Yun, Chris

Comp 465 – Mon/Wed 7pm

Professor Karamian

Project 2 – Lego Blocks Extended

Contents

[Introduction 1](#_Toc27404535)

[Design Goals and Objectives 2](#_Toc27404536)

[System Behavior 2](#_Toc27404537)

[Menu Scene. 2](#_Toc27404538)

[Game Scene. 2](#_Toc27404539)

[Build Scene. 2](#_Toc27404540)

[Logical View 5](#_Toc27404541)

[User Interaction 7](#_Toc27404542)

[User Interface 8](#_Toc27404543)

[Screenshots 8](#_Toc27404544)

[Lessons Learned 11](#_Toc27404545)

# Introduction

This document is a walkthrough of the design, goals, and objectives of the Lego Blocks Project. The project a mix of 3d Tetris and Legos. A board is generated at the start of the game, and blocks with random sizes and colors are generated above it. The blocks can be moved and rotated along with the camera.

Also added is the ability to place the blocks manually in the build mode. Blocks can be chosen with various sizes and colors. The board is not generated, but placed before hand with the scale of the Y-axis shrunken down to 0.08 to match the size of the lego pieces.

# Design Goals and Objectives

The goals were to create an experience that is user friendly and enjoyable. I wanted to make sure there wouldn’t be any confusion on how to operate the program on the user’s perspective. The goals for the designer was to focus on the basic tasks and work my way up to more difficult tasks that required those basics to be put in place first. Some problems had multiple solutions. These would be worked on with future scripts and design choices in mind.

The priority for designing the project was the first work on basic structures and see if they can be worked with. The canvas I see as a separate structure that can be worked on in a vacuum. I knew I wanted an overlay interface that can allow and guide the user to change the block type and color, so I worked on that first as it was simple and doesn’t affect the rest of the project in a large way.

Then the base code using the cube primitive was used and altered to work with lego blocks. There were no specific objectives, but to just work on it piece by piece until it was functional.

# System Behavior

There are two scenes that have different options and controls, both stated plainly for the user to manipulate.

## Menu Scene.

This scene toggles between two panels: the menu panel and the setup panel. When the program is loaded, the menu scene comes up first. There are three buttons. First the enter the game scene, second to toggle the setup panel, and third to exit the program. The setup panel is there to allow the user to change the board size, the difficulty level, and the toggle on the background music.

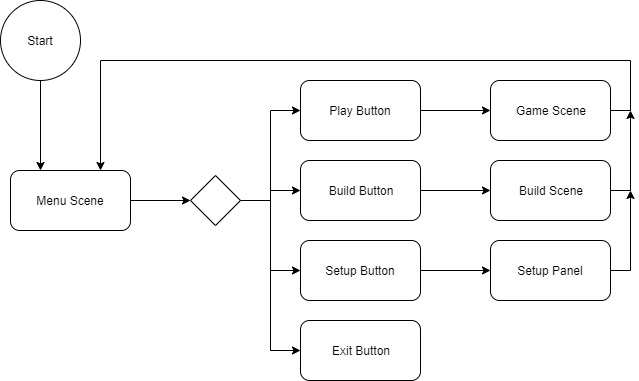
## Game Scene.

This scene is where the user experiences the game. The board is dynamically allocated and falling blocks are controlled by the user along with the camera and options to return to the menu or pause the game.

## Build Scene.

This scene allows the user to manually place lego blocks of varying sizes and colors on a flat lego board. The camera options are the same as in the Game Scene. Left clicking will allow the user to place the block wherever he chooses, while the right mouse click lets the user destroy any block. The “A” button lets the user undo however many moves, and the “S” button lets the user redo those undo presses for greater control over the build scene.

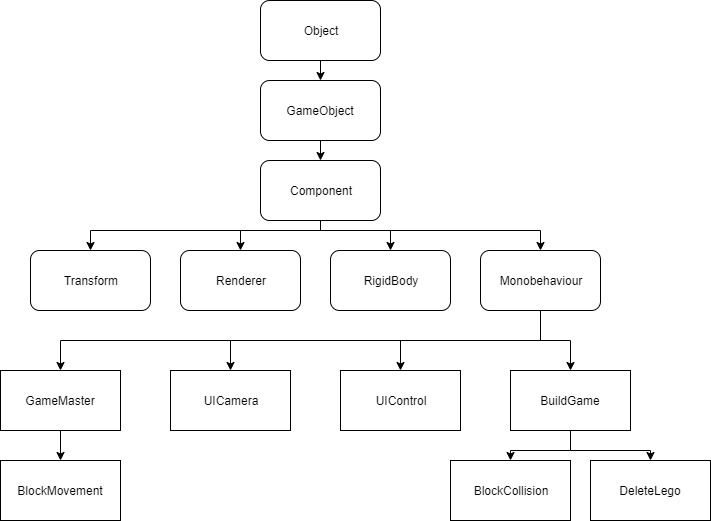
The following is the updated flow chart of how navigation works within project 2. Simply, a button was added in the Menu Scene that redirects the user to the Build Scene. From there, pressing “Q” will bring the user back to the Menu Scene.



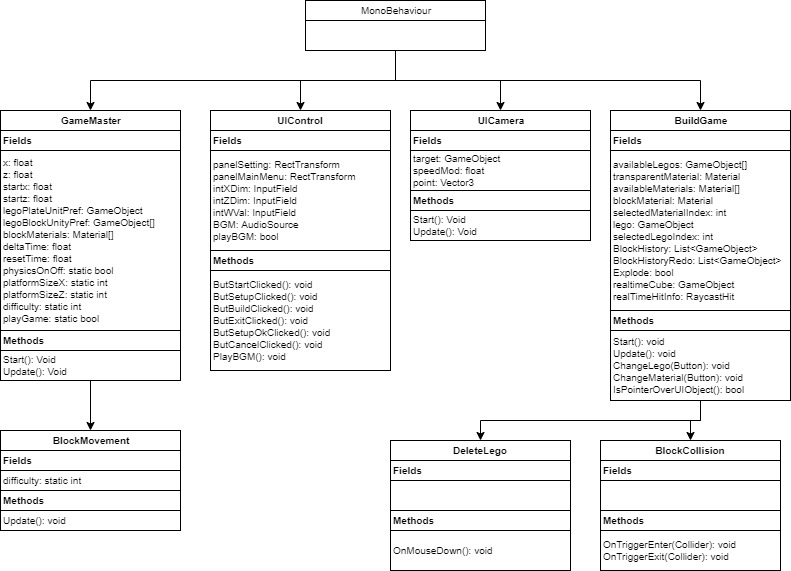
# Logical View

The logical view describes the main functional components of the system. The following is the Unity class hierarchy along with the scripts that I have created under MonoBehavior for my project.

For project 2, I have added in the BuildGame class in order to act as the “GameMaster” for the Build Scene. It controls almost everything in the scene with the help of BlockCollision and DeleteLego, which I used for the delete function.



The following is the updated UML for project 2. A function was put into the old UIControl to link the Main Scene with the new Build Scene. Other than that, it shows the connections for the BuildGame class.



# User Interaction

1. Input Fields
   1. X Input: This input determines how far the board goes along the X-axis. The default is 25.
   2. Z Input: This input determines how far the board goes along the Z-axis. The default is 25.
   3. Difficulty: This input will alter the speed of the falling blocks. 1 is the easiest difficulty with the blocks falling for approximately 5 seconds. 2 is the medium difficulty where the blocks fall for 2.5 seconds, and 3 is the hardest difficulty where the blocks fall for 1 second. The default setting is 1.
2. Clickable Buttons
   1. Play Game: This loads the Game Scene.
   2. Settings: This disables the Menu Panel and enables the Settings Panel.
   3. Exit: This terminates the program.
   4. BGM: This toggles the music.
3. Block Controls
   1. W: Moves the block forward.
   2. A: Moves the block to the left.
   3. S: Moves the block backwards.
   4. D: Moves the block to the right.
   5. R: Rotates the block clockwise in a 90 degree angle.
4. Camera Controls
   1. LeftArrow: Rotates the camera clockwise.
   2. RightArrow: Rotates the camera counter-clockwise.
   3. UpArrow: Raises the camera along the Y-axis.
   4. DownArrow: Lowers the camera along the Y-axis.
   5. Z: Camera zooms in.
   6. X: Camera zooms out.
5. Misc. Controls
   1. Q: Quits the game and loads the Menu Scene.
   2. P: Pauses the game.
6. Build Controls
   1. A: Undo removes the last placed block.
   2. S: Redo places the blocks removed by the Undo feature.
   3. LeftClick: Places a lego onto the board or other lego objects.
   4. RightClick: Deletes the clicked on lego object.
   5. Camera and Misc controls remain the same.

# User Interface

1. Menu Scene.

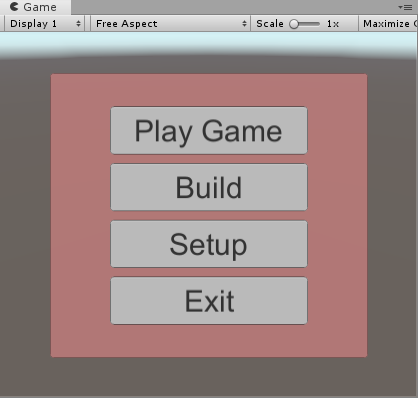
The Menu’s main function is to access different parts of the program. The first button loads the Game Scene and the Setup button loads in the Setup panel. The purpose is to give the user the option to play the game with the default settings or to alter those settings in very simple and straightforward ways.

1. Game Scene.

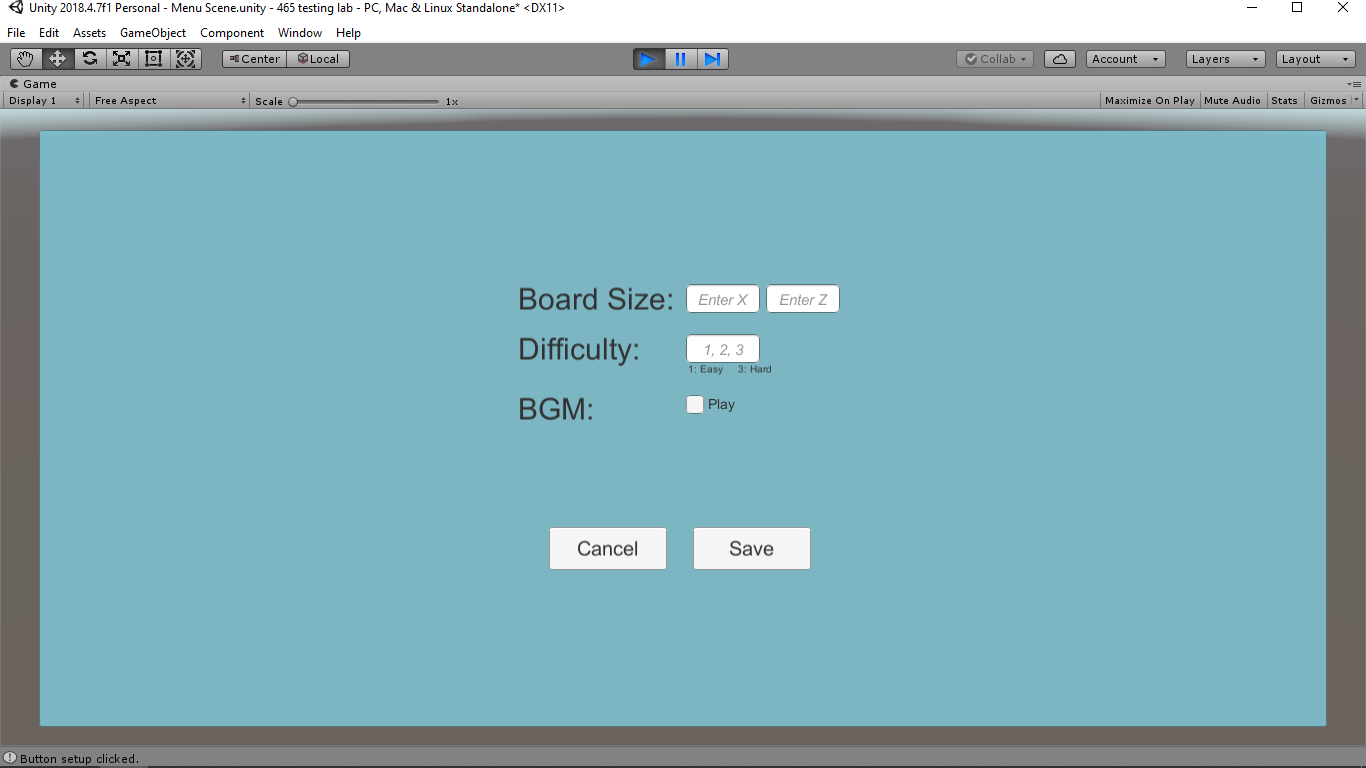
The game scene’s interface is there to show the user all of the controls for the game. On the left are the block and misc. controls, and the right side has all of the camera movements.

# Screenshots

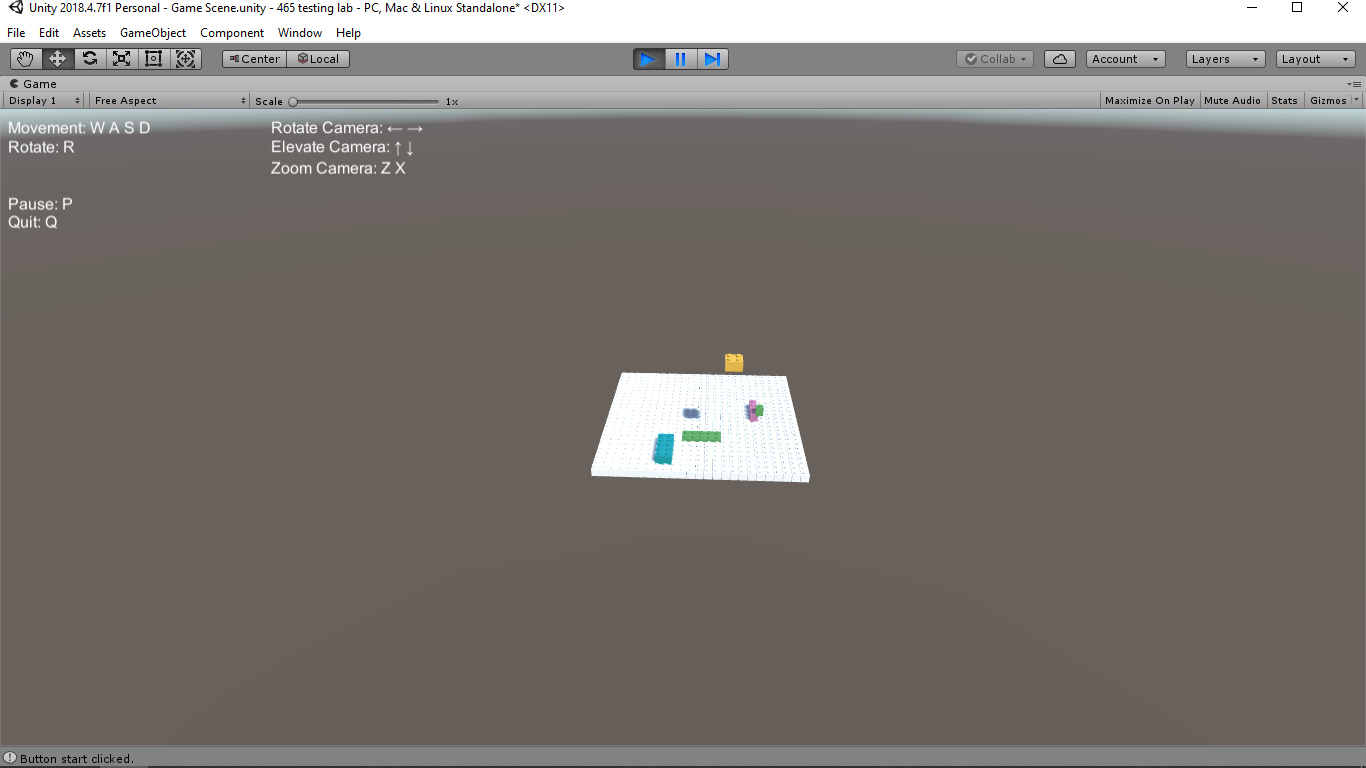
1. The first scene for the build is the Menu Scene



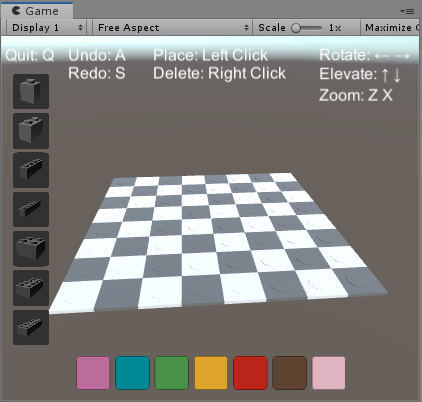
1. Clicking on the Setup button hides the menu panel and activates the setup panel.



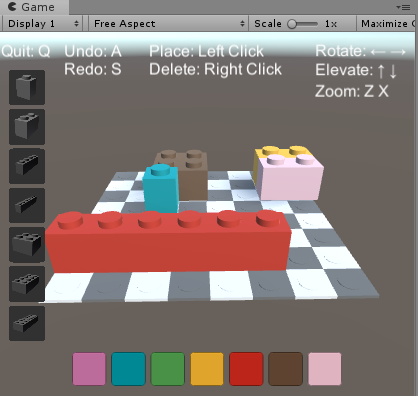
1. Clicking on the Play Game button loads the Game Scene.



1. Clicking the Build button loads in the build scene.



1. Blocks can be placed and deleted.



# Lessons Learned

The Basics

Learning the basics of creating objects, attaching scripts, and rotating them has been the building blocks for my learning experience. Everything is built upon the basics, and there isn’t much progress to be done from skipping any step. The first hurdle was learning about prefabs and how to apply them to the project. After figuring out that each game block could be created from a single cube and cylinder, creating the rest of the prefabs became really simple. Another problem that I had was figuring out how to create the scripts and figuring out which game objects to attach them to. Looking up many examples and tutorials on google was really helpful in learning the flow of creating scripts and figuring out which objects they belong to.

Panels

The 2d canvas and the panels were much easier to envision and create. They just seemed to have less variety in what they can do, so I was able to figure our the layouts, resizing and such with more ease. The scripting however took longer to figure out. Even the code for PlayerPrefs, which was half given to us, was a bit confusing at first. I didn’t really know what to do with them, but a couple of tutorials and examples later, I figured they worked as a sort of global variable and I was able to get a much better grasp of the concept.

Debugging

As usual for anything related to coding, debugging was the biggest hurdle. An aspect that helped throughout the project was Debug.Log(). Similar to using System.out in Java programs, it helped me figure out where I was in the Unity programs. It also helped in figuring out if the buttons were connected to the proper methods in the proper scripts, and if a click or button press was going to the right places in the code. The most difficult part was getting a grasp of the syntax. Since I’ve never used C#, it was sometimes a chore figuring out simple things like If I had to use GameObject or gameObject, and things like where to put brackets, triangle brackets, or parentheses.

Project 2

For the second project, I feel like I’ve gotten a better grasp of how Unity works in terms of connecting scripts to objects and how they can interact with each other and with key presses. The frustrating part of this project was constantly introducing new bugs. When I fix something, a different problem arises.

The biggest hurdle was getting the normal right for each different lego piece. When using the primitive cube, it was really easy to set normal to just that object and there weren’t any issues with accessing the components of that primitive. When using a lego block, the big problem first is that it is an empty object(lego 1x2) that has more empty objects as children(brick/plate), that also have their own children(cube/cylinder). I had a lot of trouble accessing different components from each of objects that were nested into each other. The other big issue is setting the normal to be free to change between the different dimensions of each lego piece. When using a single piece, there were not many bugs involving this, but when using 7 different pieces as I have decided to do, the pieces don’t quite match up with each other, and I was never able to solve this problem.

On the bright side, I feel like I’ve gotten much better at working with the canvas and panels. 2d is just so much easier for me. I’ve also gotten a lot better at checking my work frequently. I would just change one line of code and see the effects of it before moving on. At the earlier stages of working on this project, I would change multiple lines, get an unexpected results, and then not know where I went wrong.

Conclusion

Learning about Unity gave me a real appreciation for anything 3d and game related. The physics did really interesting things depending on their settings, and the way Unity allows for scripts to translate into controls for these physical objects made me appreciate how difficult things were made simple through Unity.

After finishing this second project, I really appreciate how relatively easy it is to use Unity. When working on OpenGL, I couldn’t even get the program to start. Working on Unity, I can see the effects of the changes I make very quickly. I love getting that kind of response when coding, rather than getting a wall of compiling errors to fix.