

---

## **THEORY OF COMPUTATION**

---

### **WEBSITE REPORT SUBMISSION**

[\*\*https://sites.google.com/view/toc-work-fa19/home\*\*](https://sites.google.com/view/toc-work-fa19/home)

Submitted to:

Dr. Muaz Ahmed Khan Niazi

Submitted by:

Muhammad Sajid Nawaz  
FA18-RCS-027

COSMOSE Research group, COMSATS University, Islamabad, Pakistan

## CONTENTS

<b>I</b>	<b>Summary of Achievements</b>	3
I-A	Mughees Ali . . . . .	3
I-B	Muhammad Sajid Nawaz . . . . .	3
I-C	Ammad Naizi . . . . .	3
<b>II</b>	<b>Summary of Project</b>	4
<b>III</b>	<b>Theme and Story line</b>	4
<b>IV</b>	<b>Language 1</b>	5
IV-A	Language 1: Mughees Ali . . . . .	5
IV-B	Proof . . . . .	5
IV-C	JFLAP Simulation . . . . .	5
IV-D	Accepted And Reject String Simulation	5
IV-E	Models . . . . .	7
IV-F	Game Demo . . . . .	9
IV-G	Web Details . . . . .	10
IV-H	Video Details . . . . .	10
IV-I	Video Description . . . . .	10
IV-J	Sounds Details . . . . .	10
<b>V</b>	<b>Language L2</b>	11
V-A	J-Flap Simulation . . . . .	11
V-B	Turing Machine Testing . . . . .	11
V-C	Game Demo . . . . .	11
V-D	Implementation details . . . . .	12
V-E	Video Details . . . . .	12
V-F	Sounds details . . . . .	12
<b>VI</b>	<b>Language 3</b>	15
VI-A	JFLAP Simulation . . . . .	15
VI-B	Game Demo . . . . .	16
VI-C	Models . . . . .	16
VI-D	Video details . . . . .	17
VI-E	Video Description . . . . .	17
VI-F	SOUNDS LINK . . . . .	17
<b>VII</b>	<b>Palindrome</b>	19
VII-A	Palindrome Simulation . . . . .	19
VII-B	Youtube Video . . . . .	19
VII-C	SOUNDS LINK . . . . .	19
<b>VIII</b>	<b>Code Screenshots</b>	21

## I. SUMMARY OF ACHIEVEMENTS

### A. Mughees Ali

Theory of automata is computer science core subject at bachelor level. Theory of automata deals with designing abstract self-propelled computing devices that follow a pre-determined sequence of operations automatically.

We have learned automata in our Computer Science Bachelor. We have missed many concepts, which we did not realize until we reached the Theory of Computation, and took us Dr. Muaz Niazi under his wings. We count ourselves on lucky under the supervision of Dr. Muaz Niazi. We really don't know, how much importance automata course have, but when we reached in the class of Theory of Computation, we really know about the importance of this course.

First of all, Dr. Muaz Niazi tells us basically Theory of computation course is the combination of the three areas automata, computability, and complexity. In the automata section, we learned Finite automata, regular languages, and regular grammars, context-free languages, and grammar, push-down automata, deterministic and nondeterministic automata. In the computability section, we learn all the problems are not solvable through computers. In this section, we learn some theorems and most important Turing Machine. We repeat all the concepts of bachelor and lot of additional things. We learn many of the formal mathematical models of the computation Along with their relations with the formal languages. We learned regular languages and context-free languages, they tell us how a programming language and compiler form and how they work. All these things, we learned in this course we implement it in different scenarios.

### B. Muhammad Sajid Nawaz

The theory of computation is the concepts of automaton, formal languages, grammar, algorithms, computability, decidability and complexity. The current focus of computer science on computational algorithm with respect to time and space complexity. I get the learning outcome of this course under the supervision of Dr. Muaz Ahmed Niazi, Chief Scientific Officer (i) design and analyse the DFA, NFA, formal languages, grammars, push-down automaton, Turing machines. (ii) able to demonstrate their understanding of basic and fundamental notations such as algorithm complexity, computability, decidability and complexity on cryptography and optimization through problem solving techniques. (iii) able to prove the basic result of theory of computations. (iv) able to state and explain the charge Turing thesis relevance. (v) able to use and run theory of computation simulation on Jflap software. (vi) I gain the knowledge of developing games in unity game environment. (vii). I get the deep knowledge of C scripting, animation, object creation, making sounds, using text materials, UI, practical system, terrain, Blender usage, and pro-builder. (viii) I also learned the Latex editor(overleaf) for document writing and assignments submission. (ix) I also get the fast-growing knowledge of latest and new technology of Quantum computing and DNA computing that is the hot area of research. After reading the different articles of the Quantum and DNA computing reading, I gain the knowledge

and research interest in field of Quantum computing. I am able to determine that how much future computing will be changed and very fast processing and storage units will be occupied. (x) I get lot of inspirational and interesting informational knowledge from Pro.Dr. Muaz Ahmed Khan Niazi. He guides me that how to handle the difficulties of life. He also gives us Islamic information and knowledge and change the mode of my life. I am proud of being his student, feel himself different from other normal students because of my honourable supervisor Dr. Prof.Muaz Ahmed Khan Niazi, chief scientific officer. I will ever remind and thankful to him.

I have also worked on Adobe photoshop and Microsoft Publisher. We use JFlap for the real modelling of automata machines. We use the concepts of Theory of Computation in Unity. We didn't know how a Unity platform work. We learn how a Turing machine is implement in Unity with the Scripts of C language. We implement Finite State Machine (FSM) and Turing Machine (TM) in Unity with C code. We also repeat all the concepts of Object-Oriented Programming (OOP). We learn, how to create a library for the C code reusability. We implement the stack in out unity game.

I have also worked on the LATEX for project documentation. All these things, we learn in this course are not possible without the help of Dr. Muaz Niazi. We really thank honourable Dr. Muaz Niazi.

### C. Ammad Naizi

I am very excited about the outcomes of this course. The aim of this course is to introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability. Moreover, to enhance/develop students' ability to understand and conduct mathematical proofs for computation and algorithms. This course has two major parts Theoretical concepts and Practical implementations. The first part defines machine models formally by defining Finite Automata, Regular Languages and Turing Machines etc. The second part comprise of hands on implementation. After completing the final project I am in a position to explore practical implementation of Turing Machine using MS Visual Studio 2017 and Unity 2018.4.f1. It was great pleasure to work on unity animations, creation of blender models, unity objects with Turing Machine running on back-end. The unity interface consist of text box which gets input from user and pass the given string to Turing Machine which checks whether the input string is valid or not.

Following are the major tasks I did during the entire course:

Theoretical concepts with theorems, examples and proofs of the following:

- Finite Automata and its applications,
- Regular Expressions and Kleene's theorem,
- Nondeterminism and conversions,
- Properties and applications of Regular Languages,

- Push Down Automata with stack for Context Free Languages and applications,
- Turing Machine for regular and non-regular languages with its variations, decidability, halting problems and reducibility,
- Space and time complexity of machines.

#### Practical Implementations:

- Designing of Finite Automata using J-Flap for regular languages (DFA, NFA).
- Designing of Push Down Automata using J-Flap for non-regular languages and also implemented the generic PDA in C using MS Visual Studio 2017
- Generic FSM class implementation in C of the Pac Man game.
- Turing Machine designing in J-Flap and its implementation in C for palindrome and the language used in our project.
- Creation of game scene in Unity 2018.4.f1.
- Designing of blender models and animations in Unity.
- Integration of Turing Machine code with Unity.

## II. SUMMARY OF PROJECT

In this project, we have the main page which consists of each group member button, buttons lead us to the three-game scene, palindrome scene button and exit button. Palindrome machine gets input string from the user, generates cubes, the input string is shown on the cubes. Different sounds affect are set on click, rejection and acceptance of string. Each group has one scene which shows the Turing machine. Each group member scene consists of models, animations and sound effects. To return to the main page, the home button is placed at each scene.

In Sajid Nawaz scene, as we give input to the Turning machine, turning machine generates cubes and each string will show on cubes. A reset button to end the scene or to generate a new string for the language. In this scene, five models are used i.e. army-men, stope, bullets, container and gun which we build using blender software. I have used five Animations in this scene i.e. particles move, bullet, stop sign mover, reject and accept. Whenever the input string is accepted, a thumbs-up is shown and it will move around. Whenever the input string is rejected, sprite animation is a move. I have also used different type of sound effect on the movement of cubes, on rejecting input string and accepting input string.

In Mughees Ali scene, as we give input to the Turning machine, turning machine generates cubes and each string will show on cubes. A reset button to end the scene or to generate a new string for the language. In this scene, five models are used i.e. box, bomb, helicopter, house, tree and wood which we build using blender software. I have used five Animations in this scene i.e. snow movement, eagle fly, helicopter movement, reject and accept. Whenever the input string is accepted, accepted animation in move around. Whenever the input string is rejected, rejected animation is move. I have also used different type of sound effect on the movement of cubes, on rejecting input string and accepting input string.

In Ammad Niazi scene, as we give input to the Turning machine, turning machine generates cubes and each string will show on cubes. A reset button to end the scene or to generate a new string for the language. In this scene, five models have used i.e. road, car, army men, floor and mini-plane which we build using blender software. I have used five Animations in this scene i.e. care movement, army-men movement, mini-plane movement, reject and accept. Whenever the input string is accepted, accepted animation in move around. Whenever the input string is rejected, rejected animation is move. I have also used different type of sound effect on the movement of cubes, on rejecting input string and accepting input string. We used different tools for modeling and simulation of machines.

- We have built theme-based defense game using unity tool, while programming is done in Visual Studio.
- Blender
- Photoshop
- Unity 3D
- JFlap

## III. THEME AND STORY LINE

The theme we selected for this game is “Call of Duty” which is military video game initially developed by infinity ward in 2003, later released by Trey Arch in 2005 and Sledgehammer Games in 2013. The initial versions of the game shows the events of World War 2. However, the later versions were upgraded to modernized warfare which focused on fifth generation cold warfare. The motivation behind choosing this theme is our brave armed forces which always stand shoulder by shoulder in the defense of mother land. Armed forces are the motivation of many young generation and every year thousands of soldiers across the country join armed forces to serve the nation. The armed forces are not only attractive by their uniforms. However, it is the discipline which made armed forces unique from others departments in the country. The main event in the life of soldiers is war which is nothing but destruction and lose situation to the economy, education, man power, land and ideology of the country. War is the actual test of human courage. However, a human courage is stronger than human itself. A military leader always do not afraid of war. In fact, He is always worried about the courage of human and nothing exists after broken hope and courage.

This project contains similar story relevant to human courage, bravery, act of valor, and glorious resolve. A company of few peoples with limited logistic and weapon support is trying to defend their piece of land under the harsh weather conditions. These soldiers represents the brave Pakistan army standing along the line of control near Indo Pak border performing their duty in the area of Siachin kargil sector. The temperature of Siachin Kargil sector is about to reach -50 degree in winter season which is unacceptable living condition for human being to survive. But these are the men made of steel and courage. These are the men bring back happy life to civil community. These are the men who stand in the roars of bullets and in the thunder of bombs knowing that they are

surrounded by death, knowing that they could leave their families alone. However they just keep on moving because some thing is pumping in their hearts and flowing in their veins known as love, devotion and courage for their county and they love death before disgrace. Millions of peoples sleep well in their homes because they have trust on their army standing on the borders. Pakistan Army has fought 4 wars against India and the war on terror since 2001.

- In the first Kashmir war against India in 1947, Pakistan army with support of local tribes was able to capture the 83000 km<sup>2</sup> land of the Kashmir. Today 73000 km<sup>2</sup> area is current Gilgit Baltistan and rest of the 13,000 km<sup>2</sup> is Azad Kashmir.
- In 1965, the second Kashmir war fought between two South Asian countries. Both claims victory. However, the story of courage and bravery by the soldiers of Pakistan army by defending their cities of Lahore, Kasur and Sialkot. In the war of 1965, Pakistan army shoot down 112 Indian fighter jets, destroyed 460 Indian tanks and killed more than 5 thousand Indian soldiers. Pakistan army not only defend the Lahore and Sialkot but captured the Indian town of Khemkaran which was the big surprise for India at that time. However, Pakistani politicians failed to resolve the Kashmir dispute and no territorial changes observed after the Tashkent agreement in 1966. Technically Pakistan army fought well and had up hand over Indian armed forces.
- In the war of 1971 which is called Bangladesh Liberation War, Pakistan army tried its best to defend the both eastern and western part of the country. However due to large number of Bengali traitors, Pak army failed to defend eastern border and unconditional cease fire was declare at the western end. However, Pakistan armed forces bear huge loss to its all forces including Army, Navy and Airforce.
- In 1999, Kargil war Pakistan army successfully captured the whole Siachin Glacier inside the Indian Territory. However, again due to poor political strategy and external pressure Pak army forced to withdrew

#### IV. LANGUAGE 1

##### A. Language 1: Mughees Ali

My language is an  $b^2n\ a^n$  for the values  $n=1,2,3$ , and so on. This language is taken form the book Introduction to Computation Theory by Denial Cohen, page number 474 and it's an exercise question number 13. For this language I created eight states from Q1 to Q8. At each state I read a character (a or b) replace it with other variable (x or y or z) and move left or right according to language. User enter string which consists of two type of variable i.e. a, b. At the end if the string is valid it is accepted otherwise the string is rejected by the machine.

##### B. Proof

##### For Non Regular:

Suppose a language  $a^* \ b^* \ a^*$ , in this language generated

string depends on the value of \*. If the \* value is 1,2,3,... we have equal number of aba string respectively i.e. aba, aabbaa, aaabbbaaa. While in our language case we get strings like abba, aabbbaa, aaabbbaaa which mean that they are different. That's why this language is Non-Regular language.

##### Non-Context Free Pumping Lemma:

**Reference:** Book Introduction to Computation Theory by Denial Cohen, page number 470.

$A = \{a^n b^{2n} a^n | n \geq 1\}$  is Non-Context Free.

**Proof:** Assume, to the contrary, that A is context free. By Pumping Lemma there exists a constant p such that every  $w \in A$  of length  $\geq p$  is divided into  $w = uvxyz$  such that  $|vxy| \leq p$ ,  $|vy| \geq 1$ , and for every  $I \geq 1$ ,  $uv^i xy^i z \in A$ . Let  $w = a^p b^{2p} a^p$ .

Since  $|vxy| \leq p$ , vxy is either  $a^* b^*$  or  $b^* a^*$ . So it is not the case  $uv^2 xy^2 z$  has the same number of as, bs, as.

Hence it is Non-Context free language.

#### C. JFLAP Simulation

JFLAP file screenshot is given below.

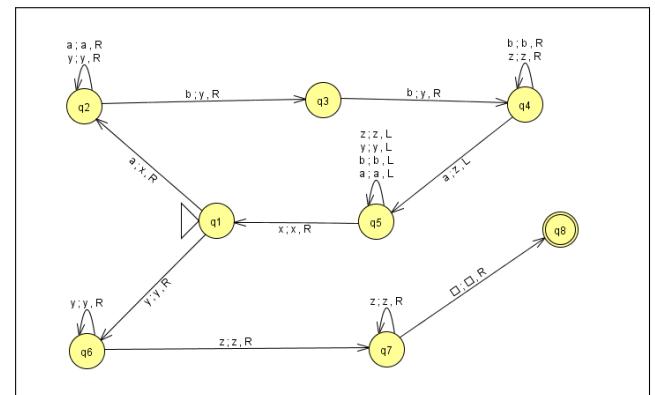


Fig. 1: JFLAP Model

R = Right Move, L = Accept Move.

There are different states from Q1 to Q8. Every state performs different work and at end finally string accepts or rejected by machine.

#### D. Accepted And Reject String Simulation

I entered a valid string abba:

The string abba is valid so machine accepts. Green color show that the string entered is successfully accepted.

Current State	Reads	Replace	Move	Next State
Q1	a	x	R	Q2
Q2	a	a	R	Q2
Q2	y	y	R	Q2
Q2	b	y	R	Q3
Q3	b	y	R	Q4
Q4	b	b	R	Q4
Q4	z	z	R	Q4
Q4	a	z	L	Q5
Q5	a	a	L	Q5
Q5	b	b	L	Q5
Q5	y	y	L	Q5
Q5	z	z	L	Q5
Q5	x	x	R	Q1
Q1	y	y	R	Q6
Q6	y	y	R	Q6
Q6	z	z	R	Q7
Q7	z	z	R	Q7
Q7	e	e	R	Q8
A	A	A	A	A

TABLE I: Language 1 Table

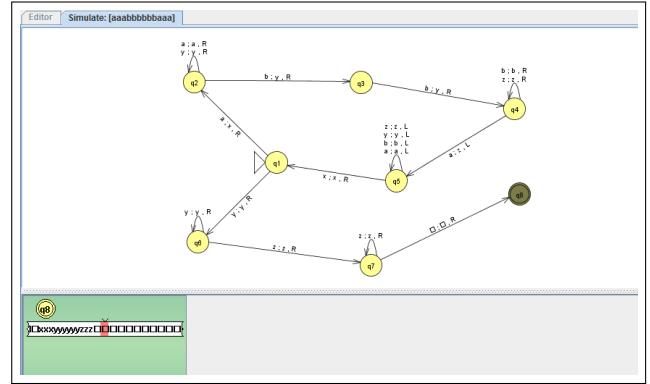


Fig. 4: aaabbbaaa Accept String Simulation

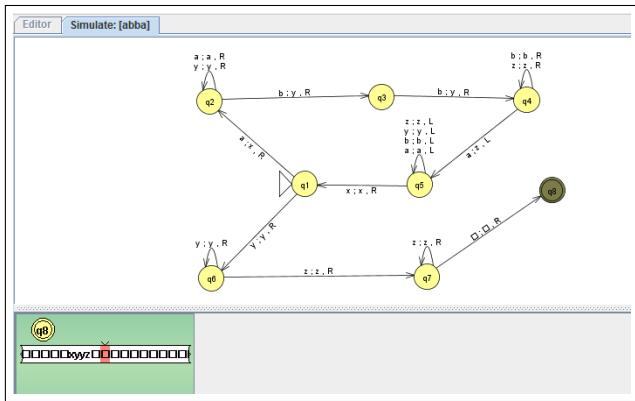


Fig. 2: abba Accept String Simulation

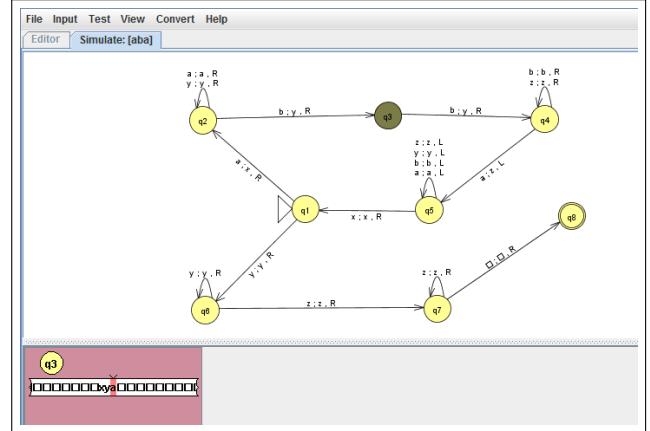


Fig. 5: aba Rejected String Simulation

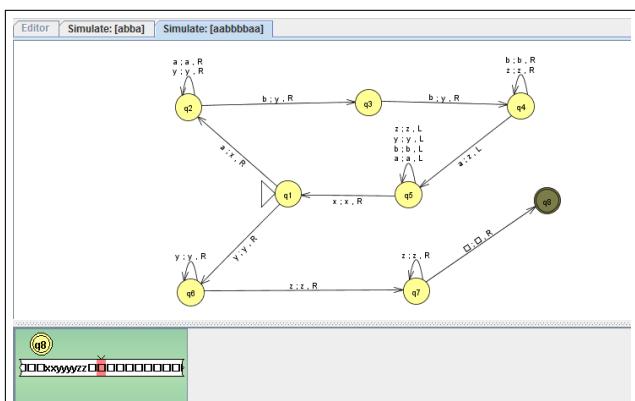


Fig. 3: aabbbbaaa Accept String Simulation

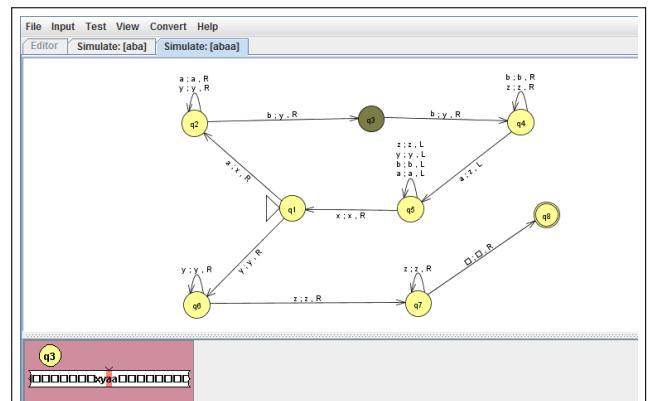


Fig. 6: abaaa Rejected String Simulation

### E. Models

Models are created in blender. I created 6 different models in blender. These models are Helicopter model, Bomb model, Box model, House model, Woods model, Tree model.

In blender we have two main modes, one is Object mode other is Edit mode.

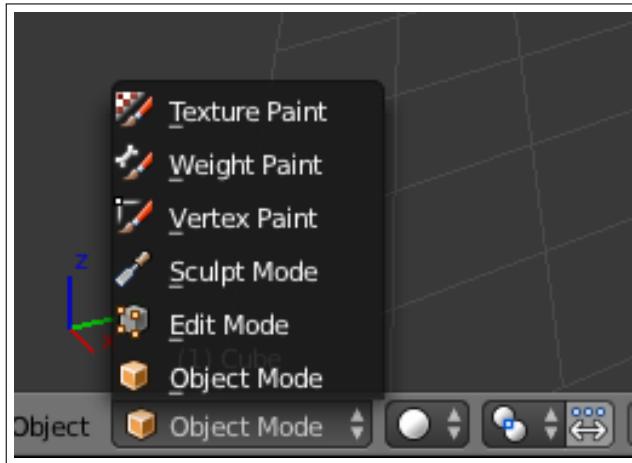


Fig. 7: Blender View

By these modes we select objects related to our models and make changes according to model requirements.

#### Helicopter Model:

This model is designed for giving a look of Theme (Call of duty). This model is also used in animation in order to show the real effect like a helicopter moves from one place to another.

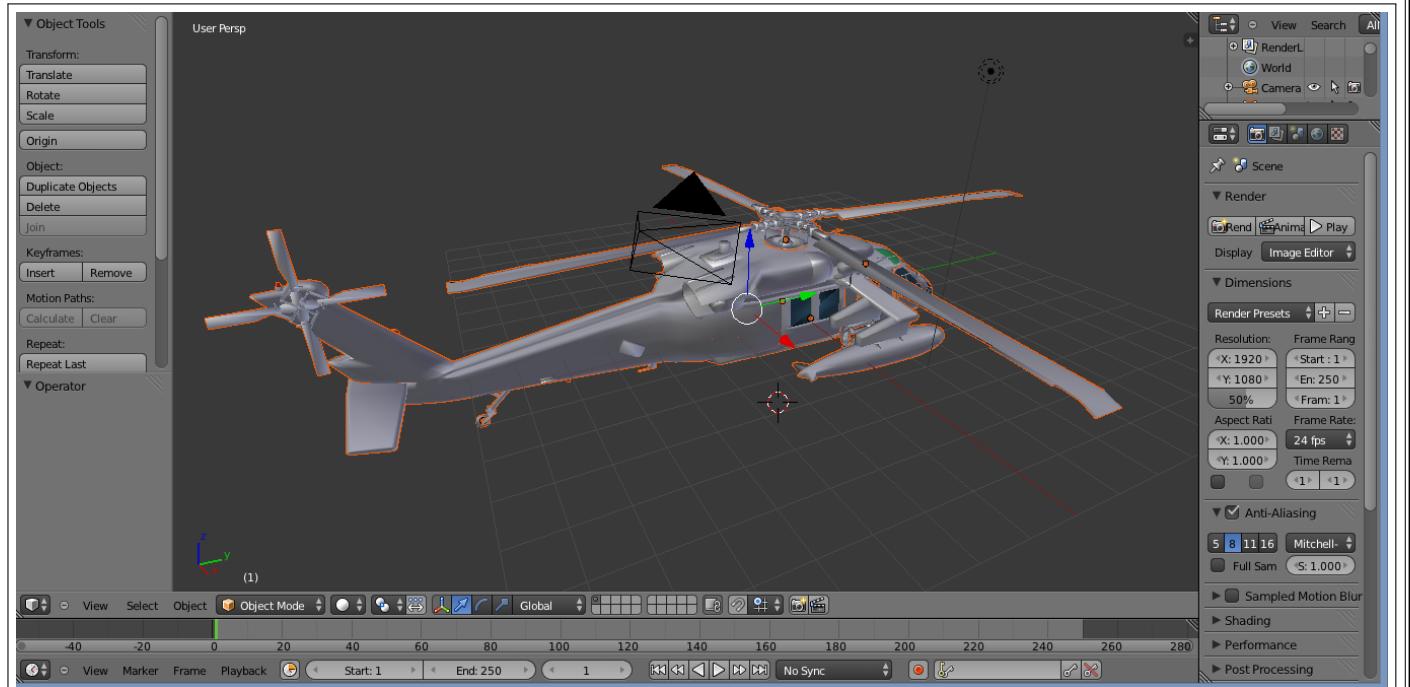


Fig. 8: Helicopter Model

#### Tree Model:

This model is designed according to theme requirement. Trees are set at the back of Scene to give a look for hiding both for soldiers and for enemy.

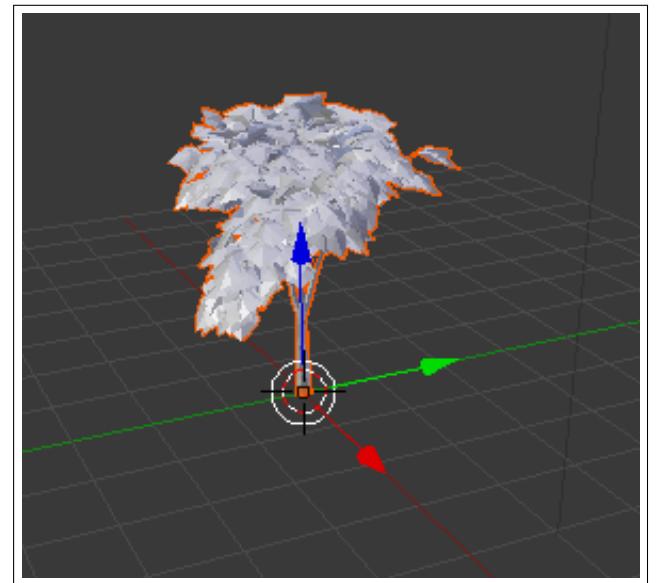


Fig. 9: Tree Model

### Bomb Model:

Bomb model is design according to theme requirement. This model is related to blast part in game.

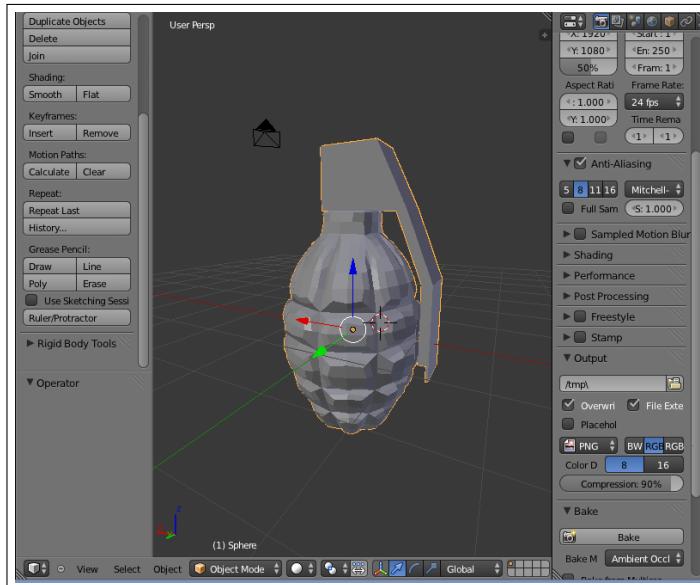


Fig. 10: Bomb Model

### Box Model:

Box model is design according to theme requirement. This model is design in order to carry bombs and other things.

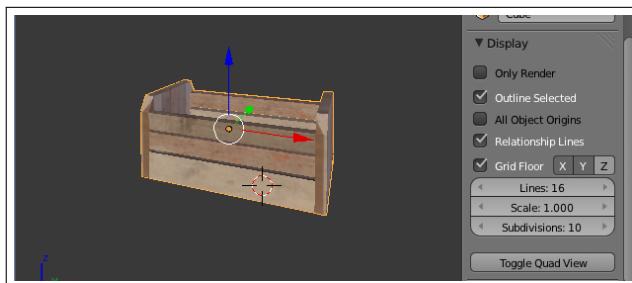


Fig. 11: Box Model

### Wood Model:

Wood model is design according to game theme. Fire animation is set on woods to show the real effect of fire.

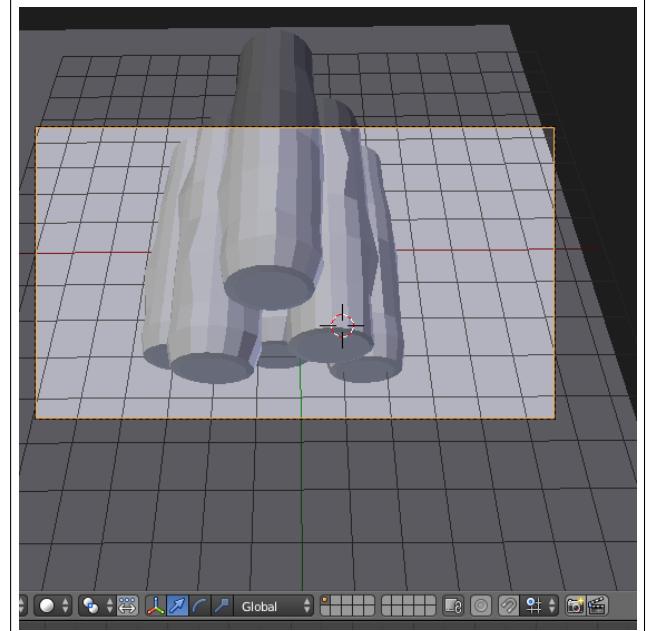


Fig. 12: Wood Model

### House Model:

This model is theme requirement. House is used for shelter in game.



Fig. 13: House Model

#### F. Game Demo

Here is the main menu. We have five different buttons in which 4 buttons are for languages and one button for the exit.

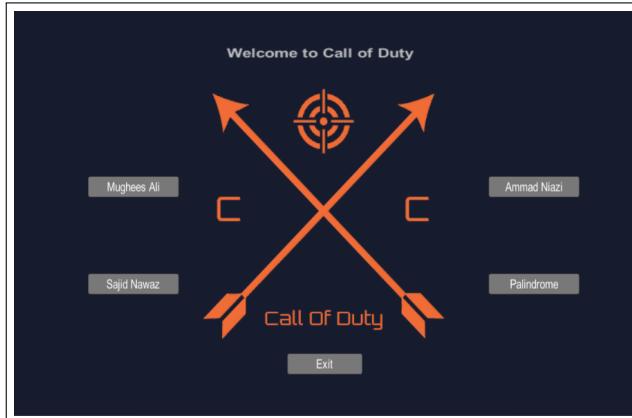


Fig. 14: Main Menu

Clicking on members name will leads you to their respective language section. I am Mughees Ali, so when I click on my name you will see my sub menu.

Here you can see three different models box model, bomb model, helicopter model. Animations like eagle flying, snow falling and helicopter moving and also set in this scene.



Fig. 15: Mughees Ali Sub Menu

In this sub menu you have to enter a valid format string, if the string is in wrong format (contain other than a and b letters) you should not precede further.

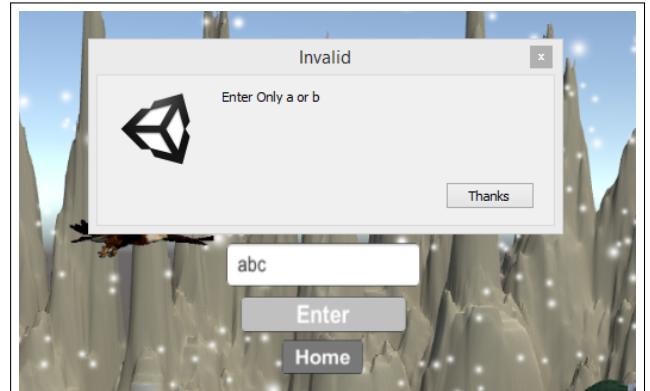


Fig. 16: Wrong Format Input

If the entered string format is valid then you go to machine scene.

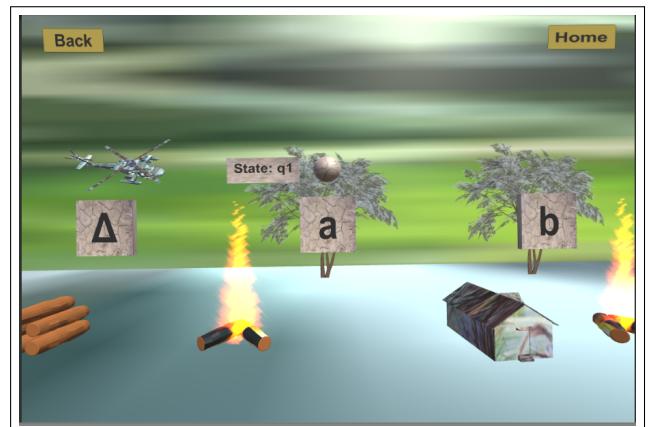


Fig. 17: Machine View

This is the machine scene. Here you can see the string characters on cubes, two button one for back to sub menu and one for back to main menu. Fire animation is set when the camera moves from one point to another point. Helicopter animation is set with the moment of camera. Accept animation starts play when the valid string is accepted, and reject animation is starts play when the invalid string is entered.

When the string is accepted animation starts play. By

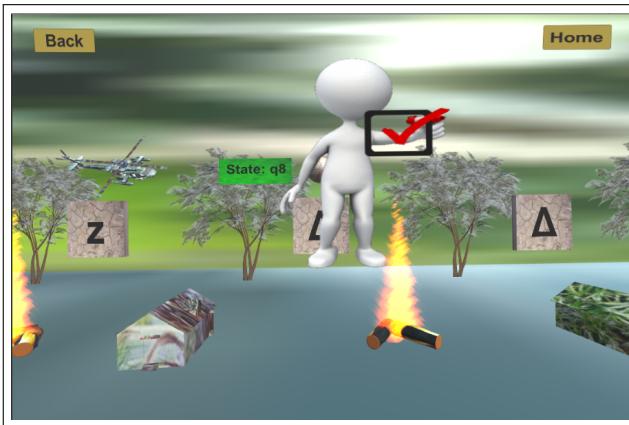


Fig. 18: Animation on Accept String

pressing Back button you can go to Mughees Ali sub menu and if you press Home button you go to Main Home Menu. When the string is invalid animation starts play.

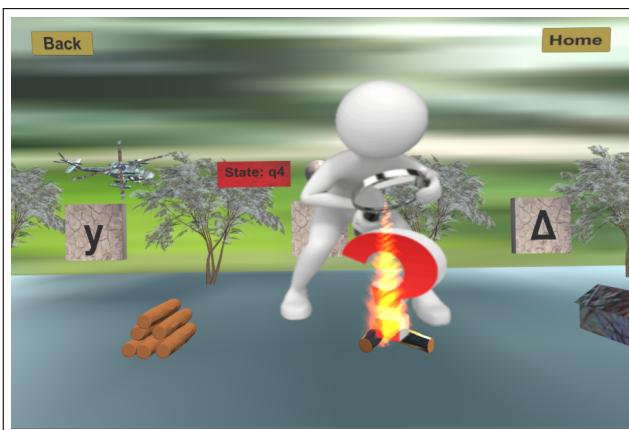


Fig. 19: Animation on Reject String

By pressing Back button you can go to Mughees Ali sub menu and if you press Home button you go to Main Home Menu.

#### G. Web Details

##### Web Link:

<https://sites.google.com/view/toc-work-fa19/home>

##### Web link of Mughees Ali Profile:

<https://sites.google.com/view/toc-work-fa19/mughees-ali>

#### H. Video Details

##### Youtube video link:

<https://youtu.be/tEq6G9DEhwA>



Fig. 20: Youtube Screenshot

#### I. Video Description

This video consists of a unity scene with Turing Machine working on the back end. The language of this Turing Machine is  $A = \{a^n b^{2n} a^n | n \geq 1\}$  for the values  $n=1,2,3,4$ . The minimum string S accepted by Turing Machine is S= 'abba'. The animations are self-designed and sounds are self-generated.

##### Web Links:

<http://nizilab.com/>

<https://sites.google.com/view/toc-work-fa19/home>

#### J. Sounds Details

Three different sounds set on my language, one is accept sound, one is reject sound and last one is space sound. Here are the links:

##### Accept Sound:

[www.soundcloud.com/hammad-niazi-483497094/g15-mughees-accept](http://www.soundcloud.com/hammad-niazi-483497094/g15-mughees-accept)

##### Reject Sound:

[www.soundcloud.com/hammad-niazi-483497094/g15-mughees-reject](http://www.soundcloud.com/hammad-niazi-483497094/g15-mughees-reject)

##### Space Sound:

[www.soundcloud.com/hammad-niazi-483497094/g15-mughees-space](http://www.soundcloud.com/hammad-niazi-483497094/g15-mughees-space)

## V. LANGUAGE L2

**Reference:** Book Exploring the Power of computing by John E. Savage

Exercise 5.3 Page 223 Models of Computation

The language of this Turing Machine is  $A = \{x^i y^j x^k | i, j, k \geq 1\}$  and  $k = i+j$ . This language is non-regular and non-context free [1]. The minimum string S accepted by Turing Machine is  $S = 'xyxx'$  where S belongs to L. However, any string that does not satisfy L will be rejected by this Turing Machine. The value of i, j and k should be greater than or equal to 1 and the value of k is addition of i and j. This Turing Machine has 8 states in total represented by Q1 to Q8 including one initial state Q1 and one final state Q8 (accepted state). This machine checks the string step by step and finally gives the result whether the string is accepted or rejected. Example: User enters the string S = 'xyxx'. The machine initially reads the first alphabet, replace it with another symbol ('a' in this case) and move ahead. If the first symbol is other than 'x' then string is rejected. Similarly the machine reads the second, third symbol and replace it with other symbols ('b' and 'c' in this case) respectively. Eventually, if the machine reads null symbol from input replace it with null and the string is accepted else rejected.

**Proof:** We will proof the language  $A = \{x^i y^j x^k | i, j, k \geq 1\}$  and  $k = i+j$  is non-regular and non- context free language by using Pumping Lemma. Assume, to the contrary, that L is context-free. By Pumping Lemma, there exists a constant P Such that  $w \in L$  of length  $\geq p$  is divided into  $w = uvxyz$  such that  $|vxy| \leq p$ ,  $|vy| \geq 1$ , And for every  $I \geq 0$ ,  $uvi xyi z \in L$ . Let  $w = xpypxp$  since  $|vxy| \leq p$ ,  $vxy$  is either in  $x^* y^* x^*$ . so it is not the case  $uv^2xy^2z$  has the same mean  $x^i y^j x^k$  where  $i, j, k \geq 1$  and  $k = i+j$ .

### A. J-Flap Simulation

The Turing Machine is designed and tested in J-Flap. Moreover, the machine is tested on 3 strings with 2 strings belongs to the language of this machine which are accepted and 1 string that does not belong to this language which is rejected. The complete step by step execution detail of the Turing Machine and snapshots of each string tested (input, output) is given below:

### B. Turing Machine Testing

In first scenario, input string is "xyxx" and the value of 'i' and 'j' are 1 and It is observed from the fig that given string is recognized by language and accepted by machine on state q8.

In second scenario, input string is "xyxxxx" and the value of 'i' and 'j' are 2 and it is observed from the fig that given string is recognized by language and accepted by machine on state q8.

In third scenario, input string is "xyx" and the value of 'i' and 'j' are 1 and it is observed from the fig that given string is not recognized by language and rejected by machine on state q4. This string is rejected because we know that value of 'k' is addition of 'i' and 'j' which should be greater than value of

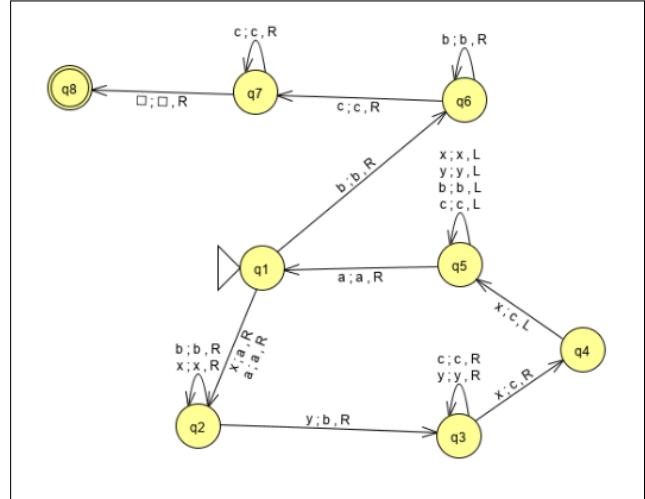


Fig. 21: JFLAP File

Current	Read	Relace	Move
q1	x	a	q2
q2	b	b	q2
q2	x	x	q2
q2	y	b	q3
q3	c	c	q3
q3	y	y	q3
q3	x	c	q4
q4	x	c	q5
q5	x	x	q5
q5	y	y	q5
q5	b	b	q5
q5	c	c	q5
q5	a	a	q1
q1	b	b	q6
q6	b	b	q6
q6	c	c	q7
q7	c	c	q7
q7	e	e	q8

TABLE II: Language 2 Table

individual 'i' and 'j'. However, in this case the value of 'x' is equal to the value of 'i' and 'j' which is not possible.

### C. Game Demo

#### Main Menu:

In the main menu, user can input the string which he wants to check whether recognized by language or not.

#### Sub Menu:

In the sub menu Turing Machine reads the input step by step and eventually gives the result. Turing Machine accepts the string if the given input is recognized by the language else reject the string.

#### Accept:

The accept menu shows that the string is recognized by language and accepted by Turing Machine.

#### Reject:

The reject menu shows that the string is not recognized by language and rejected by Turing Machine.

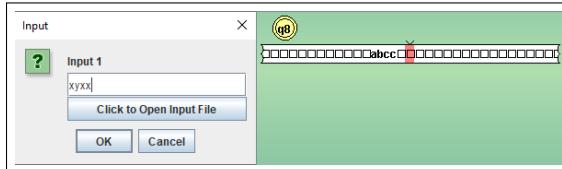


Fig. 22: Accept Simulation

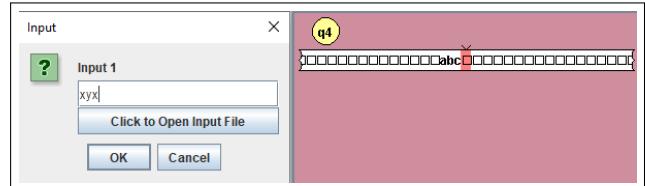


Fig. 24: Reject Simulation

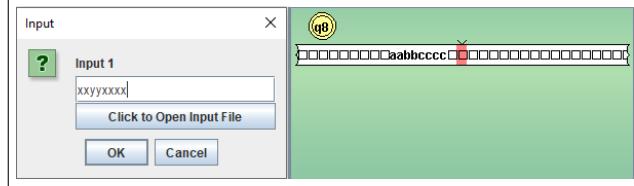


Fig. 23: Accept Simulation

#### D. Implementation details

##### Road:

The road model shows the special track made by Engineers group from the Army which is mostly used for the smooth movement of heavy armored vehicles like tanks, trucks etc. The movement of vehicles is very important inside war zones for logistic, fighting and rescue operations.

##### Jeep:

Jeep shows the movement of army convoy in a safe way. In war zones jeeps are very useful and mostly used to carry soldiers from one place to other safely. These type of jeeps are bullet proof and saves the lives of many soldiers in the war.

##### Army Man:

The army mans are the main asset and the main spirit of any war in the world. The army man represents the strong defense made by army to defend their land. Moreover, army man shows the defense strategy made by their team.

##### Mini Plane:

The mini plane represents the air power of the force. A country without a strong Air Force is at the mercy of any aggressor. Airforce plays an important role in the warfare for logistic supply, fighter and rescue operation. A country having strong and skilled air force is proved to be furious for the enemy.

##### Gun:

The gun is the most common hand mounted weapon used in the war and represents the courage and activeness of the soldier. Gun is the major weapon used in this theme and hold by every soldier.

#### E. Video Details

This video consists of a unity scene with Turing Machine working on the back end. The theme of this video shows some military installments got from the video game “Call of Duty” which is military fighting game. Moreover, this scene consist of a main menu and a sub menu. In the main menu there exist a text box in which user can enter a string to check whether the string is recognized by language or not. In the sub-menu the string runs step by step and gives the result in the end. The animations and sounds are self-generated and used to represent transition, accept and reject events in the Turing Machine.



Fig. 25: Ammad Main Menu

The language of this Turing Machine is  $A = \{x^i y^j x^k | i, j, k \geq 1\}$  and  $k = i+j$ . The minimum string S accepted by Turing Machine is  $S = 'xyxx'$ .

##### Web Links:

<http://niazilab.com/>

<https://sites.google.com/view/toc-work-fa19/hammad-niazi>

<https://youtu.be/sjxLKRHqblg>

#### F. Sounds details

All the sounds used in the project are self-generated and owned by the group member. There are 3 major events occurring in the Turing Machine, transition from 1 state to other, accept string, and reject string. All the 3 events are represented by 3 different animations supported by 3 different sounds. These sounds are uploaded on sound cloud and have open access to public. The links of these sounds from the sound cloud are given below.

##### String Accept:

[www.soundcloud.com/hammad-niazi-483497094/hammad-accept](http://soundcloud.com/hammad-niazi-483497094/hammad-accept)

##### String Reject:

[www.soundcloud.com/hammad-niazi-483497094/a-reject](http://soundcloud.com/hammad-niazi-483497094/a-reject)

##### Transition:-

[www.soundcloud.com/hammad-niazi-483497094/a-space](http://soundcloud.com/hammad-niazi-483497094/a-space)



Fig. 26: Machine View

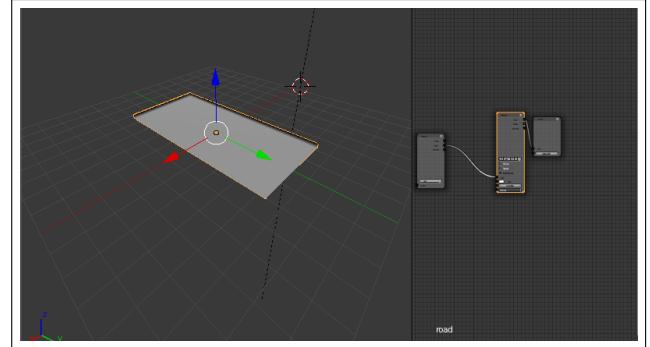


Fig. 29: Road Model



Fig. 27: Accept Animation

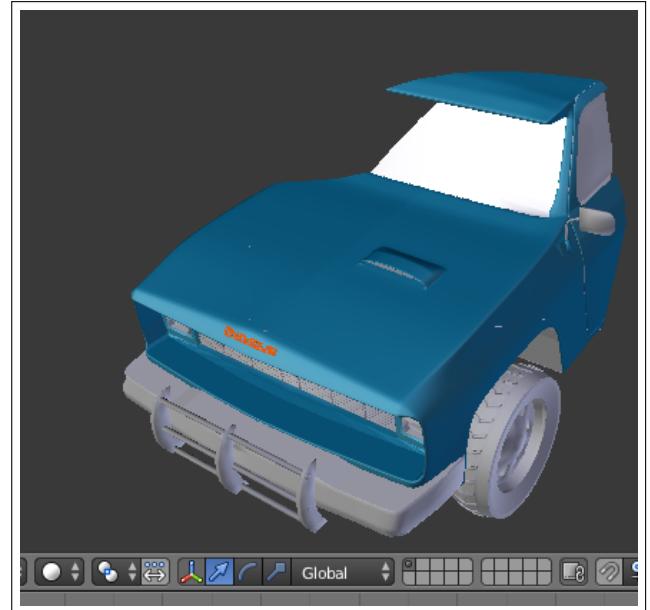


Fig. 30: Car Model



Fig. 28: Reject Animation



Fig. 31: Man Model

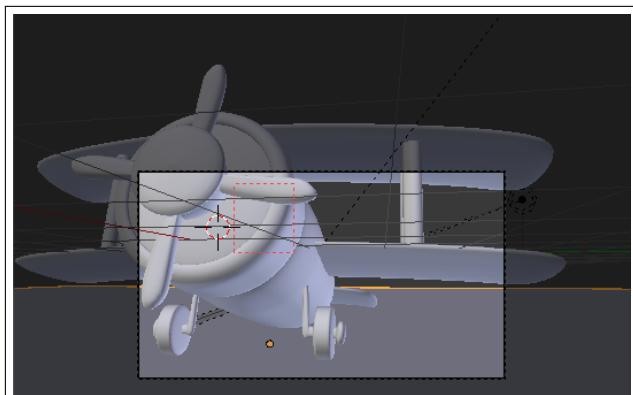


Fig. 32: Mini Plane Model

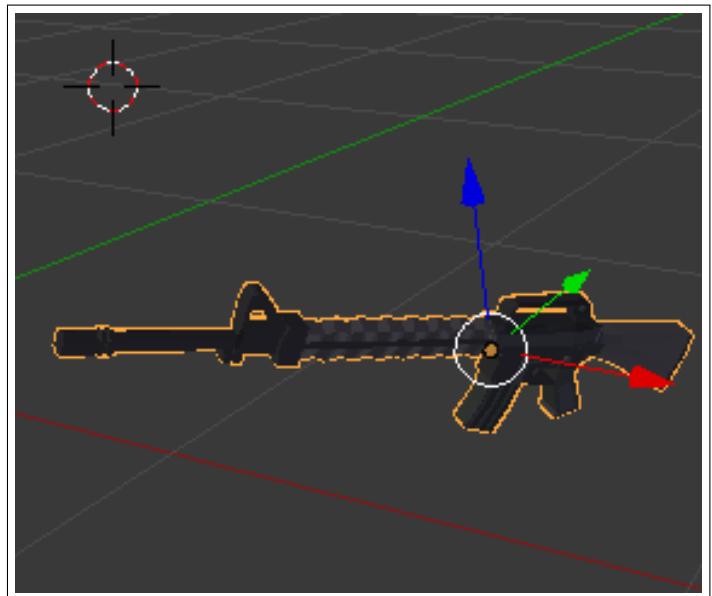


Fig. 33: Gun Model



Fig. 34: Youtube Video

## VI. LANGUAGE 3

**Reference:** Book Introduction to Languages and the Theory of Computation by John C Martin, Page 220, Q 6.2

The language implements in JFLAP tool, which contain 10 states ( $q_1, q_2, q_3, \dots, q_{10}$ ) where  $q_1$  initial state and  $q_{10}$  is a final state, machine tape and machine header which move left-right, movement depend on a given input string. i.e. if machine get input string  $a$  at  $q_1$ , machine replace  $a$  by  $w$  and header move to right, now current state is  $q_2$ , on state  $q_2$  if machine get input string  $a$ , machine replace  $a$  by  $a$  and header move to state  $q_2$  if machine get  $x$  as input string at state  $q_2$ , machine replace  $x$  by  $x$  and header move to state  $q_2$  if machine get input string  $b$  at state  $q_2$ , which replace by  $w$  header move to state  $q_3$ . Final state  $q_{10}$  will accept or reject the input string.

**Proof** We will proof the language  $A = \{a^n b^m a^n b^{n+m} | n = m \geq 1\}$  is non-regular and non- context free language by using Pumping Lemma.

Assume, to the contrary, that  $A$  is context-free. By Pumping Lemma, there exists a constant  $P$  Such that  $w \in A$  of length  $\geq p$  is divided into  $w = uvxyz$  such that  $|vxy| \leq p$ ,  $|vy| \geq 1$ , And for every  $I \geq 0$ ,  $uv^I xy^I z \in A$ .

Let  $w = ap bp ap bp + p$  since  $|vxy| \leq p$ ,  $vxy$  is either in  $a^*b^*$  or  $b^*a^*$ . so it is not the case  $uv^2xy^2z$  has the same mean  $bmanbn + m$ .

Thus,  $A$  is a non-regular and non-context free language.

### A. JFLAP Simulation

Our machine consists of states ( $q_1, q_2, q_3, \dots, q_{10}$ ), tape and header which move right(R) left(L), where  $q_1$  initial state and  $q_{10}$  is a final state, machine tape and machine header which move left-right, movement depend on a given input string. i.e. if machine get input string  $a$  at  $q_1$ , machine replace  $a$  by  $w$  and header move to right goes to state  $q_2$ , on state  $q_2$  if machine get input string  $a$ , machine replace  $a$  by  $a$  and header move to state  $q_2$  if machine get  $x$  as input string at state  $q_2$ , machine replace  $x$  by  $x$  and header move to state  $q_2$ , if machine get input string  $b$  at state  $q_2$ , which replace by  $w$  header move to state  $q_3$ . Final state  $q_{10}$  will accept or reject the input string. Transition table of machine is given below

On testing the machine, we give input string  $ababb$  to a machine which accepts at state  $q_{10}$  as shown in fig.

On testing the machine, we give input string  $aabbaabb$  to a machine which accepts at state  $q_{10}$  as shown in fig.

Current-State	Symbol-Read	Replace-By	Header-Move	Next-State
Q1	a	w	R	Q2
Q1	x	x	R	Q7
Q2	a	a	R	Q2
Q2	x	x	R	Q2
Q2	b	y	R	Q3
Q3	b	b	R	Q3
Q3	y	y	R	Q3
Q3	a	y	R	Q4
Q4	z	z	R	Q4
Q4	a	a	R	Q4
Q4	b	z	R	Q5
Q5	b	z	L	Q6
Q6	a	a	L	Q6
Q6	b	b	L	Q6
Q6	x	x	L	Q6
Q6	y	y	L	Q6
Q6	z	z	L	Q6
Q6	w	w	R	Q1
Q7	x	x	R	Q7
Q7	y	y	R	Q8
Q8	y	y	R	Q8
Q8	z	z	R	Q9
Q9	z	z	R	Q9
Q9	e	e	R	Q10

TABLE III: Language 3 Table

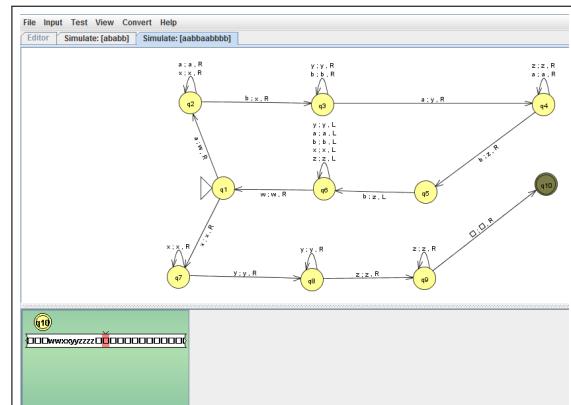


Fig. 35: Accept Simulation

On testing the machine, we give input string  $ab$  to a machine which rejects at state  $q_3$  as shown in fig.

On testing the machine, we give input string  $bbaa$  to a machine which rejects at state  $q_1$  as shown in fig.

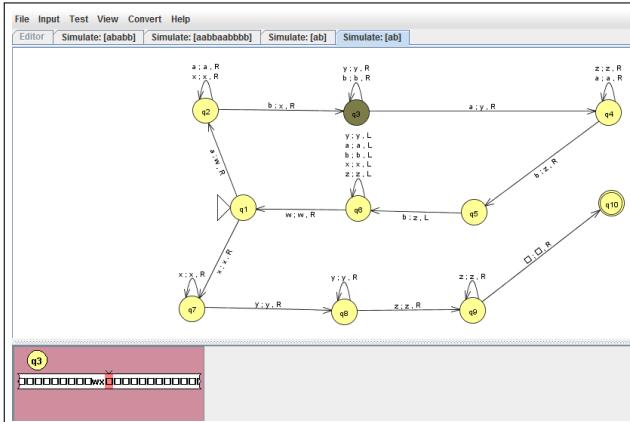


Fig. 36: Reject Simulation

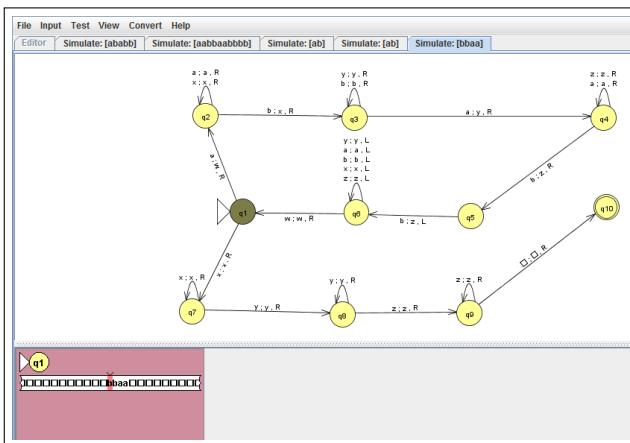


Fig. 37: Reject Simulation

### B. Game Demo

Our project main screen consists of five buttons i.e. Sajid Nawaz button which leads to my submenu, Ammad Niazi button, palindrome button and Exit button use to exit from the project, as shown in below figure. After clicking the Sajid

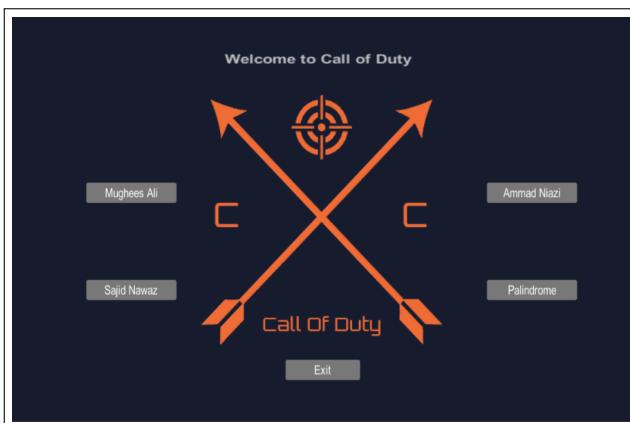


Fig. 38: Main Menu

Nawaz button submenu will appear. This menu consists of an input field, home button and enters button. Input field used to get the input string from the user and home button used for

return to the home menu. After given input, we click enter button to run a machine. As shown in the below figure.



Fig. 39: Sajid Sub Menu

After given an input string to a machine, the machine will generate boxes and set each character on boxes. This screen shows the current state of the machine and home button which is used to return to the home page. As shown in the below figure.



Fig. 40: Sajid machine

If the user enters the correct input string (ababb) machine will be accepted and show string accepted. Different animations will appear on accept of input strong. As shown in figure.

if the user enters wrong input string to the machine, the machine will halt and reject the input string. On rejection of string different animation will show as shown below figure.

### C. Models

Blender models I build which are army men, container, bullet, gun and stopper. Detail of each model is given below. The army men model, which is in army uniform like to kill the enemy in the project. We build an army using a blender, as shown in the below figure. Bullet model, build using a blender. The army men use this bullet model to kill the enemy by gun. fig.

The container model is used to block the enemy and the army men use container model to save his self. The container also uses to block the road. As shown below in the figure.

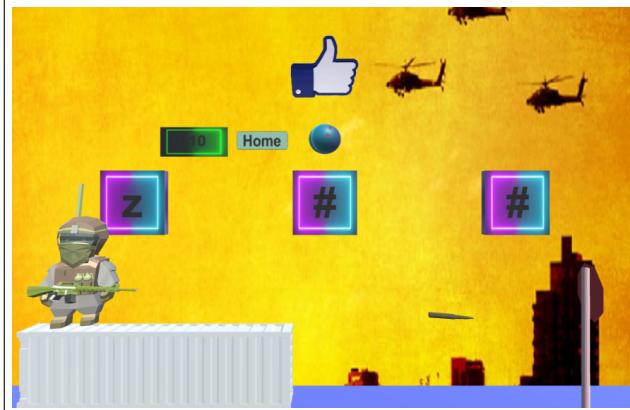


Fig. 41: Sajid machine Accept

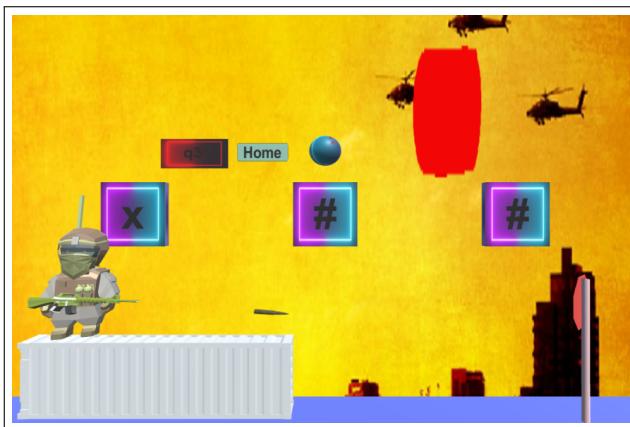


Fig. 42: Sajid machine Reject

Gun model used to kill the enemy. The army men get gun and fire to kill them. Army men can reload the gun to fire. As shown in the below figure.

Stop model use in a theme which we build using a blender, stop sign indicated the army men not to go ahead. Stop sign also use to stop the public not to go ahead. As shown below figure.

#### D. Video details

##### **YOUTUBE LINK**

<https://www.youtube.com/watch?v=AgVR33YXt9Y>

#### E. Video Description

This video consists of a unity scene with Turing Machine working on the back end. The language of this Turing Machine is  $A = \{a^n b^m a^n b^{n+m} | n = m \geq 1\}$ . The minimum string S accepted by Turing Machine is S= 'ababb'. The animations are self-designed and sounds are self-generated.

#### F. SOUNDS LINK

##### **Accept:**

[www.soundcloud.com/ammad-niazi-483497094/g15-sajid-accepted](http://www.soundcloud.com/ammad-niazi-483497094/g15-sajid-accepted)

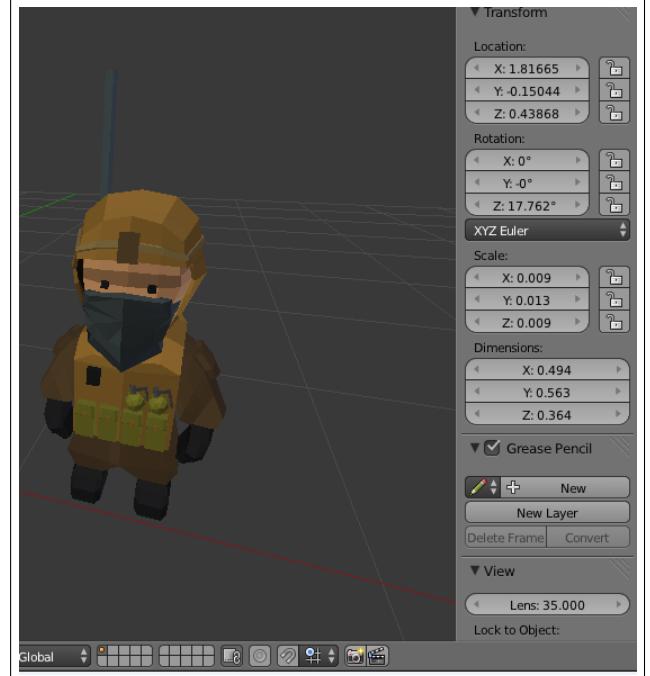


Fig. 43: Army Man Model

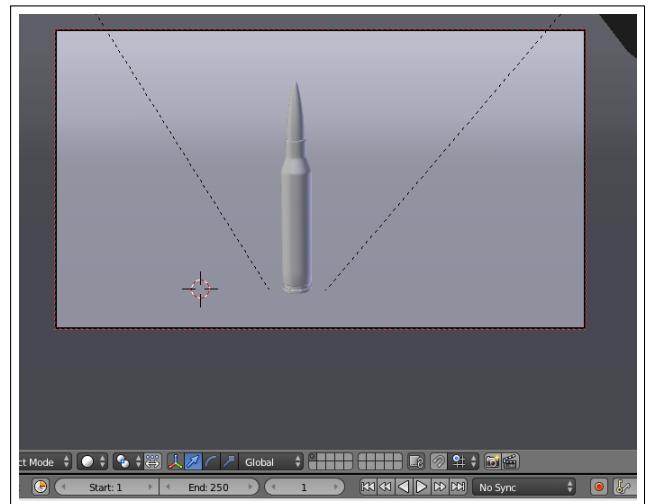


Fig. 44: Bullet Model

**Reject:** [soundcloud.com/ammad-niazi-483497094/g15\\_sajid\\_rejected](http://soundcloud.com/ammad-niazi-483497094/g15_sajid_rejected)

**Space:** [soundcloud.com/ammad-niazi-483497094/g15-sajid-space](http://soundcloud.com/ammad-niazi-483497094/g15-sajid-space)

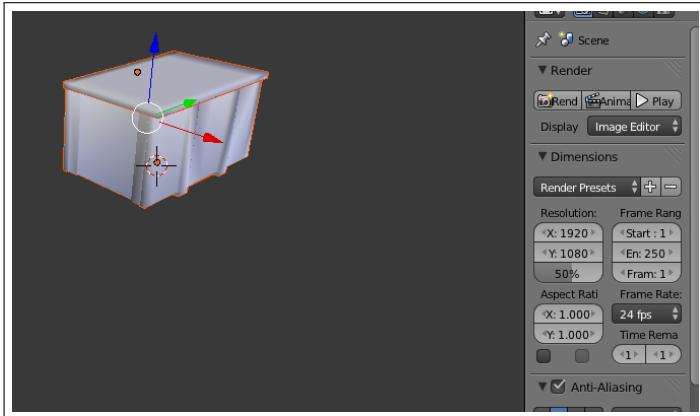


Fig. 45: Container Model

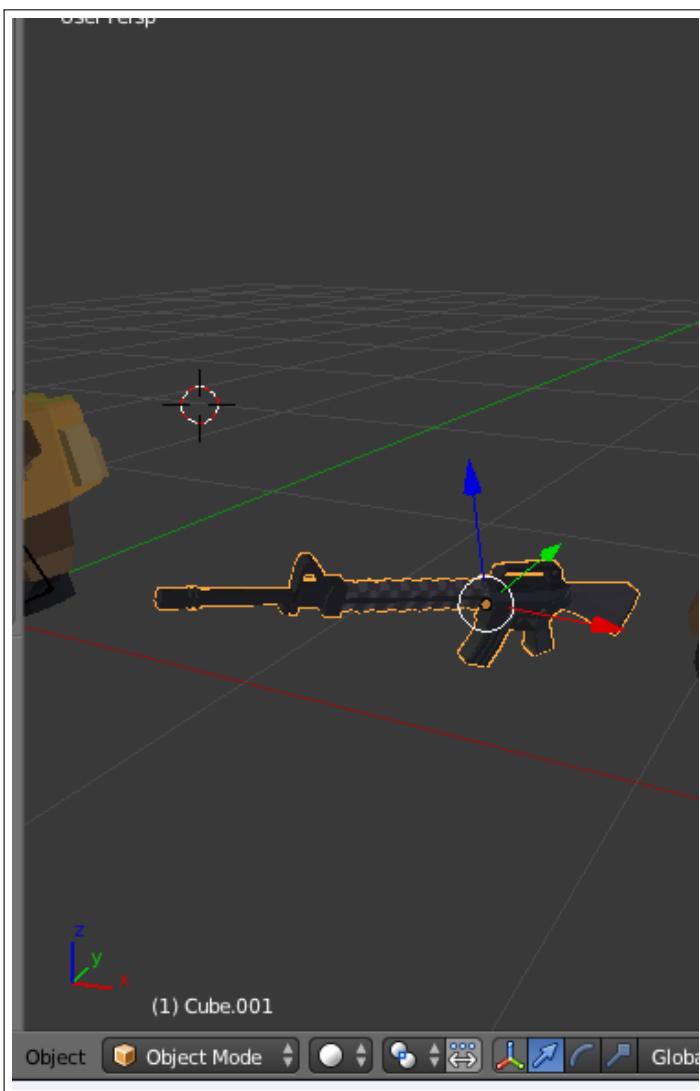


Fig. 46: Gun Model

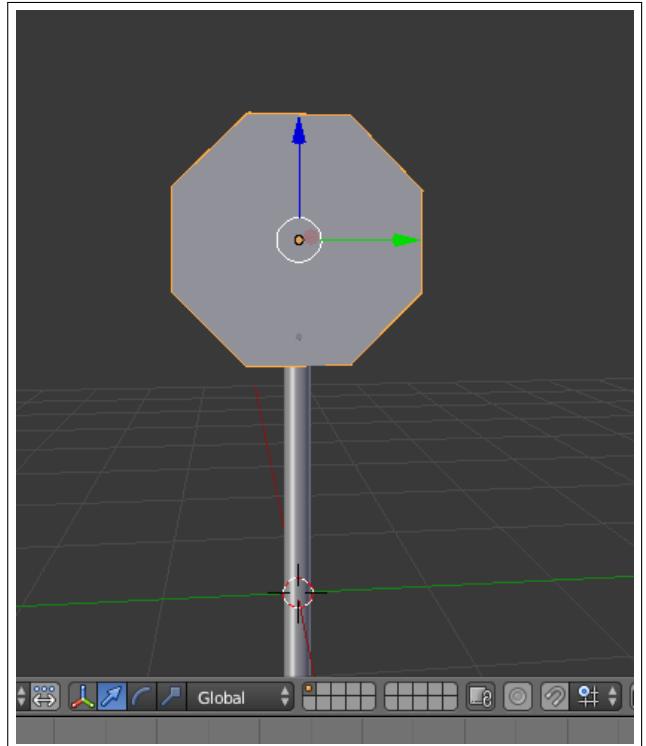


Fig. 47: Stop Model



Fig. 48: Youtube Video

## VII. PALINDROME

All palindrome for Binary Values.

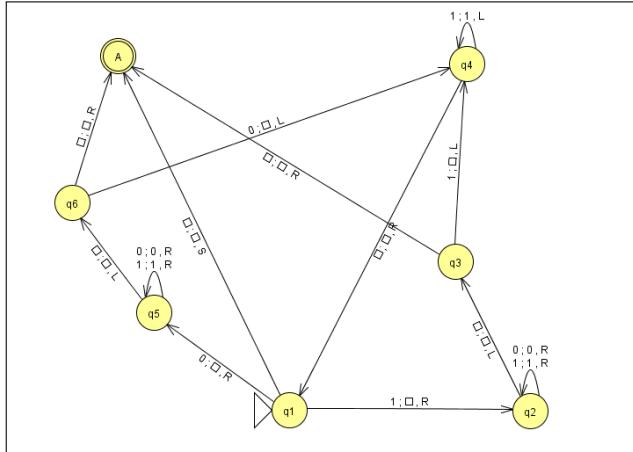


Fig. 49: Palindrome JFLAP

### A. Palindrome Simulation

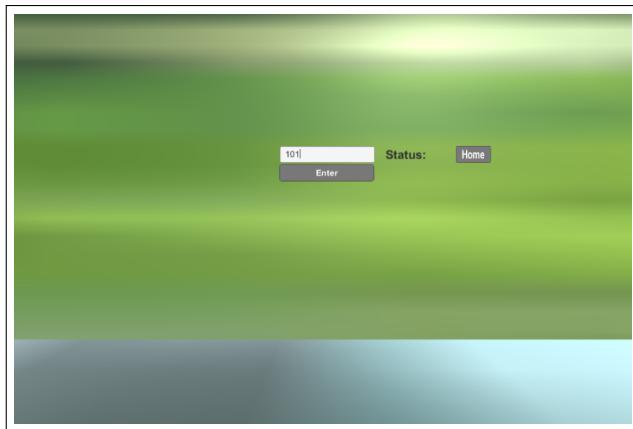


Fig. 50: Palindrome Main



Fig. 51: Palindrome Scene

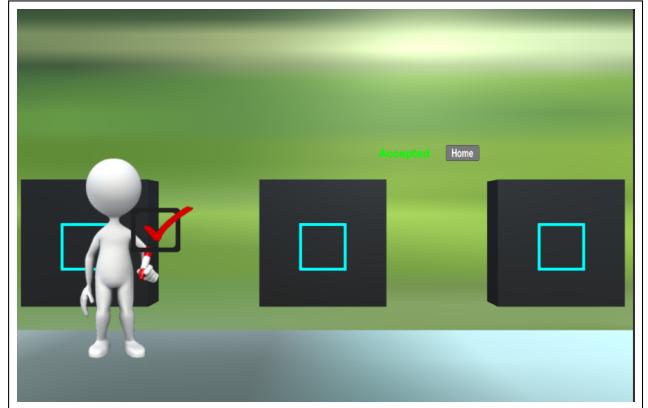


Fig. 52: Palindrome Accept Animation

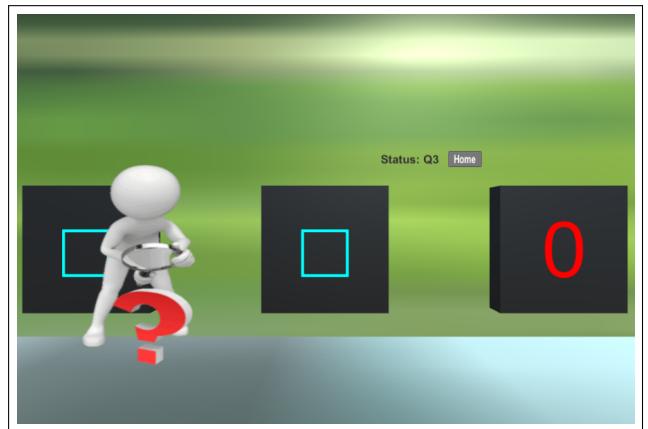


Fig. 53: Palindrome Reject Animation

### B. Youtube Video



Fig. 54: Palindrome Reject Animation

**Link:** <https://youtu.be/-yRG2FO2FUo>

### C. SOUNDS LINK

**Accept:** [www.soundcloud.com/hammad-niazi-483497094/g15-palindrome-accepted](http://www.soundcloud.com/hammad-niazi-483497094/g15-palindrome-accepted)

**Reject:** [www.soundcloud.com/hammad-niazi-483497094/g15-palindrome-reject](http://www.soundcloud.com/hammad-niazi-483497094/g15-palindrome-reject)

**Space:** [www.soundcloud.com/hammad-niazi-483497094/g15-palindrome-space](http://www.soundcloud.com/hammad-niazi-483497094/g15-palindrome-space)

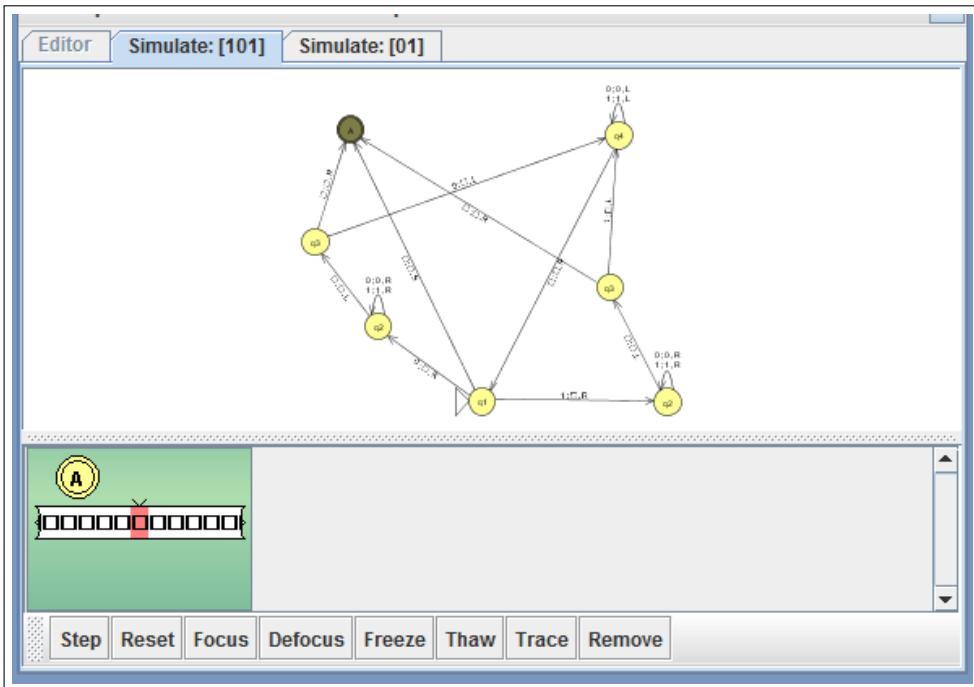


Fig. 55: Accept Simulation

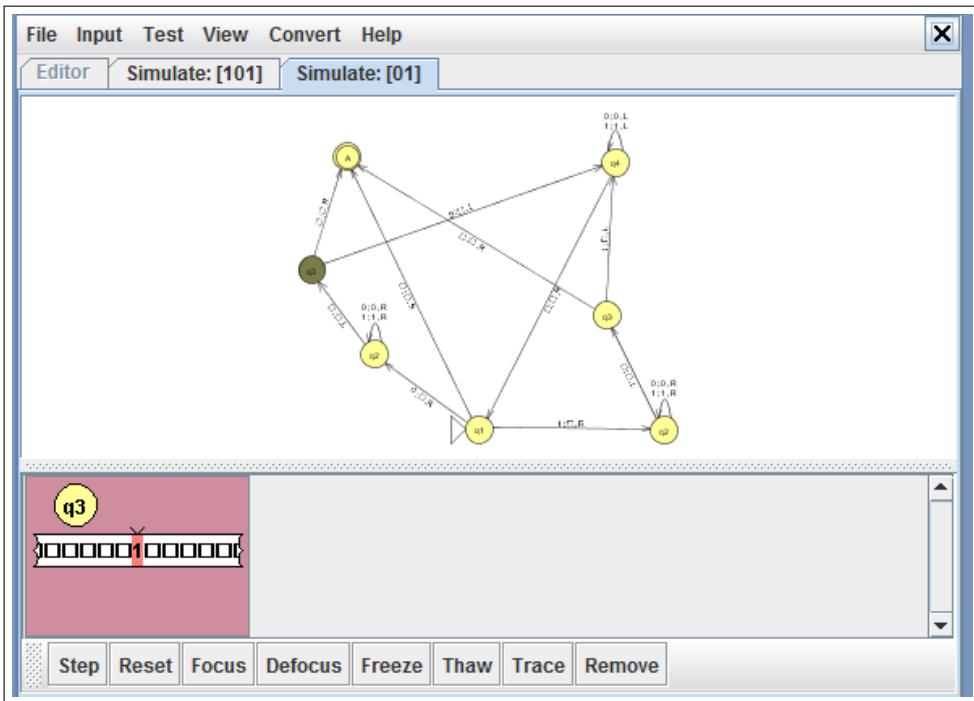


Fig. 56: Reject Simulation

### VIII. CODE SCREENSHOTS

```

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
using UnityEngine.SceneManagement;
using System.Text.RegularExpressions;
using UnityEditor;
using System;
using System.Linq;
public class G15_PalindromeWorkFile : MonoBehaviour
{
    // <summary>
    public Button HomeButton;
    public AudioSource SpaceButtonMusic;
    public AudioSource AcceptedMusic; public AudioSource RejectedMusic;
    public Image HideAcceptedImageWork;
    public Image HideRejectImageWork;
    public Text UserInputString;
    private string UserStringSave;
    private int FirstInputValue;
    private int LastInputValue;
    private int InputStringLength;
    private int StateVariable;
    public int NoOfCubes;
    public float PositionOfCube;
    string InputString;
    public float CamPositon;
    public string RayCastTxt;
    public Text LabelValue;
    private GameObject CreateCubes;
    private float InitialPosition;
    public Camera cameras;
    private float CameraMovePosition;
    public bool Flag;
    private int MoveNextVariable;
    private int MoveBackVariable;
    public GameObject ButtonOff;
    public GameObject InputOff;
    public GameObject TxtMeshTxt;
}

```

```

if (!RayCastTxt.Equals("Δ"))
{
    if (Flag == true)
    {
        if (MoveNextVariable == 1)
        {
            statevariable += 1;

            FirstInputValue = int.Parse(RayCastTxt);
            MoveBackVariable = 1;
            TxtMeshTxt.GetComponent<TextMeshPro>().text = "Δ";

            CamPosition += 1.5f;
            State();
            Forward();
            MoveNextVariable += 1;
        }
        else if (Flag == false)
        {
            if (MoveBackVariable == 1)
            {

                LastInputValue = int.Parse(RayCastTxt);
                if (FirstInputValue == LastInputValue)
                {
                    StateVariable += 1;
                    TxtMeshTxt.GetComponent<TextMeshPro>().text = "Δ";
                }
                else
                {
                    LabelValue.text = "Rejected";
                    //State();
                    LabelValue.color = Color.red;
                }
            }
        }
    }
}

```

```

public GameObject CubeMeshTxt;
public GameObject TxtMeshTxt;
void Start()
{
    Flag = true;
    InitialPosition = -1.5f;
    CameraMovePosition = 1.5f;
    MoveNextVariable = 1;
    MoveBackVariable = 1;
    CamPositon = 1.5f;
    StateVariable = 1;

    HomeButton.onClick.AddListener(homefunction);
}

private void homefunction()
{
    SceneManager.LoadScene("G15_MainMenu");
}

void Update()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        SpaceButtonMusic.Play();
        Ray myray = cameras.ViewportPointToRay(new Vector3(0.5f, 0.5f, 0.0f));
        RaycastHit hit;
        if (Physics.Raycast(myray, out hit, 100))
        {
            TxtMeshTxt = hit.transform.GetChild(0).gameObject;
            RayCastTxt = TxtMeshTxt.GetComponent<TextMeshPro>().text;
            Debug.DrawLine(myray.origin, hit.point, Color.red);

            NoOfCubes = InputStringLength / 2;
            PositionOfCube = NoOfCubes * 1.5f;
        }
    }
}

```

```

        LabelValue.text = "Accepted";
        HideAcceptedImageWork.gameObject.SetActive(true);
        LabelValue.color = Color.green;
        //TxtMeshTxt.GetComponent<TextMeshPro>().color = Color.green;
        SpaceButtonMusic.Stop();
        AcceptedMusic.Play();

        enabled = false;
    }
}
else
{
    if (Flag == true)
    {
        CamPositon += 1.5f;
        StateVariable += 1;
        State();
        Backward();
        Flag = false;
        if (MoveNextVariable == 1 && RayCastTxt.Equals("Δ") && StateVariable == 2)
        {
            LabelValue.text = "Accepted";
            HideAcceptedImageWork.gameObject.SetActive(true);
            LabelValue.color = Color.green;
            //TxtMeshTxt.GetComponent<TextMeshPro>().color = Color.green;
            SpaceButtonMusic.Stop();
            AcceptedMusic.Play();

            enabled = false;
        }
    }
}

```

```

    CamPositon += 1.5f;
    StateVariable = 1;
    State();
    Forward();
    Flag = true;
    if (MoveNextVariable == 2 && RayCastTxt.Equals("Δ"))
    {
        LabelValue.text = "Accepted";
        HideAcceptedImageWork.gameObject.SetActive(true);
        LabelValue.color = Color.green;
        //TxtMeshTxt.GetComponent<TextMeshPro>().color = Color.green;
        SpaceButtonMusic.Stop();
        AcceptedMusic.Play();
        enabled = false;
    }
}
}
2 references
public void Forward()
{
    CameraMovePosition += 1.5f;
    cameras.transform.position = new Vector3(CameraMovePosition, 0.6f, -1.6f);
}
references
public void Backward()
{
    CameraMovePosition -= 1.5f;
    cameras.transform.position = new Vector3(CameraMovePosition, 0.6f, -1.6f);
}

```

```

public void clickbutton()
{
    InputString = userInputString.text.Trim();
    Regex regex = new Regex("[^0-1]*$");
    if (!regex.IsMatch(InputString))
    {
        userInputString.text = " ";
        EditorUtility.DisplayDialog("Invalid", "Enter Only 0 or 1", "Thanks");
        return;
    }
    else
    /////
    ButtonOff.SetActive(false);
    InputOff.SetActive(false);

    UserStringSave = "Δ" + userInputString.GetComponent<Text>().text + "Δ";
    InputStringLength = UserStringSave.Length;

    int aa = 0;
    char[] Chararray = UserStringSave.ToCharArray();

    for (int i = 0; i < Chararray.Length; i++)
    {
        InitialPosition += 1.5f;
        //print(Chararray[i]);

        CreateCubes = GameObject.CreatePrimitive(PrimitiveType.Cube);
        CreateCubes.transform.position = new Vector3(InitialPosition, 0.5f, 0.0f);

        CreateCubes.transform.localScale = new Vector3(0.8f, 0.8f, 0.1f);
        CreateCubes.GetComponent<MeshRenderer>().material.color = Color.clear;
        GameObject go = new GameObject();
        go.transform.parent = CreateCubes.transform;
        CubeMeshPro = CreateCubes.transform.GetChild(0).gameObject;
    }
}

```

```

private GameObject cube;
private float positionCube;
private string currentState;
private List<char> word = new List<char>();
private int HeadPosition = 1;
private char symbol;
private string inputText;
private int MovementSpeed;
private GameObject CurrentCube;
//bool isAnimationstoped = true;
// bool IsMovementstoped = true;

0 references
void Start()
{
    positionCube = 0;
    Movementspeed = 14;
    symbol = 'Δ';
    HeadPosition = 1;
    currentState = "q1";
    // headAudiosource.clip = headMovementsound;
    // CubeRotationSoundSource.clip = CubeRotationSound;
    // AcceptedSoundSource.clip = Acceptedsound;
    // RejectedSoundSource.clip = Rejectedsound;

    //ButtonPlay.onClick.AddListener(ResetMachine);
    //ButtonPlay.onClick.AddListener(GetInput);

    //GetInput();
    InitializeMachine();
    HomeButton.onClick.AddListener(homefunction);
    BackButton.onClick.AddListener(backfunction);
}

1 reference
private void backfunction()
{
    SceneManager.LoadScene("G15_L1_MugheesMain");
}

```

```

    InitialPosition += 1.5f;
    //print(Chararray[i]);

    CreateCubes = GameObject.CreatePrimitive(PrimitiveType.Cube);
    CreateCubes.transform.position = new Vector3(InitialPosition, 0.5f, 0.0f);

    CreateCubes.transform.localScale = new Vector3(0.8f, 0.8f, 0.1f);
    CreateCubes.GetComponent<MeshRenderer>().material.color = color.clear;
    GameObject go = new GameObject();
    go.transform.parent = CreateCubes.transform;
    CubeMeshPro = CreateCubes.transform.GetChild(0).gameObject;
    CubeMeshPro.AddComponent<TextMeshPro>();
    CubeMeshPro.fontSize = 5;
    CubeMeshPro.alignment = TextAlignmentOptions.Center;
    CubeMeshPro.GetComponent<TextMeshPro>.text = Chararray[i].ToString();
    CubeMeshPro.GetComponent<TextMeshPro>.color = color.cyan;
    CubeMeshPro.transform.Rotate(0, 0, 0);
    Vector3 pos = CubeMeshPro.transform.localPosition;
    pos.x = 0;
    pos.y = 0;
    pos.z = -0.5f;
    CubeMeshPro.transform.localPosition = pos;
    aa = aa + 2;
}
State();
}

5 references
public void State()
{
    LabelValue.text = "Status: " + Statevariable;
}
}

```

```

1 reference
private void backfunction()
{
    SceneManager.LoadScene("G15_L1_MugheesMain");
}

1 reference
private void homefunction()
{
    SceneManager.LoadScene("G15_MainMenu");
}

1 reference
private void OnEnable()
{
    inputText = PlayerPrefs.GetString("input");
}

0 references
void GetInput()
{
    //validation textbox
    //inputText = userInputString.GetComponent<Text>().text;// get text from input field
    //print("value:" + inputText);

    //ButtonPlay.gameObject.SetActive(false);
    //hide input and button
    InitializeMachine();
}

2 references
private void InitializeMachine()
{
    //userInputString.gameObject.SetActive(false);
}

```

```

7 using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine.UI;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine;
using UnityEngine.SceneManagementManagement;
using UnityEngine;

0 references
public class G15_L1_MugheesMachine : MonoBehaviour
{
    //Wet and Dry //Audio
    public AudioSource SpaceButtonMusic; public AudioSource AcceptedMusic; public AudioSource RejectedMusic;
    public Image HideRejectedImage; public Image HideAcceptedImage;
    public Button HeadDisplay;
    public Button BackButton;
    public Button Head;
    //public GameObject userInputString;
    public GameObject TapHead;
    // public AudioClip headMovementSound;
    // public AudioClip headMovementSource;
    //public AudioClip cubeCreationSource;
    //public AudioClip CubeRotationSound;
    //public AudioClip AcceptedSound;
    //public AudioSource AcceptedSoundSource;
    //public AudioSource RejectedSoundSource;
    //public AudioClip RejectedSound;
    //public Button ButtonPlay;
    public GameObject HeadDisplay;
    public TextMeshPro CurrentstateText;

    private GameObject cube;
    private float positionCube;
    private string currentState;
    private List<char> word = new List<char>();
    private int HeadPosition = 1;
    private char symbol;
    private string inputText;
    private int MovementSpeed;
    private GameObject CurrentCube;
}

```

```

string input = symbol.ToString() + inputText + symbol.ToString() + symbol.ToString() + symbol.ToString();
CreateMachine(word);
CurrentstateText.GetComponent<TextMeshPro>().text = "State: " + currentState;

// public void ResetMachine()
// {
//     SceneManager.LoadScene("InputScene");
// }

1 reference
private List<char> ConvertToCharList(string input)
{
    List<char> list = new List<char>();
    foreach (char c in input)
    {
        list.Add(c);
    }
    return list;
}

1 reference
void CreateMachine(List<char> list)
{
    foreach (char c in list)
    {
        Addblock(c.ToString());
        GameObject.FindGameObjectWithTag("cube").SetActive(false);
    }
}

1 reference
void Addblock(string text)
{
    CreateCube(text, positionCube);
}

```

```

1 reference
void CreateCube(string _cubeText, float position_X)
{
    cube = Instantiate(gameObject.FindGameObjectWithTag("cube")) as GameObject;
    cube.transform.position = new Vector3(position_X, 0f, 0f);
    cube.transform.localScale = new Vector3(100f, 100f, 10f);
    cube.name = "CubeMachine";
    cube.tag = "CubeMachine";
    TextMeshPro cubeText = cube.GetComponentInChildren<TextMeshPro>();
    cubeText.text = _cubeText;
}

bool ishalt = false;
void update()
{
    if (Input.GetKeyDown(KeyCode.Space) && !ishalt)
    {
        SpaceButtonMusic.Play();
        //Movementstoed = false;
        //HeadbuttonMusic.Play();
        System.Threading.Thread.Sleep(200);
        ishalt = GetNextstate();
        CurrentStateText.GetComponent<TextMeshPro>().text = "State: " + Currentstate;
    }
}
//(
//(
    if (!isAnimationstoped)
        write();
}

```

```

private void UpdateHeadDisplay(Color screenColor, String displayText)
{
    HeadDisplay.GetComponent<Renderer>().material.SetColor("_Color", screenColor);
    CurrentStateText.GetComponent<TextMeshPro>().text = displayText;
    //ButtonPlay.gameObject.SetActive(true);
}

1 reference
public void write()
{
    int currentPosition = Convert.ToInt32(TapeHead.transform.position.X);
    int newPosition = Convert.ToInt32(HeadPosition * 400);
    if (currentPosition < newPosition)
    {
        currentPosition += MovementSpeed;
        TapeHead.transform.position = new Vector3(currentPosition, 100, 0);
        if (currentPosition + MovementSpeed >= newPosition)
        {
            TapeHead.transform.position = new Vector3(newPosition, 100, 0);
        }
    }
    else if (currentPosition > newPosition)
    {
        currentPosition -= MovementSpeed;
        TapeHead.transform.position = new Vector3(currentPosition, 100, 0);
        if (newPosition > currentPosition - MovementSpeed)
        {
            TapeHead.transform.position = new Vector3(newPosition, 100, 0);
        }
    }
    else
    {
        print("hhh");
        headaudioSource.Stop();
        ISMovementstoed = true;
    }
}

```

```

J

8 references
private char getCurrentChar()
{
    return word[Headposition];
}

3 references
private void addExtraCube(int position_X, string text)
{
    cube = Instantiate(gameObject.FindGameObjectWithTag("cubeMachine")) as GameObject;
    cube.transform.position = new Vector3(position_X, 0f, 0f);
    cube.transform.localScale = new Vector3(100f, 100f, 10f);
    cube.name = "CubeMachine" + text;
    cube.tag = "CubeMachine";
    TextMeshPro cubeText = cube.GetComponentInChildren<TextMeshPro>();
    cubeText.text = text;
}

1 reference
public bool GetNextstate()
{
    switch (CurrentState)
    {
        case "q1":
            switch (getCurrentchar())
            {
                case 'a':
                    PerformTransaction('x', Movement.R, "q2");
                    break;

                case 'y':
                    PerformTransaction('y', Movement.R, "q6");
                    break;
            }

```

```

3 reference
public bool GetNextstate()
{
    switch (CurrentState)
    {
        case "q1":
            switch (getCurrentchar())
            {
                case 'a':
                    PerformTransaction('x', Movement.R, "q2");
                    break;

                case 'y':
                    PerformTransaction('y', Movement.R, "q6");
                    break;

                default:
                    HideRejectImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");
                    SpaceButtonMusic.Stop();
                    RejectedMusic.Play();
                    //RejectedSoundSource.Play();
                    return true;
            }
        break;

        case "q2":
            switch (getCurrentchar())
            {
                case 'a':
                    PerformTransaction('a', Movement.R, "q2");
                    break;

                case 'y':
                    PerformTransaction('y', Movement.R, "q2");
                    break;
            }

```

```

break;

case "q2":
    switch (getCurrentchar())
    {
        case 'a':
            PerformTransaction('a', Movement.R, "q2");
            break;

        case 'y':
            PerformTransaction('y', Movement.R, "q2");
            break;

        case 'b':
            PerformTransaction('y', Movement.R, "q3");
            break;

        default:
            HideRejectImageWork.gameObject.SetActive(true);
            UpdateHeadDisplay(Color.red, "Rejected");
            SpaceButtonMusic.Stop();
            RejectedMusic.Play();
            //RejectedSoundSource.Play();
            return true;
    }
}
break;

case "q3":
    switch (getCurrentchar())
    {
        case 'k':
            PerformTransaction('y', Movement.R, "q4");
            break;

        case 'b':
            PerformTransaction('y', Movement.R, "q4");
            break;
    }

```

```

break;

case "q4":
    switch (getCurrentchar())
    {
        case 'a':
            PerformTransaction('z', Movement.L, "q5");
            break;
        case 'z':
            PerformTransaction('z', Movement.R, "q4");
            break;

        case 'b':
            PerformTransaction('b', Movement.R, "q4");
            break;

        default:
            HideRejectImageWork.gameObject.SetActive(true);
            UpdateHeadDisplay(Color.red, "Rejected");
            SpaceButtonMusic.Stop();
            RejectedMusic.Play();
            //RejectedSoundSource.Play();
            return true;
    }
}
break;

case "q5":
    switch (getCurrentchar())
    {
        case 'z':
            PerformTransaction('z', Movement.L, "q5");
            break;

        case 'a':
            PerformTransaction('a', Movement.L, "q5");
            break;

        case 'y':
            PerformTransaction('y', Movement.L, "q5");
            break;
    }

```

```

        case "q5":
            switch (getCurrentChar())
            {
                case 'z':
                    PerformTransaction('z', Movement.L, "q5");
                    break;

                case 'a':
                    PerformTransaction('a', Movement.L, "q5");
                    break;

                case 'y':
                    PerformTransaction('y', Movement.L, "q5");
                    break;

                case 'b':
                    PerformTransaction('b', Movement.L, "q5");
                    break;

                case 'x':
                    PerformTransaction('x', Movement.R, "q1");
                    break;
                ///////////////
                default:
                    HideRejectImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");
                    SpaceButtonMusic.Stop();
                    RejectedMusic.Play();
                    //RejectedSoundSource.Play();
                    return true;
            }
            break;

        case "q6":
            switch (getCurrentChar())
            {
                case 'y':
                    PerformTransaction('y', Movement.R, "q6");
                    break;
            }
    }
}

```

```

    if (Physics.Raycast(ray, out hit, Mathf.Infinity))
    {
        CurrentCube = hit.collider.gameObject;
        if (Word[HeadPosition] != tempChar)
        {
            // CurrentCube.GetComponent<Animator>().Play("WriteCube");
            //CubeRotationSoundSource.Play();
            CurrentCube.GetComponentInChildren<TextMeshPro>().text = replaceChar.ToString();
        }
        HeadPosition += (int)movement;
        if (HeadPosition <= 0)
        {
            addExtracube(HeadPosition * 400 - 400, symbol.ToString());
            Word.Insert(0, symbol);
            addExtracube(HeadPosition * 400 - 400, symbol.ToString());
        }
        else if (HeadPosition >= Word.Count - 2)
        {
            addExtracube(HeadPosition * 400 + 400, symbol.ToString());
            Word.Add(symbol);
        }
        currentState = nextState;
    }
}

```

```

        uiBox,
        case "q6":
            switch (getCurrentChar())
            {
                case 'y':
                    PerformTransaction('y', Movement.R, "q6");
                    break;

                case 'z':
                    PerformTransaction('z', Movement.R, "q7");
                    break;

                default:
                    HideRejectImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");
                    SpaceButtonMusic.Stop();
                    RejectedMusic.Play();
                    //RejectedSoundSource.Play();
                    return true;
            }
            break;

        case "q7":
            switch (getCurrentChar())
            {
                case 'z':
                    PerformTransaction('z', Movement.R, "q7");
                    break;

                case 'a':
                    PerformTransaction('a', Movement.H, "q8");
                    break;

                default:
                    HideRejectImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");
                    SpaceButtonMusic.Stop();
                    RejectedMusic.Play();
                    //RejectedSoundSource.Play();
                    return true;
            }
            break;
    }
}

```

```

1 reference
public bool GetNextState()
{
    switch (currentState)
    {
        case "q1":
            switch (getCurrentChar())
            {
                case 'x':
                    PerformTransaction('a', Movement.R, "q2");
                    break;

                case 'a':
                    PerformTransaction('a', Movement.R, "q2");
                    break;

                case 'b':
                    PerformTransaction('b', Movement.R, "q6");
                    break;

                default:
                    HideRejectedImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");
                    //RejectedSoundSource.Play();
                    SpaceButtonMusic.Stop();
                    RejectedMusic.Play();
                    return true;
            }
            break;

        case "q2":
            switch (getCurrentChar())
            {
                case 'u':

```

```

                    RejectedMusic.Play();
                    //RejectedSoundSource.Play();
                    return true;
            }
            break;

        case "q6":
            HideRejectedImageWork.gameObject.SetActive(true);
            UpdateHeadDisplay(Color.green, "Accepted");
            SpaceButtonMusic.Stop();
            AcceptedMusic.Play();
            //AcceptedSoundSource.Play();
            return true;
    }
    return false;
}

13 references
private void PerformTransaction(char replaceChar, Movement movement, string nextState)
{
    char tempChar = Word[HeadPosition];
    Word[HeadPosition] = replaceChar;
    Ray ray = new Ray(new Vector3(transform.position.x, transform.position.y - 2, transform.position.z), Vector3.forward); //trying to single select
    RaycastHit hit;
    if (Physics.Raycast(ray, out hit, Mathf.Infinity))
    {
        CurrentCube = hit.collider.gameObject;
        if (Word[HeadPosition] != tempChar)
        {
            // CurrentCube.GetComponent<Animator>().Play("WriteCube");
            //CubeRotationSoundSource.Play();
            CurrentCube.GetComponentInChildren<TextMeshPro>().text = replaceChar.ToString();
        }
    }
}

```

```

        break;

        case "q2":
            switch (getCurrentChar())
            {
                case 'y':
                    PerformTransaction('b', Movement.R, "q3");
                    break;

                case 'b':
                    PerformTransaction('b', Movement.R, "q2");
                    break;

                case 'x':
                    PerformTransaction('x', Movement.R, "q2");
                    break;

                default:
                    HideRejectedImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");
                    SpaceButtonMusic.Stop();
                    RejectedMusic.Play();
                    //RejectedSoundSource.Play();
                    return true;
            }
            break;

        case "q3":
            switch (getCurrentChar())
            {
                case 'x':
                    PerformTransaction('c', Movement.R, "q4");
                    break;
            }
    }
}

```

```

        }
        break;

    case "q3":
        switch (getCurrentChar())
        {
            case 'x':
                PerformTransaction('c', Movement.R, "q4");
                break;

            case 'y':
                PerformTransaction('y', Movement.R, "q3");
                break;

            case 'c':
                PerformTransaction('c', Movement.R, "q3");
                break;

            default:
                HideRejectedImageWork.gameObject.SetActive(true);
                UpdateHeadDisplay(Color.red, "Rejected");
                SpaceButtonMusic.Stop();
                RejectedMusic.Play();

                //RejectedSoundSource.Play();
                return true;
        }
        break;

    case "q4":
        switch (getCurrentChar())
        {
            case 'x':
                PerformTransaction('c', Movement.L, "q5");
                break;

```

```

        }

        case 's':
            PerformTransaction('s', Movement.H, "q8");
            break;

        default:
            HideAcceptedImageWork.gameObject.SetActive(true);
            UpdateHeadDisplay(Color.red, "Rejected");
            SpaceButtonMusic.Stop();
            RejectedMusic.Play();

            //RejectedSoundSource.Play();
            return true;
    }
    break;

    case "q8":
        HideAcceptedImageWork.gameObject.SetActive(true);
        getCurrentChar();
        UpdateHeadDisplay(Color.green, " Accepted");
        SpaceButtonMusic.Stop();
        AcceptedMusic.Play();

        //AcceptedSoundSource.Play();
        return true;

```

```

        }

        case 'q5':
            switch (getCurrentChar())
            {
                case 'a':
                    PerformTransaction('a', Movement.R, "q1");
                    break;

                case 'c':
                    PerformTransaction('c', Movement.L, "q5");
                    break;

                case 'b':
                    Performtransaction('b', Movement.L, "q5");
                    break;

                case 'y':
                    PerformTransaction('y', Movement.L, "q5");
                    break;

                case 'x':
                    PerformTransaction('x', Movement.L, "q5");
                    break;

                default:
                    HideRejectedImageWork.gameObject.SetActive(true);
                    UpdateHeadDisplay(Color.red, "Rejected");

```

```

        }

        public bool GetNextState()
        {

            switch (currentState)
            {
                case "q1":
                    switch (getCurrentChar())
                    {
                        case 'a':
                            PerformTransaction('w', Movement.R, "q2");
                            break;

                        case 'x':
                            PerformTransaction('x', Movement.R, "q7");
                            break;

                        default:
                            HideRejectedImageWork.gameObject.SetActive(true);
                            UpdateHeadDisplay(Color.red, "Reject");
                            SpaceButtonMusic.Stop();
                            RejectedMusic.Play();

                            // RejectedSoundSource.Play();
                            return true;
                    }
                    break;

                case "q2":
                    switch (getCurrentChar())
                    {
                        case 'a':
                            PerformTransaction('a', Movement.R, "q2");
                            break;

                        case 'x':

```

```

                            PerformTransaction('x', Movement.R, "q2");
                            break;

                        default:
                            HideRejectedImageWork.gameObject.SetActive(true);
                            UpdateHeadDisplay(Color.red, "Rejected");
                            SpaceButtonMusic.Stop();
                            RejectedMusic.Play();

                            //RejectedSoundSource.Play();
                            return true;
                    }
                    break;

                case "q6":
                    switch (getCurrentChar())
                    {
                        case 'c':
                            PerformTransaction('c', Movement.R, "q7");
                            break;

                        case 'b':
                            PerformTransaction('b', Movement.R, "q6");
                            break;

                        default:
                            HideRejectedImageWork.gameObject.SetActive(true);
                            UpdateHeadDisplay(Color.red, "Rejected");
                            SpaceButtonMusic.Stop();
                            RejectedMusic.Play();

                            //RejectedSoundSource.Play();
                            return true;
                    }
                    break;

                case "q7":
                    switch (getCurrentChar())
                    {
                        case 'c':
                            PerformTransaction('c', Movement.R, "q7");
                            break;

```

```

                        case 'q2':
                            switch (getCurrentChar())
                            {
                                case 'a':
                                    PerformTransaction('a', Movement.R, "q2");
                                    break;

                                case 'x':
                                    PerformTransaction('x', Movement.R, "q2");
                                    break;

                                case 'b':
                                    PerformTransaction('x', Movement.R, "q3");
                                    break;

                                default:
                                    HideRejectedImageWork.gameObject.SetActive(true);
                                    UpdateHeadDisplay(Color.red, "Reject");
                                    SpaceButtonMusic.Stop();
                                    RejectedMusic.Play();

                                    //RejectedSoundSource.Play();
                                    return true;
                            }
                            break;

                        case "q3":
                            switch (getCurrentChar())
                            {
                                case 'b':
                                    PerformTransaction('b', Movement.R, "q3");
                                    break;

                                case 'y':
                                    PerformTransaction('y', Movement.R, "q3");
                                    break;

```

```

    {
        case 'b':
            PerformTransaction('b', Movement.R, "q3");
            break;

        case 'y':
            PerformTransaction('y', Movement.R, "q3");
            break;

        case 'a':
            PerformTransaction('y', Movement.R, "q4");
            break;

        default:
            HideRejectedImageWork.gameObject.SetActive(true);
            UpdateHeadDisplay(Color.red, "Reject");
            SpaceButtonMusic.Stop();
            RejectedMusic.Play();

            //RejectedSoundSource.Play();
            return true;
    }
}

case "q4":
    switch (getCurrentChar())
    {
        case 'a':
            PerformTransaction('a', Movement.R, "q4");
            break;

        case 'z':
            PerformTransaction('z', Movement.R, "q4");
            break;
        case 'b':
    }

    RejectedMusic.Play();

    //RejectedSoundSource.Play();
    return true;
}
break;

case "q5":
switch (getCurrentChar())
{
    case 'b':
        PerformTransaction('z', Movement.L, "q5");
        break;

    default:
        HideRejectedImageWork.gameObject.SetActive(true);
        UpdateHeadDisplay(Color.red, "Reject");
        SpaceButtonMusic.Stop();
        RejectedMusic.Play();

        //RejectedSoundSource.Play();
        return true;
    }
break;

case "q6":
switch (getCurrentChar())
{
    case 'z':
        PerformTransaction('z', Movement.L, "q6");
        break;

    case 'b':
        PerformTransaction('b', Movement.L, "q6");
        break;
}

return true;
}
break;

case "q6":
switch (getCurrentChar())
{
    case 'z':
        PerformTransaction('z', Movement.L, "q6");
        break;

    case 'b':
        PerformTransaction('b', Movement.L, "q6");
        break;

    case 'a':
        PerformTransaction('a', Movement.L, "q6");
        break;

    case 'x':
        PerformTransaction('x', Movement.L, "q6");
        break;

    case 'y':
        PerformTransaction('y', Movement.L, "q6");
        break;

    case 'w':
        PerformTransaction('w', Movement.R, "q1");
        break;

    default:
        HideRejectedImageWork.gameObject.SetActive(true);
        UpdateHeadDisplay(Color.red, "Reject");
        SpaceButtonMusic.Stop();
        RejectedMusic.Play();

        //RejectedSoundSource.Play();
        return true;
}
break;

case "q7":
switch (getCurrentChar())
{
    case 'x':
        PerformTransaction('x', Movement.R, "q7");
        break;

    case 'y':
        PerformTransaction('y', Movement.R, "q8");
        break;

    default:
        HideRejectedImageWork.gameObject.SetActive(true);
        UpdateHeadDisplay(Color.red, "Reject");
        SpaceButtonMusic.Stop();
        RejectedMusic.Play();

        //RejectedSoundSource.Play();
        return true;
}
break;

case "q8":
switch (getCurrentChar())
{
    case 'y':
        PerformTransaction('y', Movement.R, "q8");
        break;

    case 'z':
        PerformTransaction('z', Movement.R, "q9");
        break;
}

case "q9":
switch (getCurrentChar())
{
    case 'z':
        PerformTransaction('z', Movement.R, "q9");
        break;

    case '#':
        PerformTransaction('#', Movement.H, "q10");
        break;

    default:
        HideRejectedImageWork.gameObject.SetActive(true);
        UpdateHeadDisplay(Color.red, "Reject");
        SpaceButtonMusic.Stop();
        RejectedMusic.Play();

        //RejectedSoundSource.Play();
        return true;
}
break;

case "q10":
HideAcceptedImageWork.gameObject.SetActive(true);
getCurrentChar();
UpdateHeadDisplay(Color.green, "Accept");
}

default:
break;
}
return false;
}

```