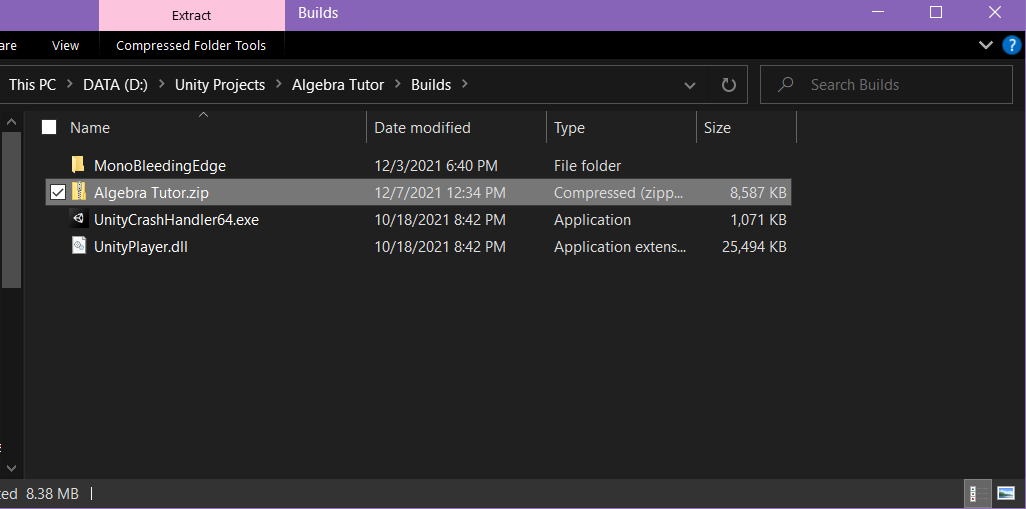
* Click “Go to Main Menu”
  + Go to Main Menu

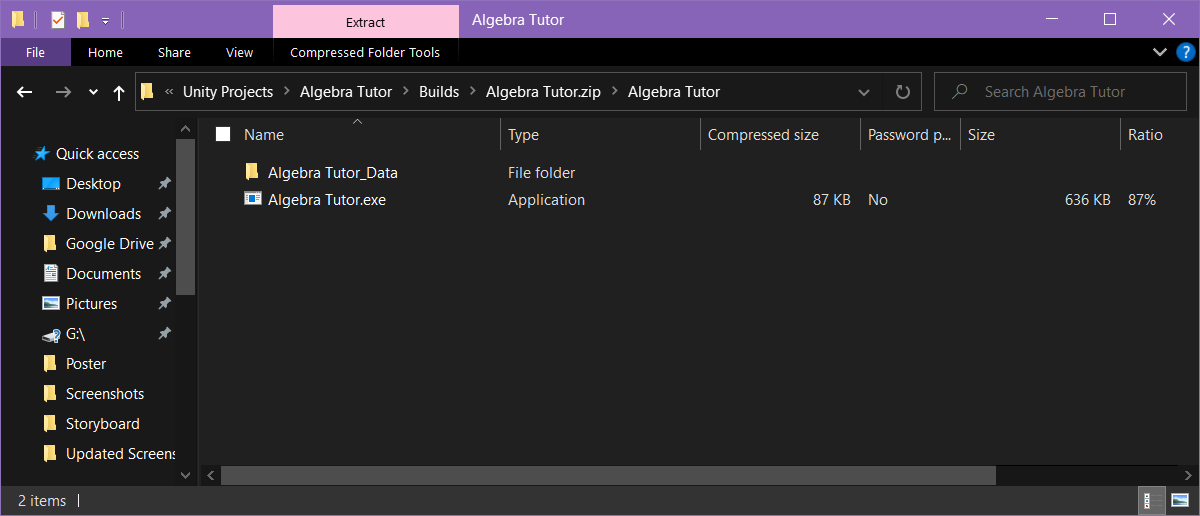
Installation

The following section consists of instructions for user and developer setup of Algebra Tutor on a Windows 10 machine.

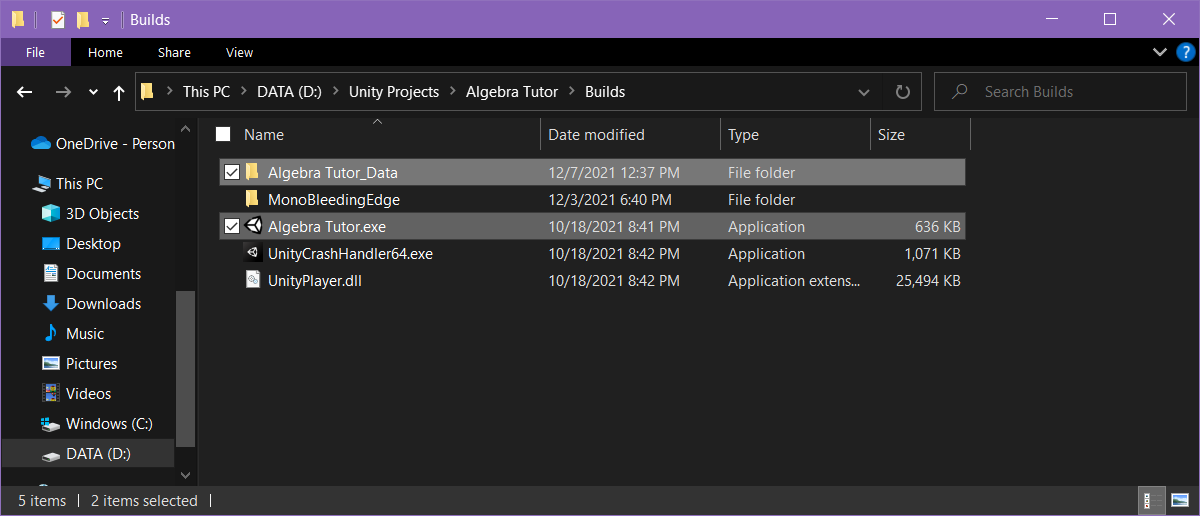
User setup

* Upon receiving the game file in AlgebraTutor.zip, extract the contents of the file: Algebra Tutor.exe and a folder named Algebra Tutor\_Data. **The .exe file and the \_Data folder MUST be in the same location for the game to work; the \_Data folder contains all the game’s files and the .exe file needs it to properly run the game.**

****

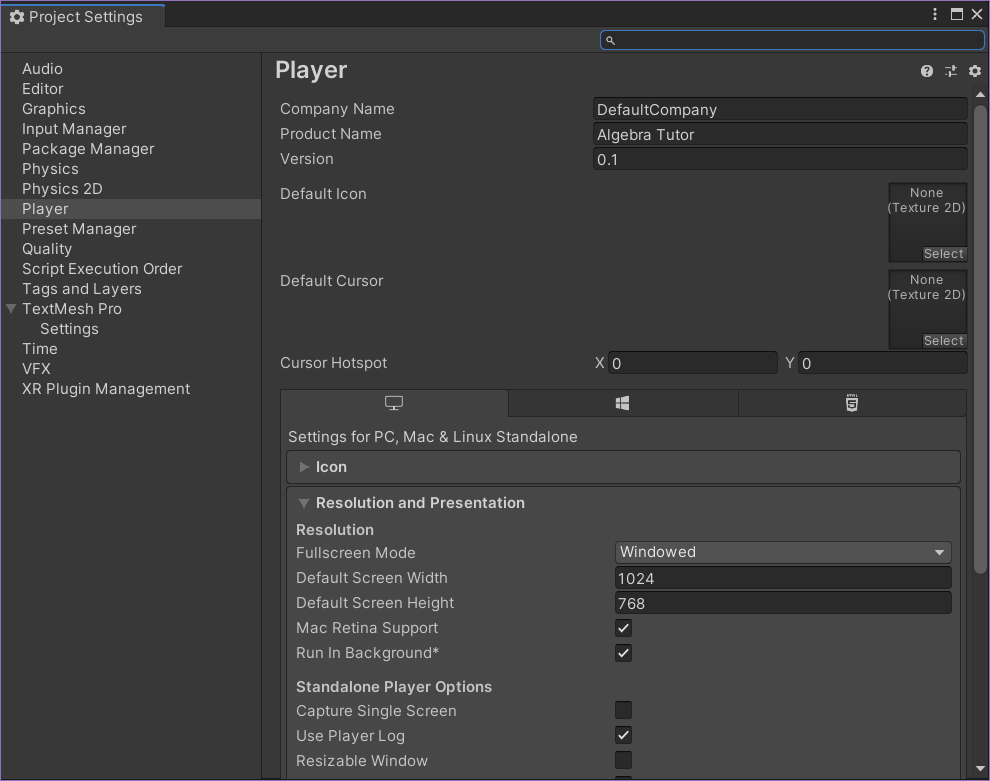
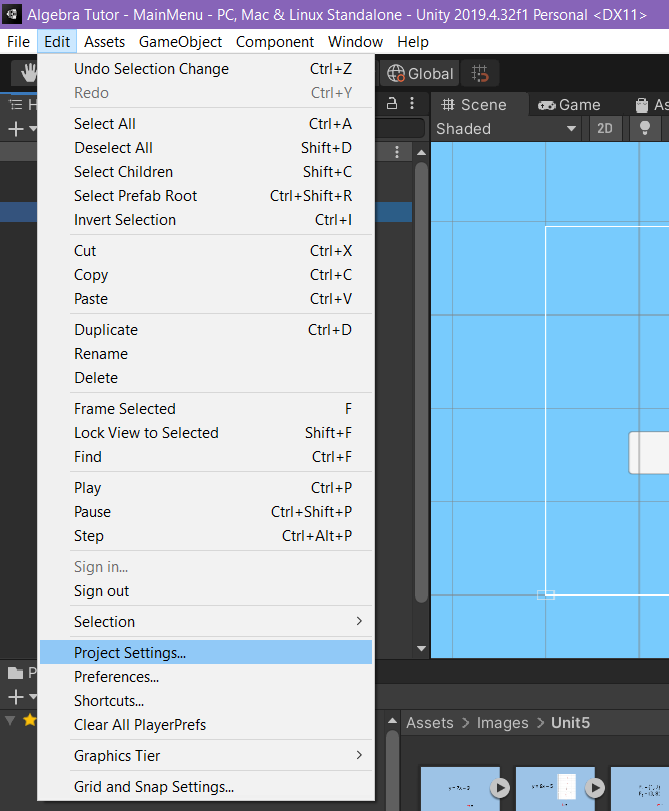


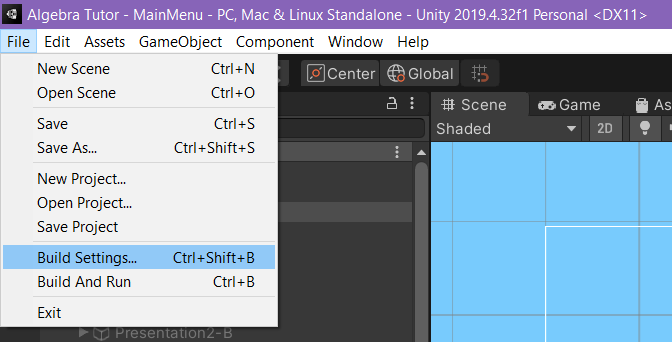
* Once you have confirmed that the two files are in the same location, click on Algebra Tutor.exe to launch the game.

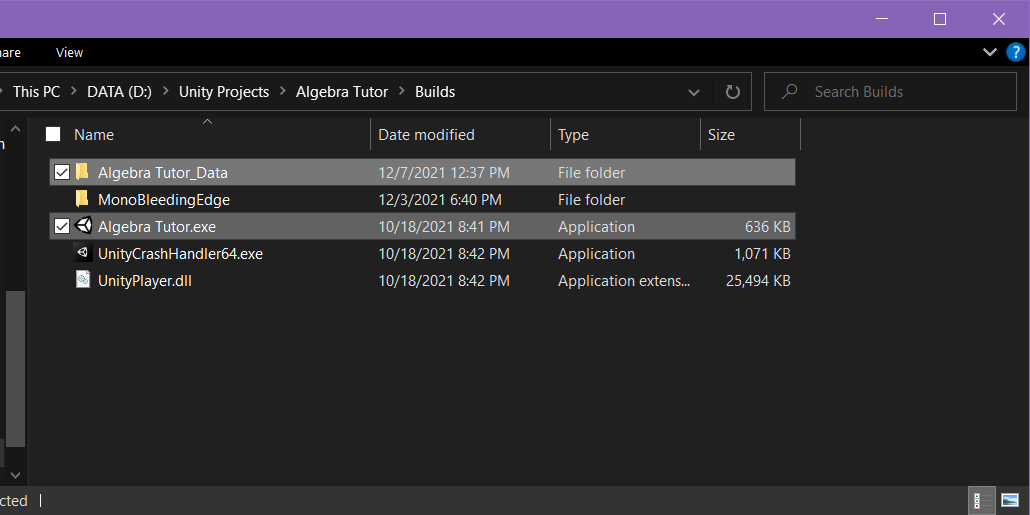


Developer setup (Instructions and screenshots are based on Unity version 2019.4.32f1)

* Launch the Unity Hub.
* Go to “Projects”.
* Click the Project containing the Algebra Tutor game. Make sure the Unity Version is set to **2019.4.32f1.**
* Once the Unity editor has been launched, make sure the game’s product name is “Algebra Tutor”. Check by going to Edit > Project Settings > Player.



* Change the Product Name to “Algebra Tutor” if it is not already named as such. Close the Product Settings Window.
* Go to File > Build Settings. This will launch the Build Settings window.
* In the Build Settings window, make sure that under “Scenes in Build”, “Scenes/Main Menu” is listed. If not, click “Add Open Scenes” to add it.
* For “Platform”, choose “PC, Mac and Linux Standalone”
* For “Target Platform” choose “Windows”.
* For “Architecture” choose “x86\_64”.
* At this point the Build Settings window should look like this:
* Click the “Build” button. Unity will open File Explorer to ask you where you want the build files (Algebra Tutor.exe and the folder Algebra Tutor\_Data) to be stored. Choose the folder of your choice and click “Select Folder”.
* Once Unity finishes building the game, you should find Algebra Tutor’s build files in the location you selected.

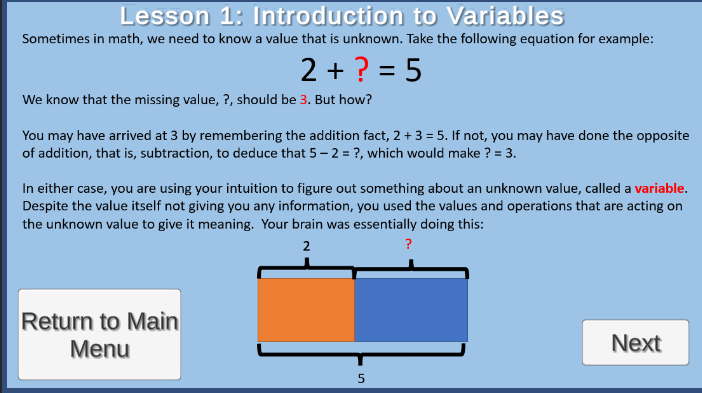


Sample Session

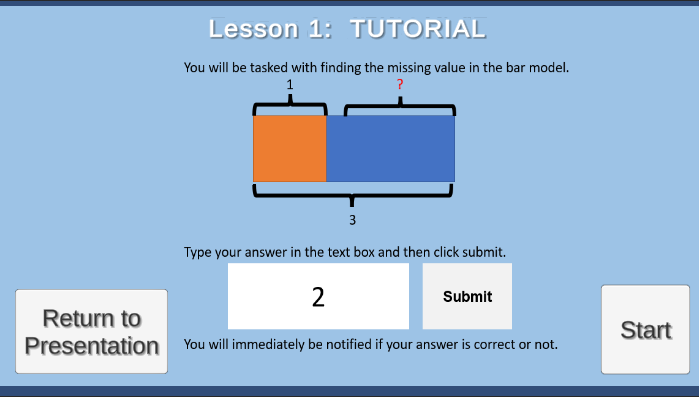
The following is a detailed description of a sample play session of Algebra Tutor. In this session, the user launches the game, clicks play, and goes through the lessons in order. After all the five lessons have been completed, the completion screen is shown and the user is returned to the main menu where they click “Quit” to close the application.



Upon launching the tutor, the main menu is displayed. The player clicks “Play” to go to the first lesson



The presentation for Lesson 1 is displayed. The user clicks “Next” to go the Lesson 1 tutorial.



The tutorial for Lesson 1 is displayed. The user clicks “Start” to go to the first question of Lesson 1.

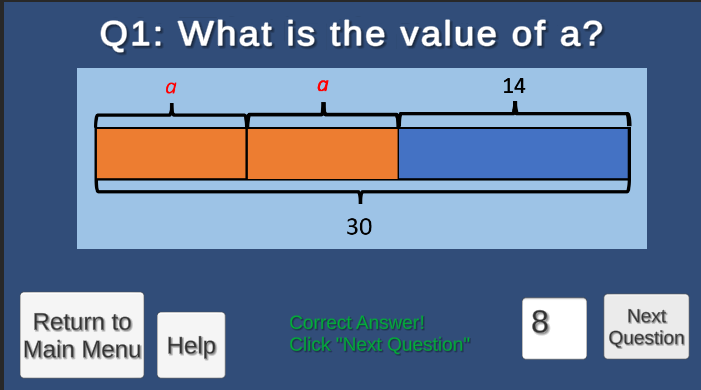


The first question for Lesson 1 is displayed. The user submitted a wrong answer and the wrong answer prompt is displayed. The user clicks “Help” to go to the question’s hint screen.

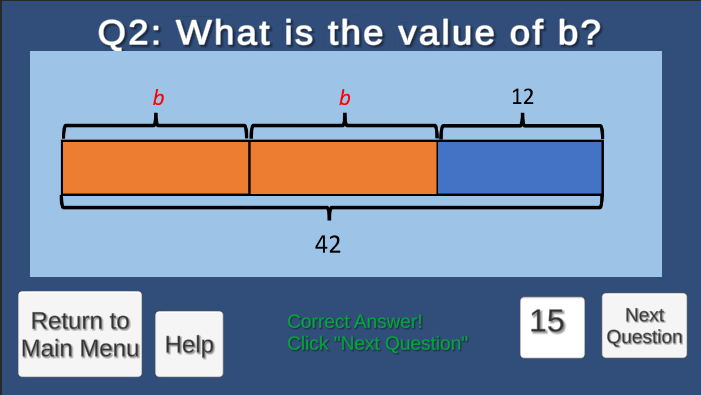
Chart, box and whisker chart

Description automatically generated

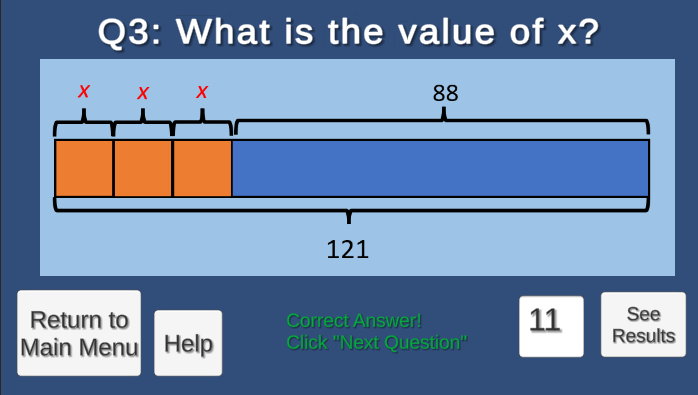
The hint screen for the first question is displayed. The user clicks “Return to Question” to return to the first question.



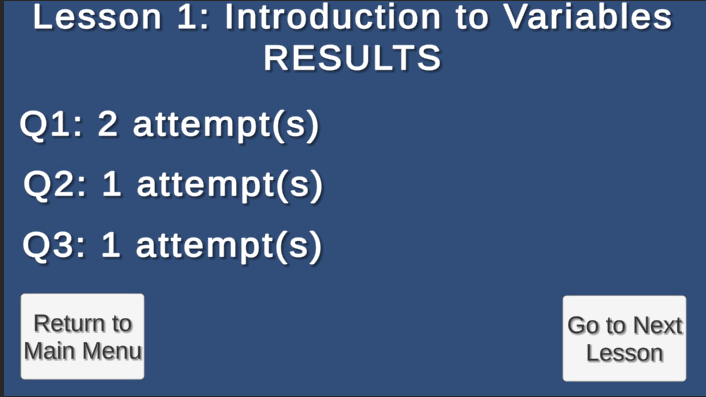
The first question for Lesson 1 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



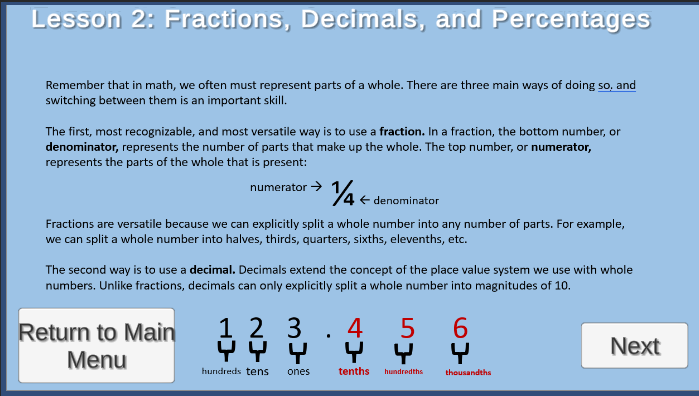
The second question for Lesson 1 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



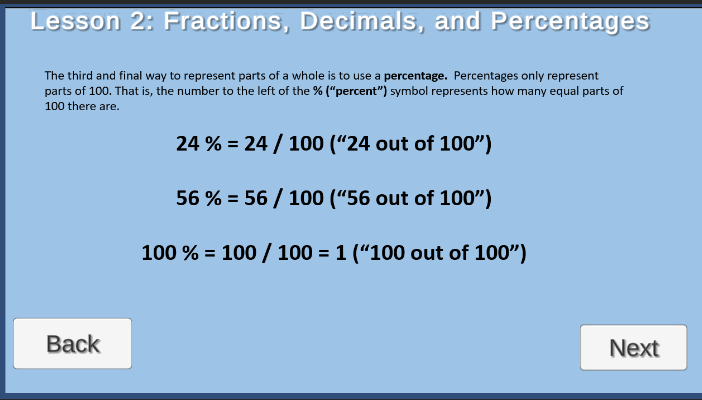
The third question for Lesson 1 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “See Results” to go to the results screen.



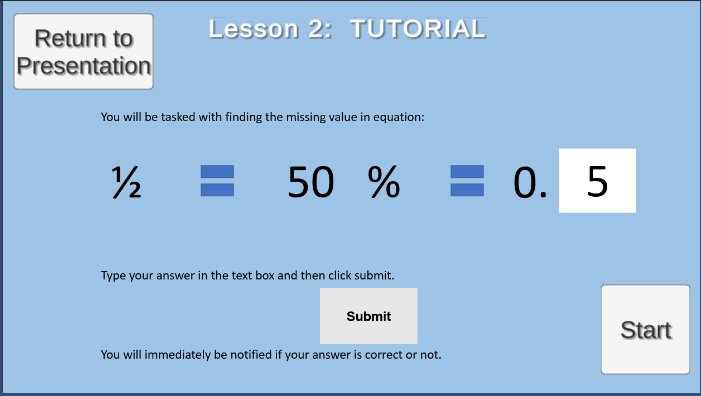
The user’s Lesson 1 results are displayed. The user clicks “Go to Next Lesson” to go to Lesson 2.



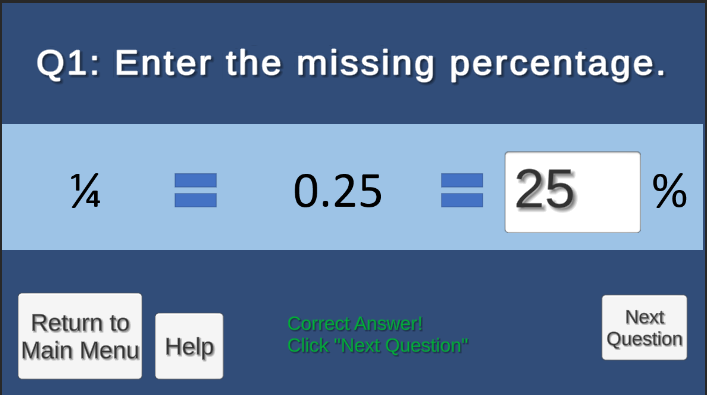
The first page presentation for Lesson 2 is displayed. The user clicks “Next” to go to the second page of the Lesson 2 Presentation.



The second page presentation for Lesson 2 is displayed. The user clicks “Next” to go to the second page of the Lesson 2 Tutorial.

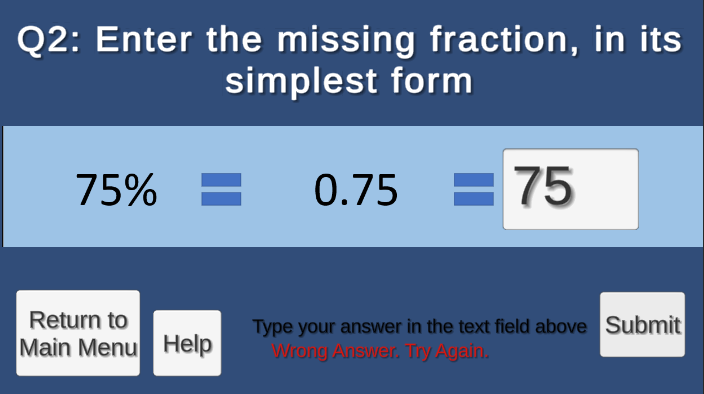


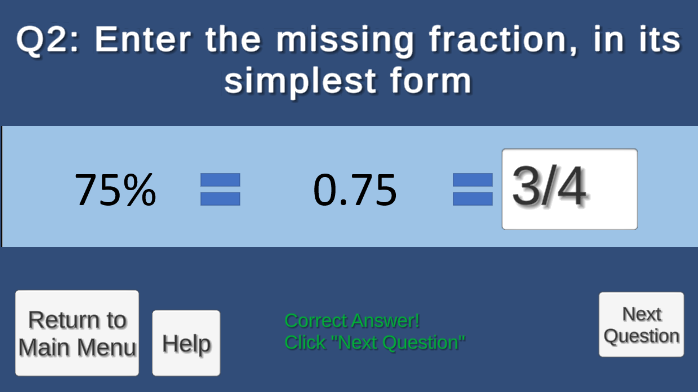
The tutorial for Lesson 2 is displayed. The user clicks “Start” to go to the first question of Lesson 2.



The second question for Lesson 2 is displayed. The user submitted a wrong answer and the wrong answer prompt is displayed.

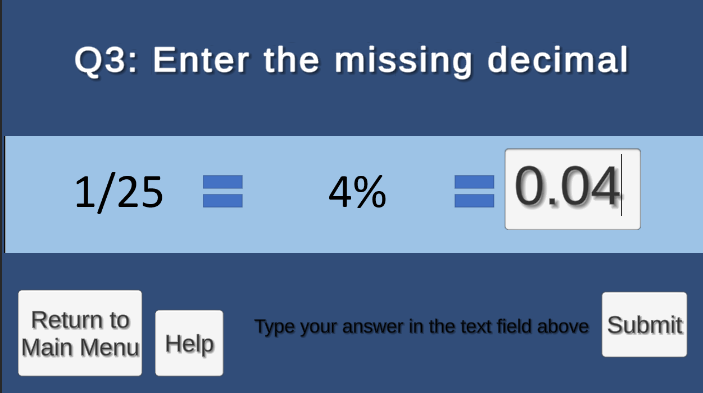
The first question for Lesson 2 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.





The third question for Lesson 2 is displayed. The user will submit the correct answer, and the correct answer prompt will be displayed. The “Submit” button will become the “See Results” button. The user will click the “See Results” button to go to the results screen.

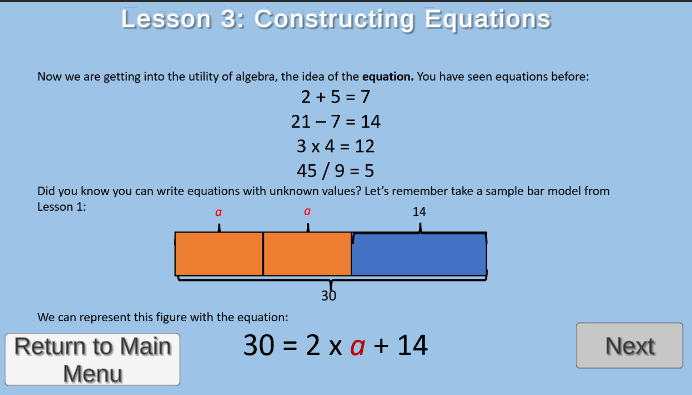
The second question for Lesson 2 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



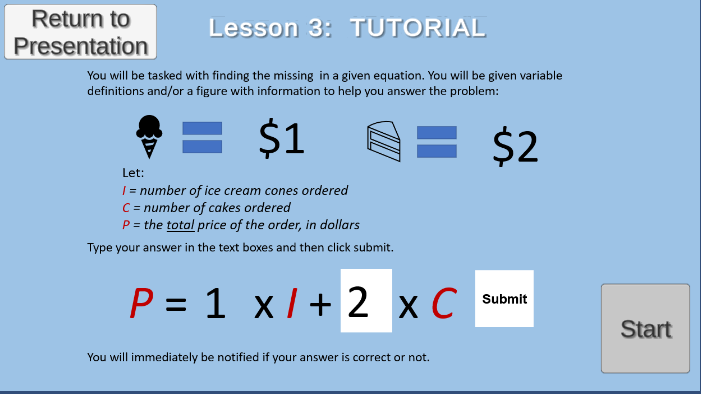
Graphical user interface, text, application

Description automatically generated

The user’s Lesson 2 results are displayed. The user clicks “Go to Next Lesson” to go to Lesson 3



The presentation for Lesson 3 is displayed. The user clicks “Next” to go the Lesson 3 tutorial.



The tutorial for Lesson 3 is displayed. The user clicks “Start” to go to the first question of Lesson 3.

Graphical user interface, text, application

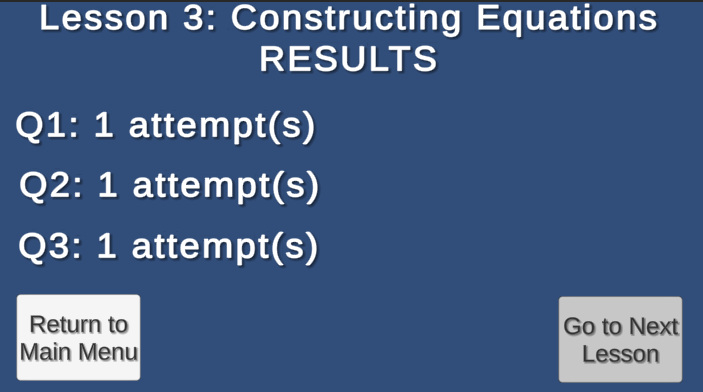
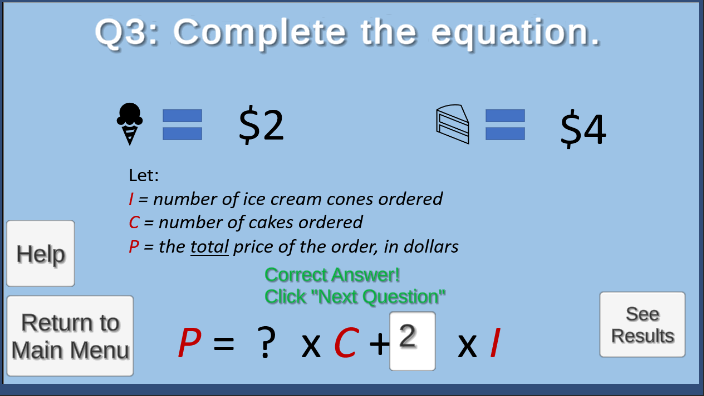
Description automatically generated

The first question for Lesson 3 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.

Graphical user interface, text, application

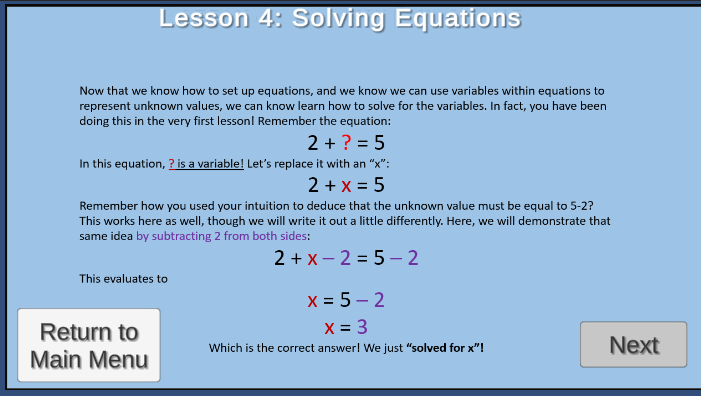
Description automatically generated

The second question for Lesson 3 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



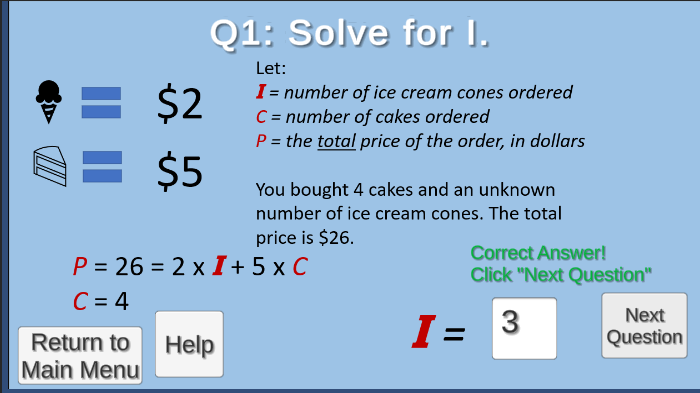
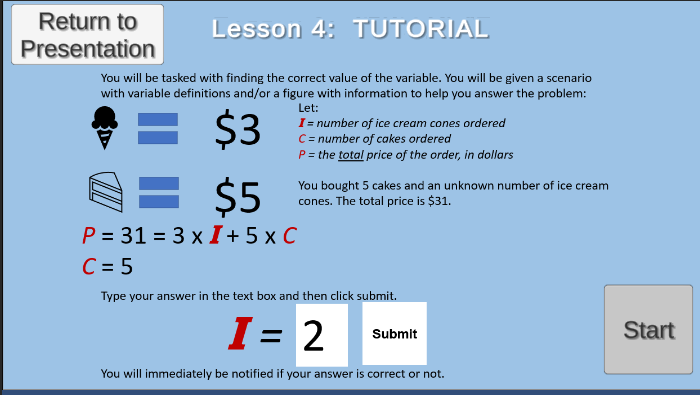
The user’s Lesson 3 results are displayed. The user clicks “Go to Next Lesson” to go to Lesson 4.

The third question for Lesson 3 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “See Results” to go to the results screen.

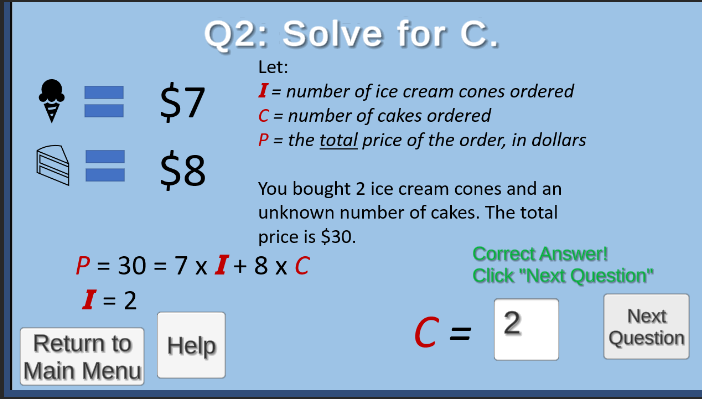


The tutorial for Lesson 4 is displayed. The user clicks “Start” to go to the first question of Lesson 4.

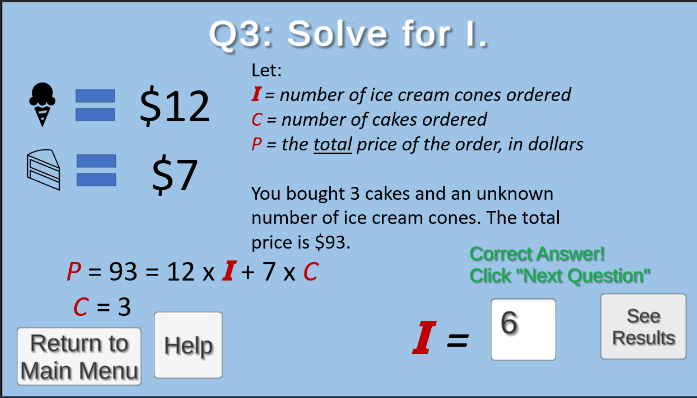
The presentation for Lesson 4 is displayed. The user clicks “Next” to go the Lesson 4 tutorial.



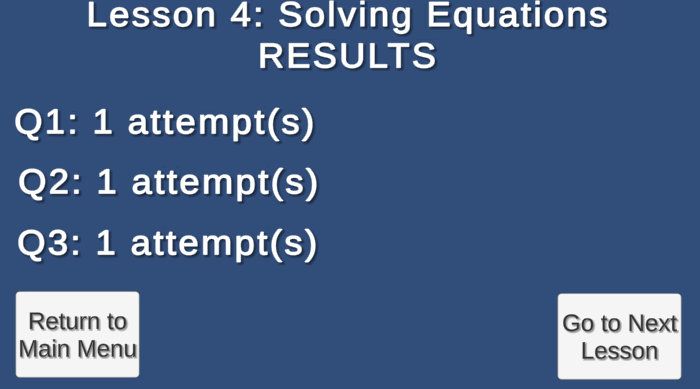
The first question for Lesson 4 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



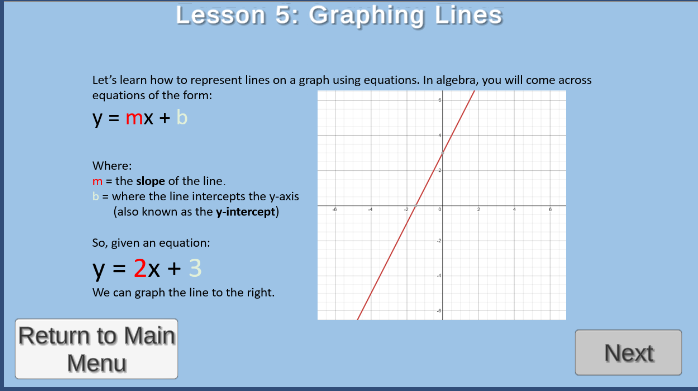
The second question for Lesson 4 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



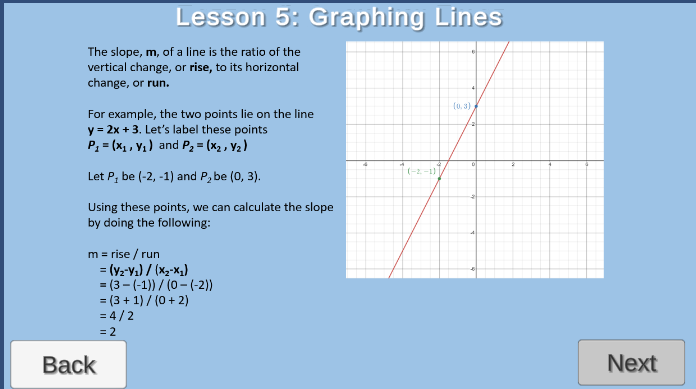
The third question for Lesson 4 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “See Results” to go to the results screen.



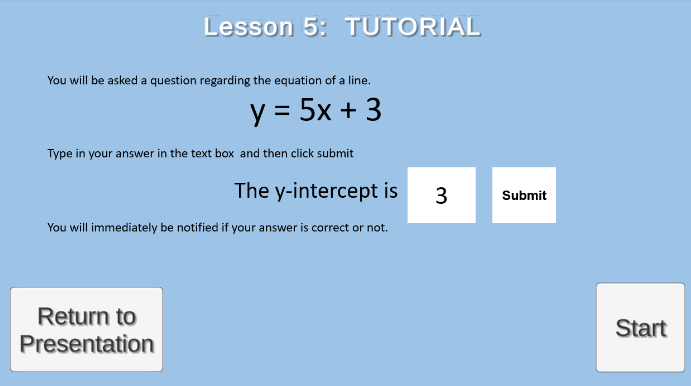
The user’s Lesson 4 results are displayed. The user clicks “Go to Next Lesson” to go to Lesson 5.



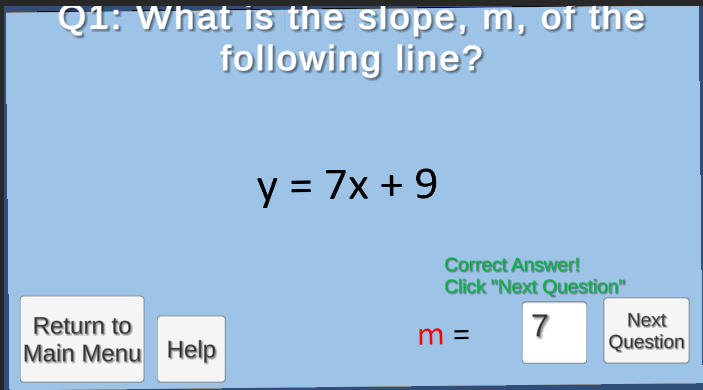
The first page presentation for Lesson 5 is displayed. The user clicks “Next” to go to the second page of the Lesson 5 Presentation.



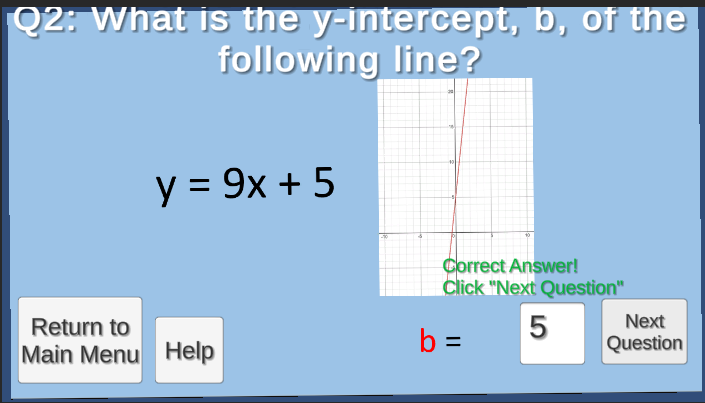
The second page presentation for Lesson 5 is displayed. The user clicks “Next” to go to the second page of the Lesson 5 Tutorial.



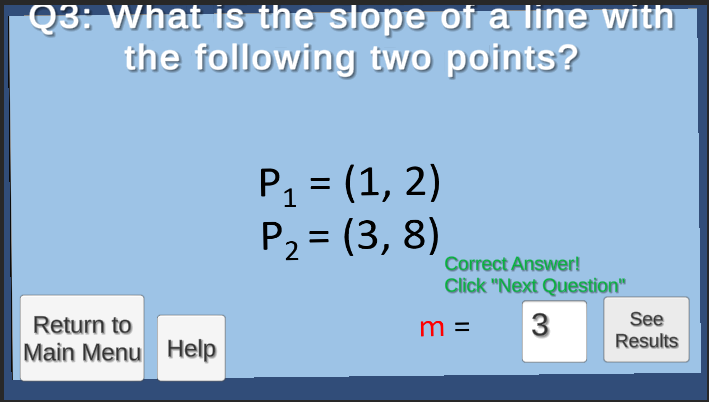
The tutorial for Lesson 5 is displayed. The user clicks “Start” to go to the first question of Lesson 5.



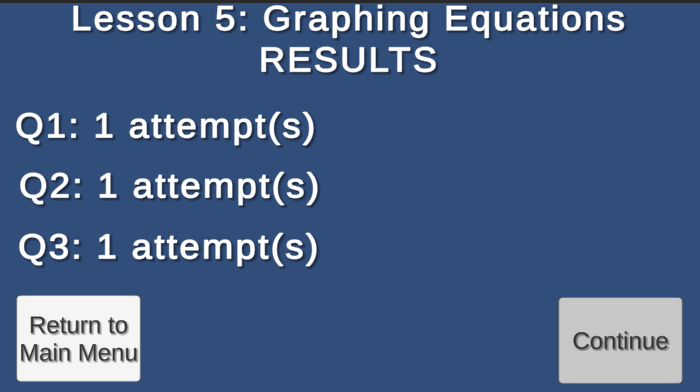
The first question for Lesson 5 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



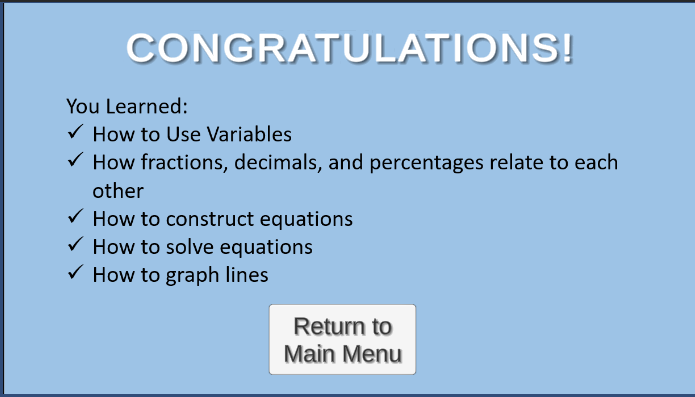
The second question for Lesson 5 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “Next Question” to go to the next question.



The third question for Lesson 5 is displayed. The user submitted the correct answer, and the correct answer prompt is displayed. The user clicks “See Results” to go to the next question.



The user’s Lesson 5 results are displayed. The user clicks “Continue” to go to Completion Screen.



The completion screen is displayed. The player will click “Return to Main Menu” to return to the main menu.



The main menu is displayed. The user clicks “Quit” to close the application.

Modifications

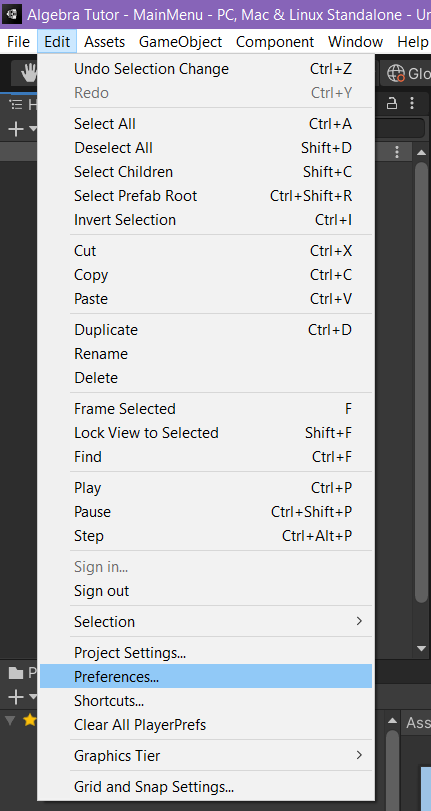
The following sections will give instructions on how to modify Algebra Tutor’s code through the Unity Editor, as well as a list of changes that may improve Algebra Tutor in later versions.

How to modify Algebra Tutor’s code through the Unity Editor (version 2019.4.32f1):

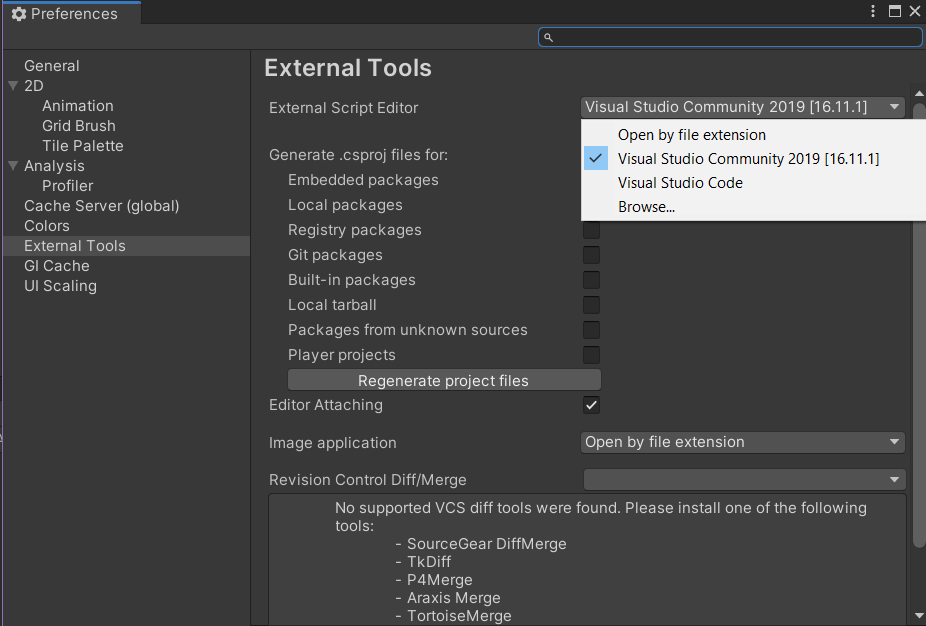
In Unity, the files of C# code used to program the project are called scripts. If you want to edit a script, an IDE that supports C# is heavily recommend, and the IDE that Unity recommends is Visual Studio 2019, which will be the IDE used in the following examples. In addition, the Unity Editor used is version **2019.4.32f1.**

**To change your preferred IDE:**

* Go to Edit > Preferences, then click. This should open the Preferences window



* Go to External Tools > External Script Editor and set the option to your preferred IDE (NOTE: The “Open by file extension” option opens the IDE that is set by the Windows user to open automatically for the extension type of the file requested, which, in the case of C# scripts, is .cs)



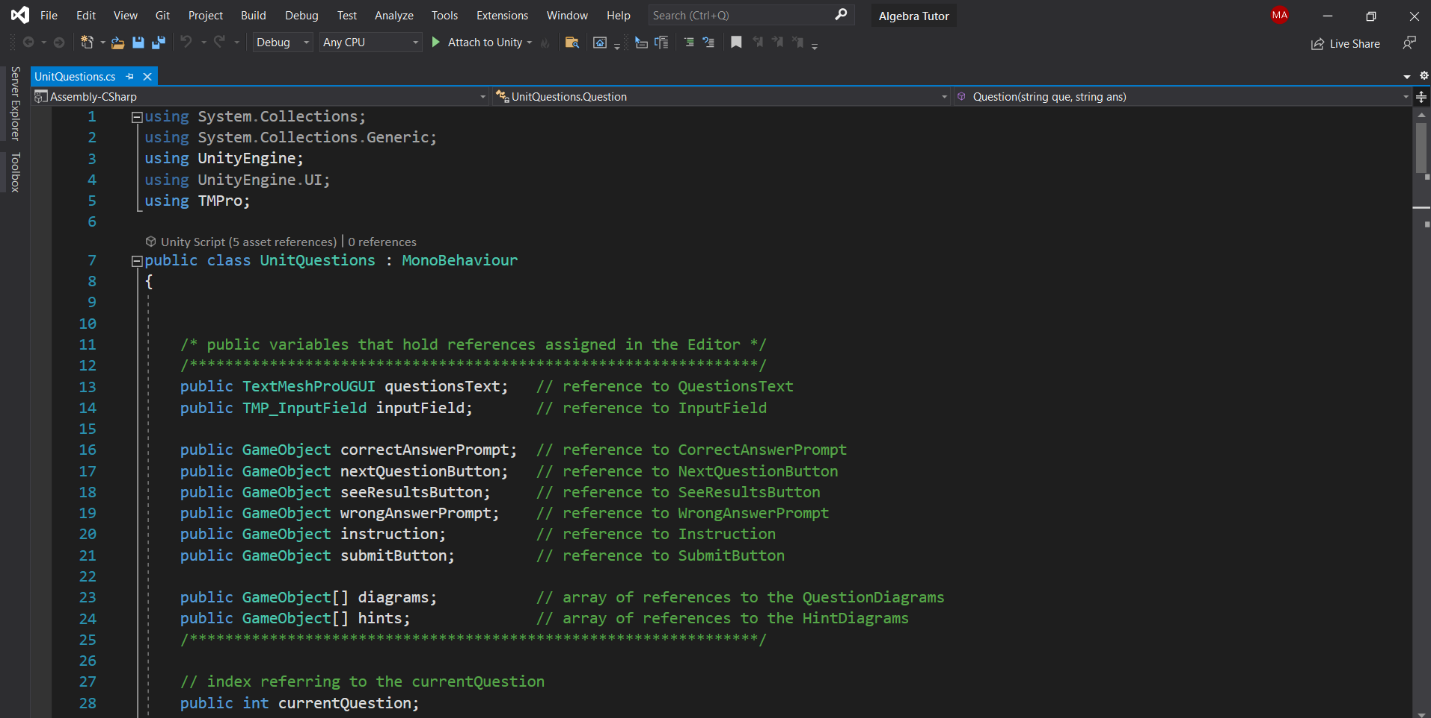
**If you know the name or directory of the C# script you want to modify:**

* In the Project Window, find the script you want to edit in the file hierarchy in the left or search the script’s name using the search bar in the top right.



* Click on the script. Information about the script should show up in the Inspector. Here, click “Open” to launch the script in your preferred IDE

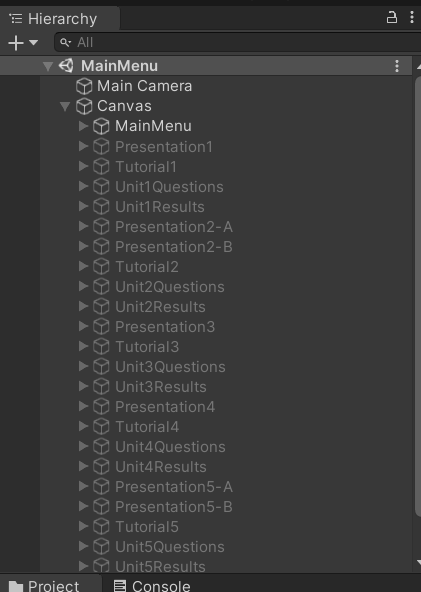




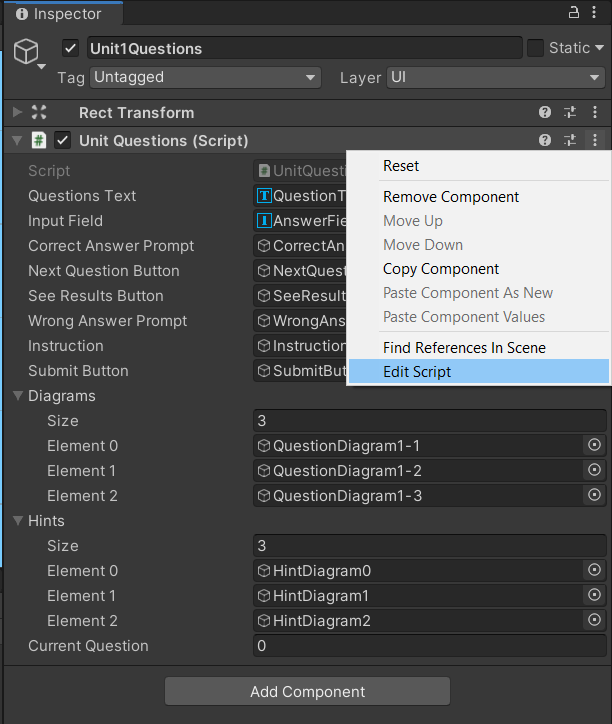
* After making changes to the script, save the changes and close the IDE. Unity should automatically apply the changes to the game.

**If you do not know the name of the C# script you want to modify but you know the objects that use it:**

* In the Hierarchy, find the GameObject(s) that has behavior you want to modify.



* Click on that GameObject. Information about the GameObject should show up in the Inspector. Scripts that are associated with the GameObject will be displayed as components. Find the toolbar of the script you want to edit and click the symbol with the three dots on the far right. Then, click Edit Script to open the script in your preferred IDE.

****

* After making changes to the script, save the changes and close the IDE. Unity should automatically apply the changes to the game.

Changes to consider:

* Version 2
  + Clean up the UI elements
  + Improve readability of text elements.
  + Improve readability of line graphs in the fifth lesson.
  + Improve the screen resolution on the Unity .exe build.
  + Add a quick blurb on the title screen and a overview screen that both tell the user what they are going to learn.
  + Add a menu bar at the bottom of the screen that allows the user to freely return to each lesson once they are unlocked.
* Version 3
  + Make the questions randomized
  + Have 5 questions per lesson
* Version 4
  + Have 10 questions per lesson
  + Add more lessons, for example:
    - Graphing Lines: Point-slope form
    - Graphing Lines: Standard form of linear equations
    - Inequalities
    - Review on square roots
    - Extension on square roots
    - Quadratic equations: Introduction
    - Quadratic Equations: Quadratic formula

Troubleshooting

Below is a table of common problems faced by users as well as their causes behind them and solutions to them.

|  |  |  |
| --- | --- | --- |
| Symptom | Cause | Solution |
| Algebra Tutor.exe cannot be found in File Explorer. | For users:  The Algebra Tutor.zip either has not been received or unzipped.  For developers:  The tutor has not been built or has been built under a different name. | For users:  Make sure you have received Algebra Tutor.zip. Extract the files and make sure Algebra Tutor.exe and Algebra Tutor\_Data are in the same location (See **Installation > User Setup** for more details).  For developers:  Follow the instructions listed in **Installation > Developer Setup** (please note that the instructions are based on Unity version 2019.4.32f1) |
| Algebra Tutor.exe will not launch or shows a blank screen. | Algebra Tutor.exe and Algebra Tutor\_Data are not in the same location. | Find Algebra Tutor\_Data and put it in the same location as Algebra Tutor.exe.  **If Algebra Tutor\_Data cannot be found:**  Users: Obtain a new copy of Algebra Tutor.zip and make sure both Algebra Tutor.exe and Algebra Tutor\_Data are included then extract.  Developers: Rebuild Algebra Tutor in the desired location then make sure both Algebra Tutor.exe and Algebra Tutor\_Data are in that location (see **Installation > Developer Setup** for guidance). |

|  |  |  |
| --- | --- | --- |
| Symptom | Cause | Solution |
| Clicking on a menu option does not display the appropriate screen or causes the game to stall. | If downloading the game’s files from OneDrive:  All contents of Algebra Tutor\_Data have not been downloaded yet.  If the game’s files are natively stored on the PC:  Some contents of Algebra Tutor\_Data are missing. | If downloading the game’s files from OneDrive:  Make sure your PC is connected to the Internet to allow OneDrive to sync the contents on your PC, particularly the contents of Algebra Tutor\_Data and Algebra Tutor.exe. For more information, go to Microsoft’s OneDrive sync troubleshooting page at <https://support.microsoft.com/en-us/office/fix-onedrive-sync-problems-0899b115-05f7-45ec-95b2-e4cc8c4670b2>    If the game’s files are natively stored on the PC:  Users: Obtain a new copy of Algebra Tutor.zip and make sure both Algebra Tutor.exe and Algebra Tutor\_Data are included then extract.  Developers: Rebuild Algebra Tutor in the desired location then make sure both Algebra Tutor.exe and Algebra Tutor\_Data are in that location (see **Installation > Developer Setup** for guidance). |

References

“C# Docs - Get Started, Tutorials, Reference.” *C# Docs - Get Started, Tutorials, Reference. | Microsoft Docs*, Microsoft, 2021, docs.microsoft.com/en-us/dotnet/csharp/.

“Fix OneDrive Sync Problems.” *Office Support*, Microsoft, 2021, support.microsoft.com/en-us/office/fix-onedrive-sync-problems-0899b115-05f7-45ec-95b2-e4cc8c4670b2.

Liu, Shanhong. “Computer Operating Systems Market Share 2012-2021.” *Statista*, Statista, 4 Oct. 2021, www.statista.com/statistics/268237/global-market-share-held-by-operating-systems-since-2009/.

“System Requirements for Unity 2019.4.” *Unity*, Unity Technologies, 3 Dec. 2021, docs.unity3d.com/Manual/system-requirements.html.

“Unity User Manual (2019.4 LTS).” *Unity*, Unity Technologies, 3 Dec. 2021, docs.unity3d.com/Manual/index.html.

Appendix

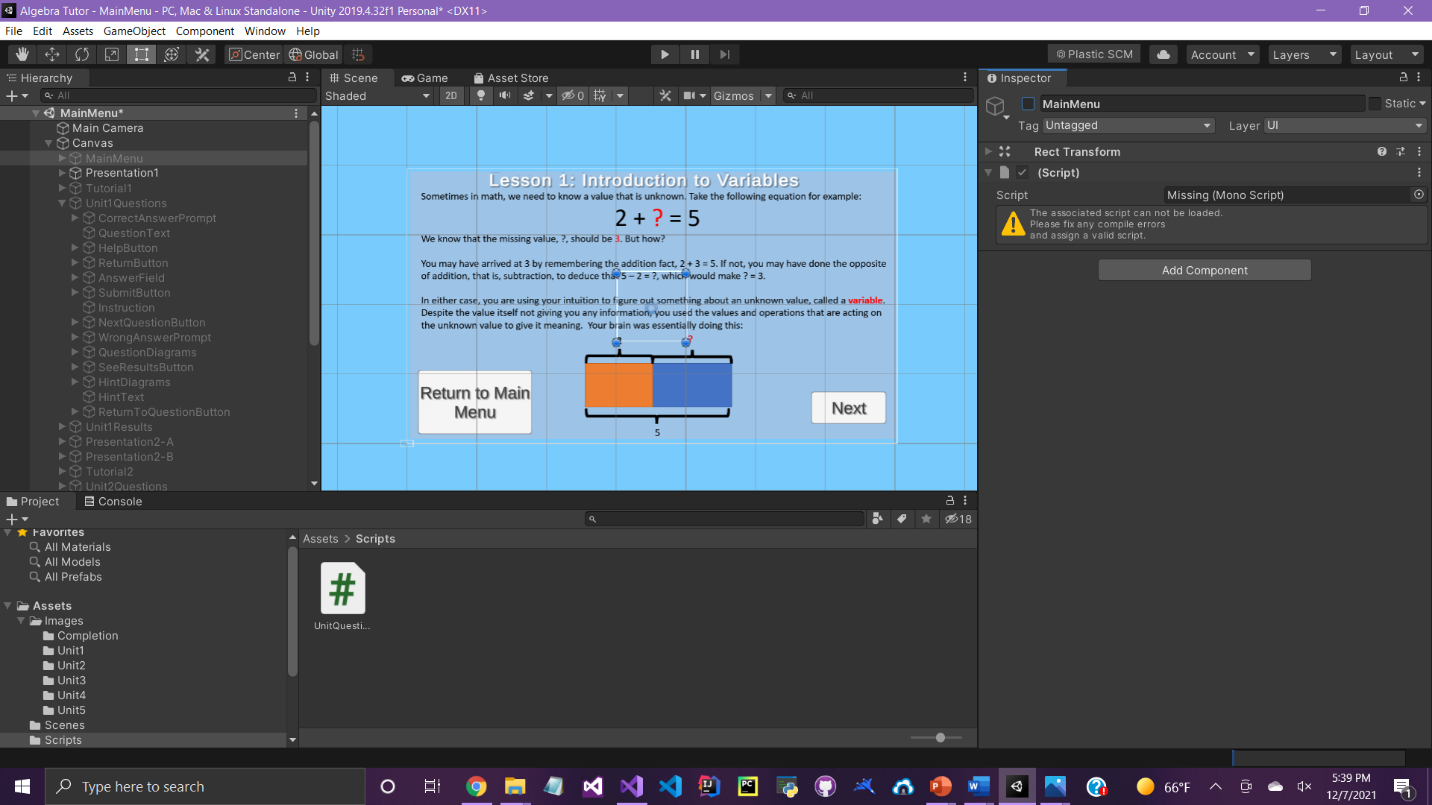
The following is a collection of development screenshots followed by the sole script of the tutor: UnitQuestions.cs.

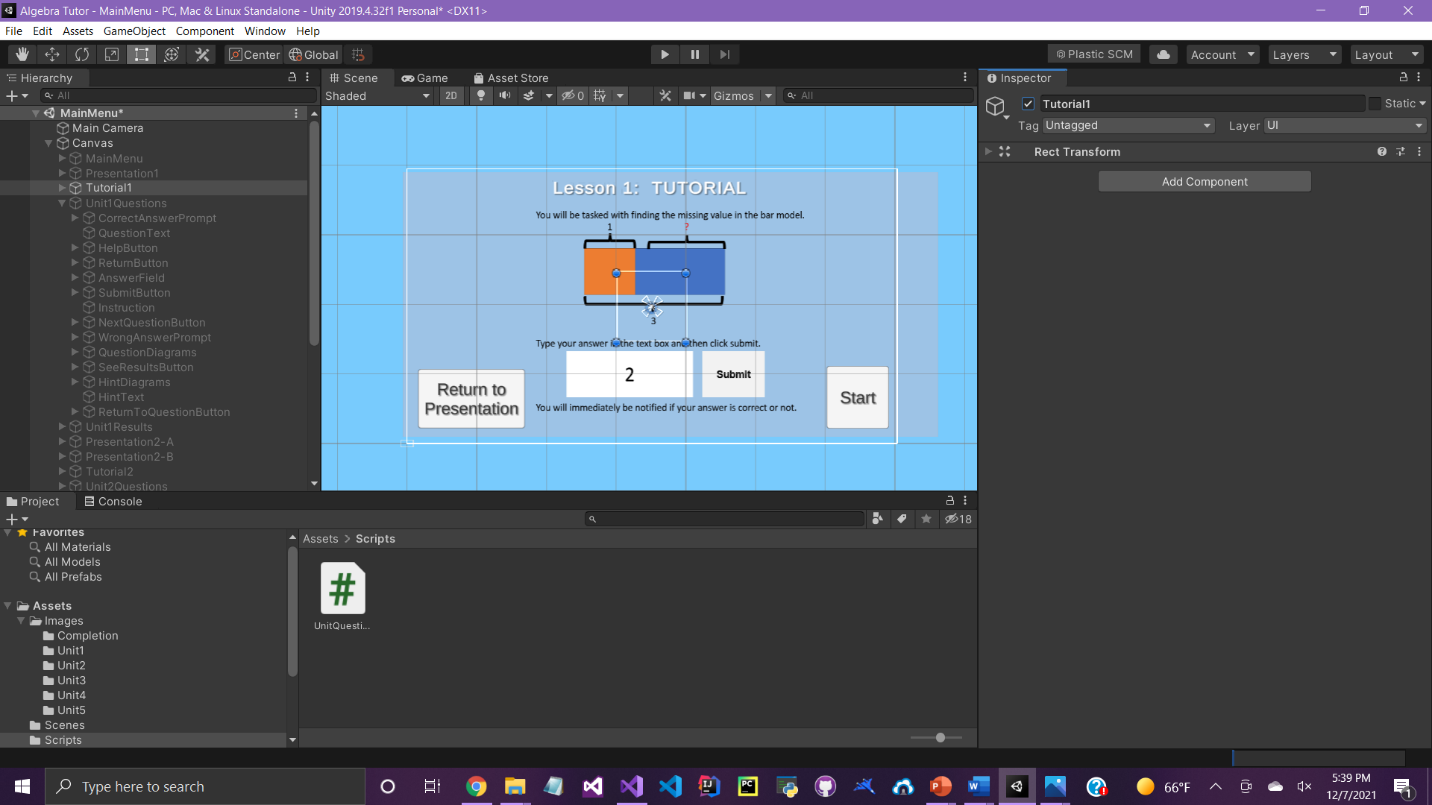
**Development Interfaces**

**Main Menu**

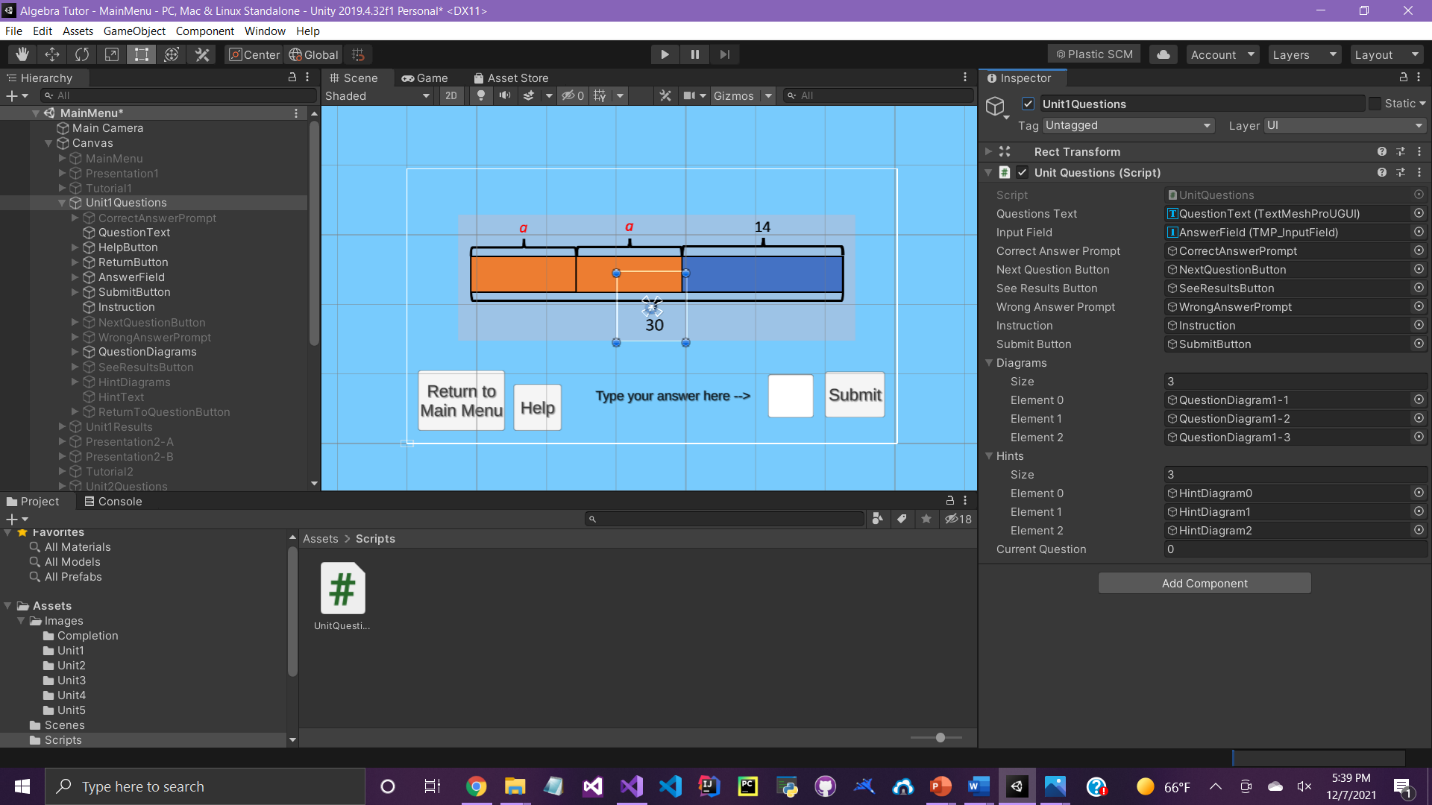


**Lesson 1 Presentation**

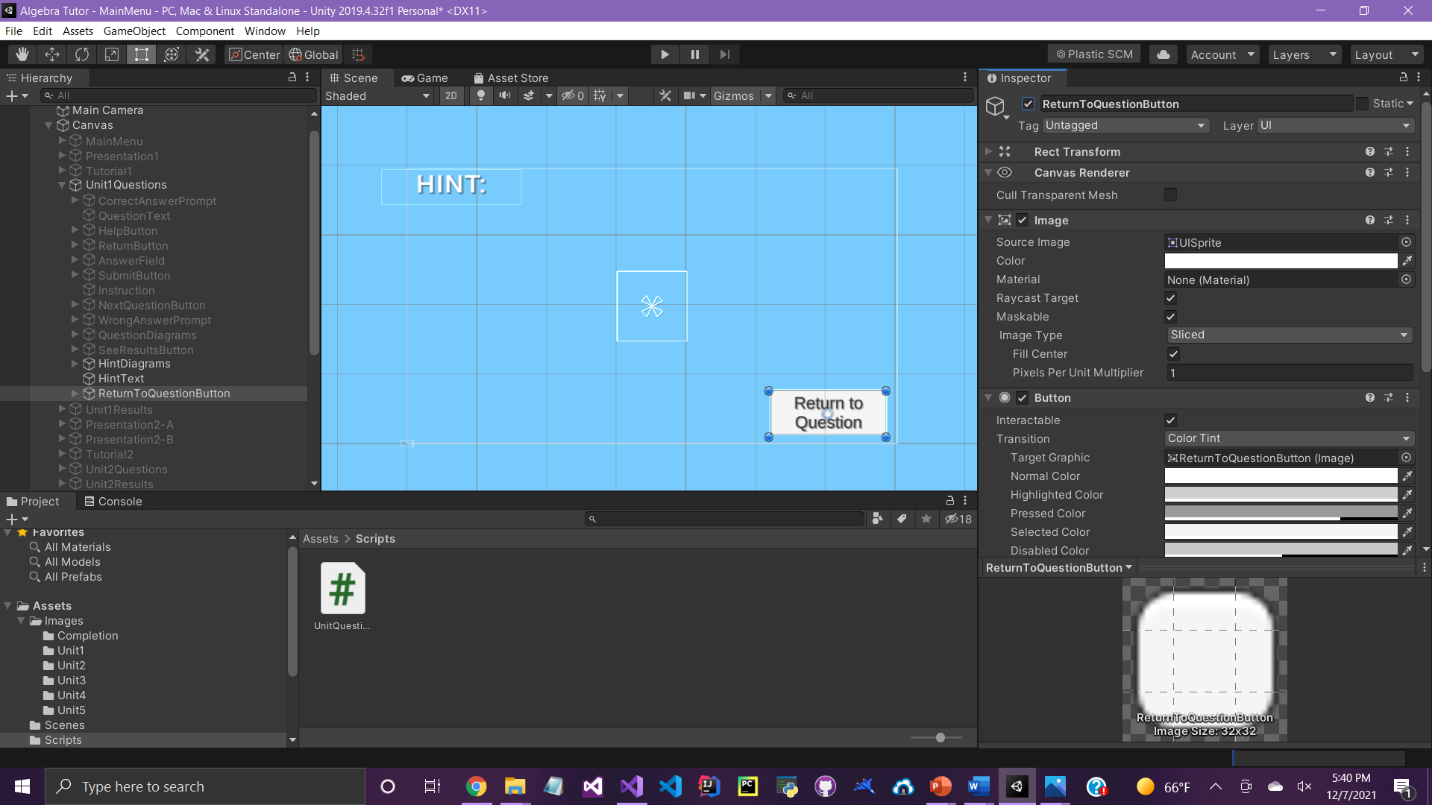
****

**Lesson 1 Tutorial**

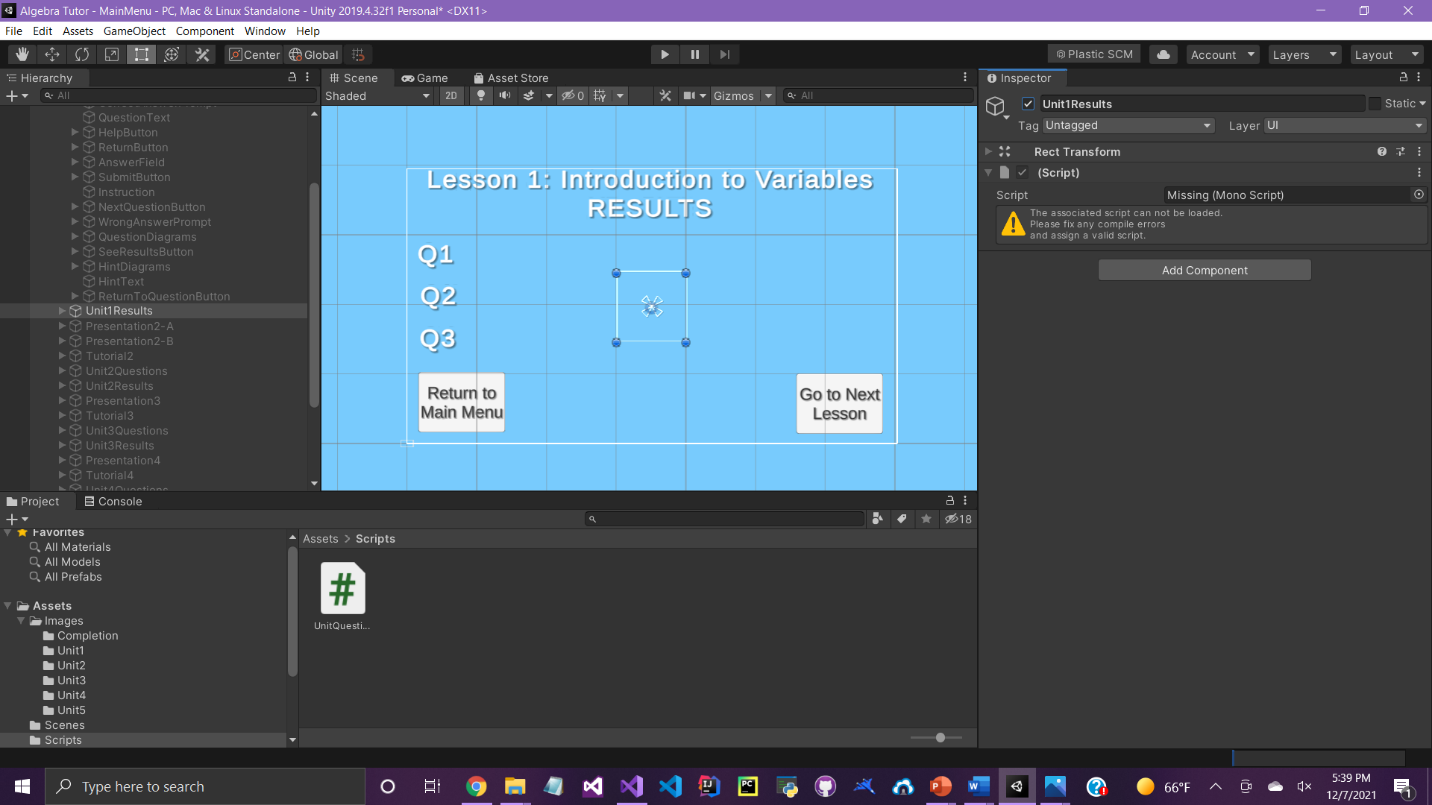
**Lesson 1 Questions**

****

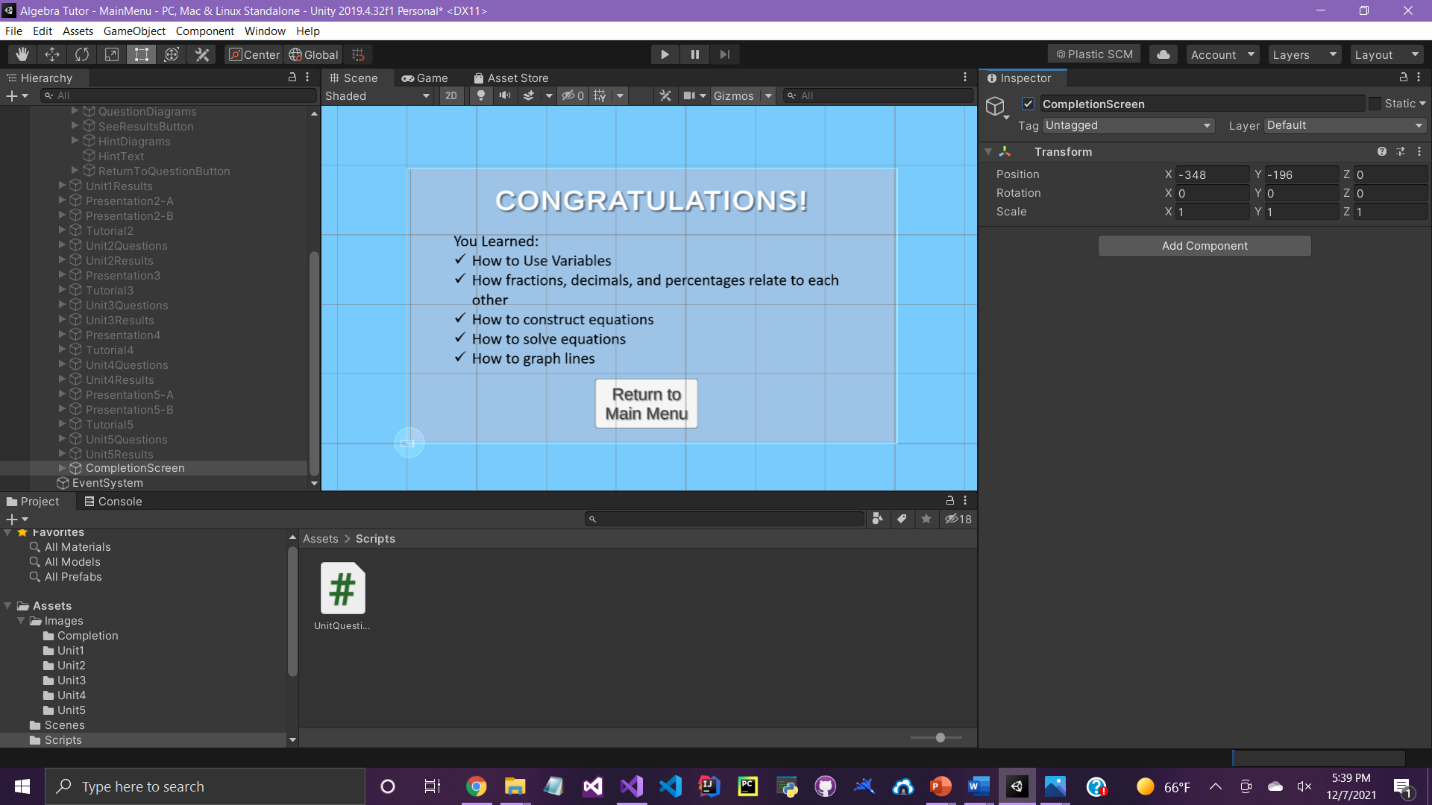
**Lesson 1 Hints**



**Lesson 1 Results**

****

**Completion Screen**

****

**UnitQuestion.cs**

**using System.Collections;**

**using System.Collections.Generic;**

**using UnityEngine;**

**using UnityEngine.UI;**

**using TMPro;**

**public class UnitQuestions : MonoBehaviour**

**{**

**/\* public variables that hold references assigned in the Editor \*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**public TextMeshProUGUI questionsText; // reference to QuestionsText**

**public TMP\_InputField inputField; // reference to InputField**

**public GameObject correctAnswerPrompt; // reference to CorrectAnswerPrompt**

**public GameObject nextQuestionButton; // reference to NextQuestionButton**

**public GameObject seeResultsButton; // reference to SeeResultsButton**

**public GameObject wrongAnswerPrompt; // reference to WrongAnswerPrompt**

**public GameObject instruction; // reference to Instruction**

**public GameObject submitButton; // reference to SubmitButton**

**public GameObject[] diagrams; // array of references to the QuestionDiagrams**

**public GameObject[] hints; // array of references to the HintDiagrams**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**// index referring to the currentQuestion**

**public int currentQuestion;**

**// array holding all of Unit One's questions**

**private Question[] questions = new Question[3];**

**// int variable holding the unit of the gameobject that called UnitQuestions**

**private int unit;**

**/\* Class definition of Question \*/**

**private class Question**

**{**

**// three attributes: the question, its answer, and the number of tries the user took**

**public string question;**

**public string answer;**

**public int tries;**

**// Question constructor**

**public Question(string que, string ans)**

**{**

**question = que;**

**answer = ans;**

**// Start tries at 1**

**tries = 1;**

**}**

**}**

**// Start(): Is called as soon as the first lesson begins**

**void Start()**

**{**

**// Determine the current unit**

**switch (gameObject.name)**

**{**

**case "Unit1Questions":**

**unit = 1;**

**break;**

**case "Unit2Questions":**

**unit = 2;**

**break;**

**case "Unit3Questions":**

**unit = 3;**

**break;**

**case "Unit4Questions":**

**unit = 4;**

**break;**

**case "Unit5Questions":**

**unit = 5;**

**break;**

**default:**

**Debug.Log("Cannot determine the unit");**

**break;**

**}**

**// Check if UnitXQuestions is active in the scene**

**if (gameObject.activeSelf == true)**

**{**

**// Activate all of UnitX's attributes**

**gameObject.transform.Find("QuestionText").gameObject.SetActive(true);**

**gameObject.transform.Find("QuestionDiagrams").gameObject.SetActive(true);**

**gameObject.transform.Find("HelpButton").gameObject.SetActive(true);**

**gameObject.transform.Find("ReturnButton").gameObject.SetActive(true);**

**gameObject.transform.Find("AnswerField").gameObject.SetActive(true);**

**gameObject.transform.Find("SubmitButton").gameObject.SetActive(true);**

**gameObject.transform.Find("Instruction").gameObject.SetActive(true);**

**gameObject.transform.Find("HintDiagrams").gameObject.SetActive(true);**

**}**

**else**

**{**

**Debug.Log("UnitXQuestions not active in the scene");**

**}**

**// Deactivate WrongAnswerPrompt**

**wrongAnswerPrompt.SetActive(false);**

**// Fill questions[] with the (question, answer) pairs**

**switch (unit)**

**{**

**case 1:**

**questions[0] = new Question("What is the value of a?", "8");**

**questions[1] = new Question("What is the value of b?", "15");**

**questions[2] = new Question("What is the value of x?", "11");**

**break;**

**case 2:**

**questions[0] = new Question("Enter the missing percentage.", "25");**

**questions[1] = new Question("Enter the missing fraction, in its simplest form", "3/4");**

**questions[2] = new Question("Enter the missing decimal", "0.04");**

**break;**

**case 3:**

**questions[0] = new Question("Complete the equation.", "5");**

**questions[1] = new Question("Complete the equation.", "7");**

**questions[2] = new Question("Complete the equation.", "2");**

**break;**

**case 4:**

**questions[0] = new Question("Solve for I.", "3");**

**questions[1] = new Question("Solve for C.", "2");**

**questions[2] = new Question("Solve for I.", "6");**

**break;**

**case 5:**

**questions[0] = new Question("What is the slope, m, of the following line?", "7");**

**questions[1] = new Question("What is the y-intercept, b, of the following line?", "5");**

**questions[2] = new Question("What is the slope of a line with the following two points?", "3");**

**break;**

**default:**

**Debug.Log("Cannot load questions");**

**break;**

**}**

**// Set the current question to the first one**

**currentQuestion = 0;**

**// Call DisplayQuestionText()**

**DisplayQuestionText();**

**}**

**// DisplayQuestionText(): displays the question text at the top of the screen**

**void DisplayQuestionText()**

**{**

**// Set the question text to "QX: question", where X is 1, 2, or 3**

**questionsText.text = "Q" + (currentQuestion + 1) + ": " + (questions[currentQuestion].question);**

**}**

**// OnClickHelp(): determines what hint to display when HelpButton is clicked**

**public void OnClickHelp()**

**{**

**// disable question elements**

**gameObject.transform.Find("QuestionText").gameObject.SetActive(false);**

**gameObject.transform.Find("QuestionDiagrams").gameObject.SetActive(false);**

**gameObject.transform.Find("HelpButton").gameObject.SetActive(false);**

**gameObject.transform.Find("ReturnButton").gameObject.SetActive(false);**

**gameObject.transform.Find("AnswerField").gameObject.SetActive(false);**

**gameObject.transform.Find("SubmitButton").gameObject.SetActive(false);**

**gameObject.transform.Find("Instruction").gameObject.SetActive(false);**

**//Display the appropriate hint image, text, and button**

**hints[currentQuestion].SetActive(true);**

**gameObject.transform.Find("HintText").gameObject.SetActive(true);**

**gameObject.transform.Find("ReturnToQuestionButton").gameObject.SetActive(true);**

**}**

**// OnReturnToQuestion(): bring the user back to the current question**

**public void OnReturnToQuestion()**

**{**

**// enable question elements**

**gameObject.transform.Find("QuestionText").gameObject.SetActive(true);**

**gameObject.transform.Find("QuestionDiagrams").gameObject.SetActive(true);**

**gameObject.transform.Find("HelpButton").gameObject.SetActive(true);**

**gameObject.transform.Find("ReturnButton").gameObject.SetActive(true);**

**gameObject.transform.Find("AnswerField").gameObject.SetActive(true);**

**gameObject.transform.Find("SubmitButton").gameObject.SetActive(true);**

**gameObject.transform.Find("Instruction").gameObject.SetActive(true);**

**// disable the appropriate hint image, text, and button**

**hints[currentQuestion].SetActive(false);**

**gameObject.transform.Find("HintText").gameObject.SetActive(false);**

**gameObject.transform.Find("ReturnToQuestionButton").gameObject.SetActive(false);**

**}**

**// OnSubmit(): determines which action to take when the user clicks SubmitButton**

**public void OnSubmit()**

**{**

**// If the user's answer is correct**

**if (inputField.text.Equals(questions[currentQuestion].answer))**

**{**

**// If \*not\* on the final question**

**if (currentQuestion < (questions.Length - 1))**

**{**

**// Deactivate WrongAnswerPrompt, SubmitButton, and Instruction**

**wrongAnswerPrompt.SetActive(false);**

**submitButton.SetActive(false);**

**instruction.SetActive(false);**

**// Activate CorrectAnswerPrompt and NextQuestionButton**

**correctAnswerPrompt.SetActive(true);**

**nextQuestionButton.SetActive(true);**

**}**

**// If on the final question**

**else**

**{**

**// Deactivate WrongAnswerPrompt, SubmitButton, and Instruction**

**wrongAnswerPrompt.SetActive(false);**

**submitButton.SetActive(false);**

**instruction.SetActive(false);**

**// Activate CorrectAnswerPrompt and SeeResultsButton**

**correctAnswerPrompt.SetActive(true);**

**seeResultsButton.SetActive(true);**

**}**

**}**

**// If the user's answer is \*not\* correct**

**else**

**{**

**// Activate WrongAnswerPrompt**

**wrongAnswerPrompt.SetActive(true);**

**// Increment the number of tries**

**questions[currentQuestion].tries++;**

**}**

**}**

**// OnContinue(): determines what to do when the user clicks NextQuestionButton**

**public void OnContinue()**

**{**

**// Verify that the user is not on the final question**

**if (currentQuestion < (questions.Length-1))**

**{**

**// Deactivate CorrectAnswerPrompt and NextQuestionButton**

**correctAnswerPrompt.SetActive(false);**

**nextQuestionButton.SetActive(false);**

**// Activate SubmitButton and Instruction**

**submitButton.SetActive(true);**

**instruction.SetActive(true);**

**// Update the diagram to the next question**

**diagrams[currentQuestion].SetActive(false);**

**currentQuestion++;**

**diagrams[currentQuestion].SetActive(true);**

**// Clear InputField**

**inputField.text = "";**

**// Call DisplayQuestionText**

**DisplayQuestionText();**

**}**

**// Otherwise, log an error in the Console**

**else**

**{**

**Debug.Log("NextQuestionButton clicked on the final question");**

**}**

**}**

**// ViewResults: View the user's results upon clicking SeeResultsButton**

**public void ViewResults()**

**{**

**GameObject results = GameObject.Find("Unit"+unit+"Results");**

**// Check if is active in the scene**

**if (results.activeSelf == true)**

**{**

**// Get references to QXResults ( X = {1, 2, 3} )**

**TextMeshProUGUI q1Results = results.transform.Find("Q1Results").gameObject.GetComponent<TextMeshProUGUI>();**

**TextMeshProUGUI q2Results = results.transform.Find("Q2Results").gameObject.GetComponent<TextMeshProUGUI>();**

**TextMeshProUGUI q3Results = results.transform.Find("Q3Results").gameObject.GetComponent<TextMeshProUGUI>();**

**// Display the number of attempts it took the user to answer correctly**

**q1Results.text = "Q1: " + questions[0].tries + " attempt(s)";**

**q2Results.text = "Q2: " + questions[1].tries + " attempt(s)";**

**q3Results.text = "Q3: " + questions[2].tries + " attempt(s)";**

**}**

**else**

**{**

**Debug.Log("UnitXResults is not active in the scene");**

**}**

**}**

**}**