



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Cognitive Game Using Unity

K Vinay Kumar^{*1}, Cherala Saiteja^{*2}, M Anjali^{*3}, K Ranjith Reddy^{*4}, Dr.G.Suwarna Gothane^{*5}

^{*1}JNTUH, Computer Science and Engineering, CMR Technical Campus, Medchal, Telangana, India

^{*2}JNTUH, Computer Science and Engineering, CMR Technical Campus, Medchal, Telangana, India

^{*3}JNTUH, Computer Science and Engineering, CMR Technical Campus, Medchal, Telangana, India

^{*4}Assistant Professor, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Telangana, India

^{*5}Associate professor, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Telangana, India

ABSTRACT

Cognition refers to "the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses". It encompasses many aspects of intellectual functions and processes such as: perception, attention, the formation of knowledge, memory and working memory, judgment and evaluation, reasoning and "computation", problem solving and decision making, comprehension and production of language. Game development on the other hand is the process of developing a video game. The paper aims at developing a Cognitive game using unity game engine to test whether it is a good choice for 2D game development. We particularly aim to work on three aspects of cognition. Language, reflexes and short term memory.

Keywords : Cognition, Unity, memory, reflex, language.

I. INTRODUCTION

Today, mobile games consume a vast market share within the games industry, it is expected that in 2018 mobile gaming will account for 43 percent of the gaming market revenue and currently there are over a third of Americans playing mobile games daily. In regards to genre; the most popular games are brain puzzle games (with over 37 million users per month playing one), closely followed by matching puzzle games. According to the UKIE's games industry map, there are nearly a thousand games companies in London. Out of these, roughly seven hundred are working on the mobile platform and over two hundred were formed in the last five years. Not all of these companies will be run by younger developers but it is safe to assume that at least a small portion of them are. Considering that nearly two billion mobile devices are running a Unity-made game, it is probable that quite a large majority of these new companies will be using Unity to develop mobile games.

This paper aims at developing a two dimensional(2D) game using unity. We are particularly interested in 2D games because Unity has its name for 2D game development. We are going to learn about everything that is required to develop a 2D game in unity.

II. LITERATURE SURVEY

There are other game engines other than unity in the market. Some of them are unreal engine, game maker studio, Godot, Amazon Lumberyard and Cryengine.

Unreal Engine is widely regarded as the best game engine overall mostly due to the graphics it is able to offer and the wide range of customization options available. You can build visually-stunning, ground-breaking games in Unreal Engine. Some of the more popular titles that have been built in Unreal Engine are: *Fortnite*, the *Borderlands* series, *Rocket League*, the *Gears of War* series, *Bioshock*, etc.

Game maker studio is an incredibly easy game engine to learn that doesn't require prior programming experience. But the easy nature of a game maker makes it limited in the types of games you can make. It does not support 3D game development. That is really a major drawback. Some examples of games that have been created with GameMaker Studio are: *Spelunky*, *Undertale*, *Hyper Light Drifter*, and *Hotline Miami*.

Godot is another game engine if we want to build 2D style games. It even supports 3D. Godot hasn't been around as long as some of the other engines on this list and there really haven't been any super successful games made with the engine.

CryEngine and **Lumberyard** are like unreal engines. High fidelity graphics and more customizable compared to others. Lumberyard is based on CryEngine, amazon bought cryengine source code and developed lumberyard.

III. PROPOSED SYSTEM

In this section we explain why unity is best for all kinds of developers when it comes to developing games for mobile and low end devices.

While Unity and Unreal Engine are often considered the top two game engines, both engines serve different purposes. Whereas Unreal Engine is best-suited for more robust games. Especially from a graphics standpoint Unity is more versatile and can be a better option for developers who are looking to build mobile games, 2D games.



Fig: Unity

Unity is in the top 3 best game engines. As you can see in the above figure Unity owns 45 percent of the global market share, this is really huge compared to other engines. It has offices all around the globe, 3.3 billion developers using unity, 600 million games and much more.

The most important feature that separates unity from other engines is its supported platforms. Unity supports 17 platforms. It can run on every major Operating system.

Lets see what 2D tools unity has that makes it great. The first thing you do in 2D game development is importing sprites into the game engine. Unity has its own sprite technology that makes working with sprites easier. Sprite creator, sprite editor, sprite renderer and sprite packer are some 2D tools unity uses to make 2D workflow easier.

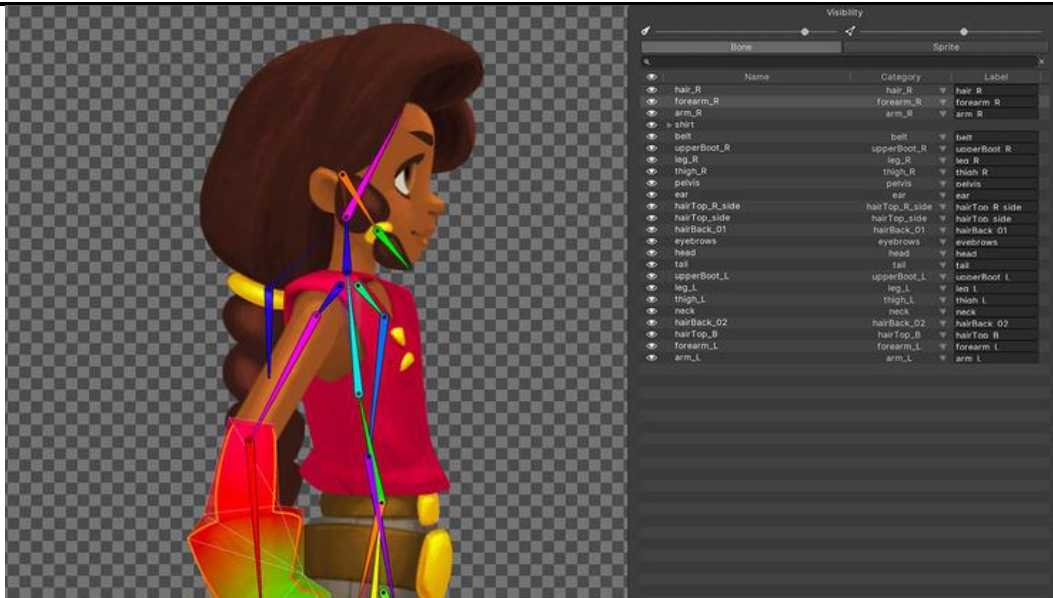


Fig: Unity Sprite Editor

Unity has visual scripting(BOLT) for non programmers. It helps team members create scripting logic with visual, drag and drop graphs instead of writing code. It also enables more seamless collaboration between programmers, artists and designers for faster programming.

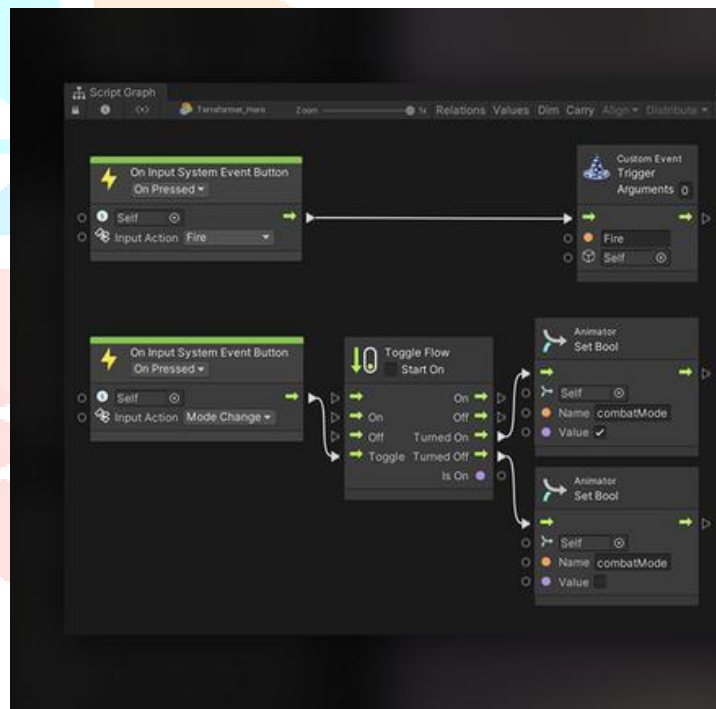


Fig: Unity Visual Scripting (BOLT)

IV. IMPLEMENTATION AND PROPOSED WORK

Assets:

Unity has its own asset store where developers can share their assets with others. It is a great place for beginners who don't want to spend time creating assets when they should be working on other parts of the game.

We downloaded a sprite UI pack from unity asset store. It has button sprites, arrows, circle sprites..etc, everything we require to create a game.

Layout and visuals:

Unity has an ECS system. Entity component system is a data oriented way of designing components. With one container we can customize it by adding only the tools that we need. It reduces usage of the extra tools.

So coming to the layouts Unity has different types of builtin layout types such as grid, horizontal layout group, vertical layout group etc. we are developing a cognitive game which requires only some of the UI tools. The games we are going to create are word type games, a game where it asks you to enter a word starting by a letter given. A reflex game where you need to press the boxes that glows and a game where you need to remember the sequence shown on the screen and select the same sequence.

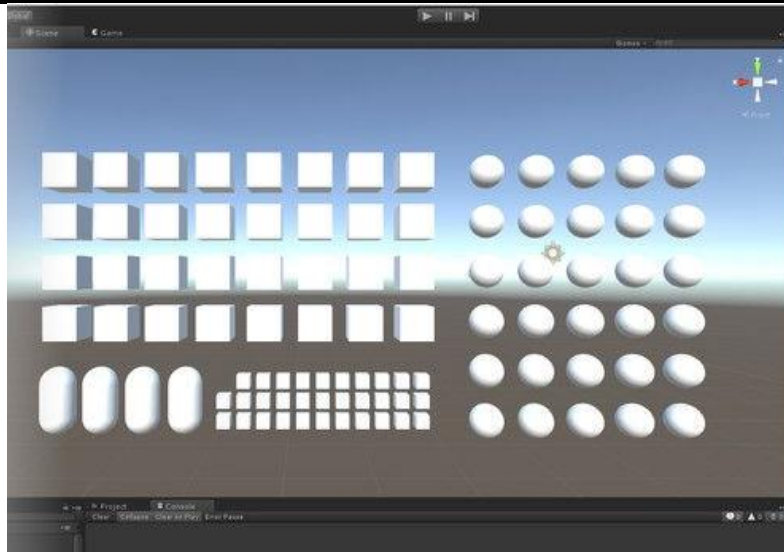


Fig: layout groups

So we need layouts for setting up the screen. We are using vertical and horizontal layout groups to make a responsive Ui that can adjust based on the screen size so that it makes it easy to convert to other platforms with minimal work.

Scripting:

As we discussed unity has visual scripting. Although it makes programming easy, it limits the functionality. We will use traditional scripting.

We need to program the functionality for the layout that we created. We are using a visual studio for programming.

Example logic for word game: we need to give a random letter to the screen so the player sees it. Then we need to take input from the user, then check whether the word entered by the user starts with the given letter and is there a word that is given by the user. Example test cases:

Letter H	User input: Hello	Result: the word hello is starting with h and is a real word
Letter K	User input: Kangaroo	Result: the word is starting with letter K and is a real word

V. RESULTS



Fig: HomeScreen

In the above figure we can see the home screen for the game we developed. It is the combination of horizontal and vertical layouts. We added some animations to the objects too, by hovering on any one of the boxes they are increased in size so we can show that they are selected. This is achieved using basic mouse events.

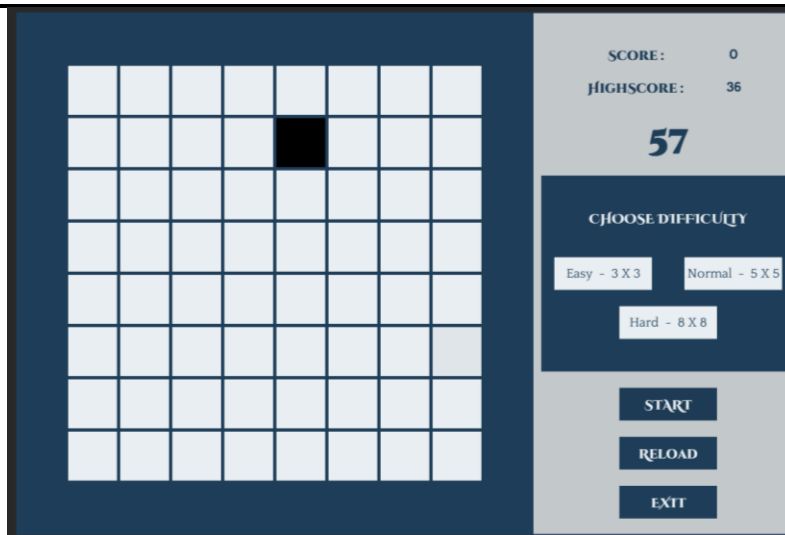


Fig: GameScreen

This is the game screen of one of the games we developed. As you can see there are 8X8 boxes. This is achieved using grid layout. We even gave the player to choose difficulty by giving the number of boxes to play with.

The left game panel and right details panel are created using a vertical layout group. The reason to use the layout groups is we can take the same layouts to the screen with small size and large size. The layout automatically adjusts to the screen size. This makes the UI more interactive and responsive.

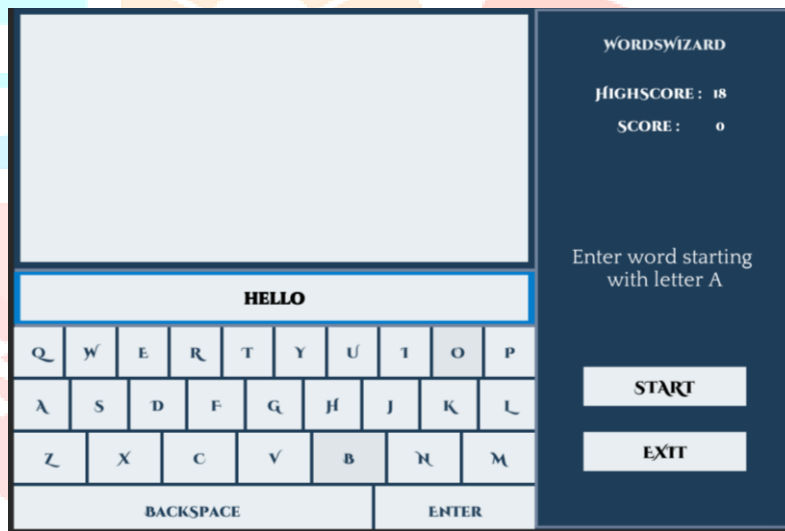


Fig: Word Game Screen

By the knowledge from the previous layouts we can see the above figure has multiple groups. Let me explain the layout structure. There are two vertical layout elements, one for details on the right and another for game view on left. Again the game view is divided into two using horizontal layout, screen and keyboard.

The game working mechanism is at the start the a letter is shown on the details side, then the player has to enter a word starting with the given letter and press enter, the user even has a button for removing deleting the letters entered before. When a player clicks enter the game takes the word from the box with hello in the above figure and checks whether the word entered is starting with the given letter and is present in the real world words list.

VI. CONCLUSION

We developed a cognitive game using unity's 2D tools. Unity's workflow is beginner friendly and easier to use. For every update and future versions the unity technologies are improving the unity engine. The tools are easy to understand and work with. With good practice and moderate effort anyone can make games with stunning visuals.

VII. ACKNOWLEDGEMENT

Apart from the efforts of us, the success of paper depends largely on the encouragement and guidelines of many others. We take this opportunity to express our profound gratitude to CMR Technical Campus College Management for motivating us and for providing all the facilities required for this work. We are deeply indebted to Chairman Shri C. Gopal Reddy, Secretary Smt. C. Vasantha Latha, Director Dr. A. Raji Reddy, HOD CSE Dr. K. Srujan Raju, PRC Co-ordinator Dr. Suvarna Gothane, Project Guide K Ranjith Reddy who always has been a constant source of inspiration for us.

VIII. REFERENCES

- [1] Unity Technologies.2017a. Learning the Interface.Available:
<https://docs.unity3d.com/Manual/LearningtheInterface.html>.
- [2] Unity Technologies.2017b. Project view.Available:<https://docs.unity3d.com/Manual/ProjectView.html>.
- [3] Unity Technologies.2017d. Game View.Available:<https://docs.unity3d.com/Manual/GameView.html>.
- [4] Unity Technologies .2017e .Hierarchy. Available:<https://docs.unity3d.com/Manual/Hierarchy.html>.
- [5] Unity Technologies. 2017f. Inspector Window.Available: <https://docs.unity3d.com/Manual/UsingTheInspector.html>. Accessed 14 September 2017.

