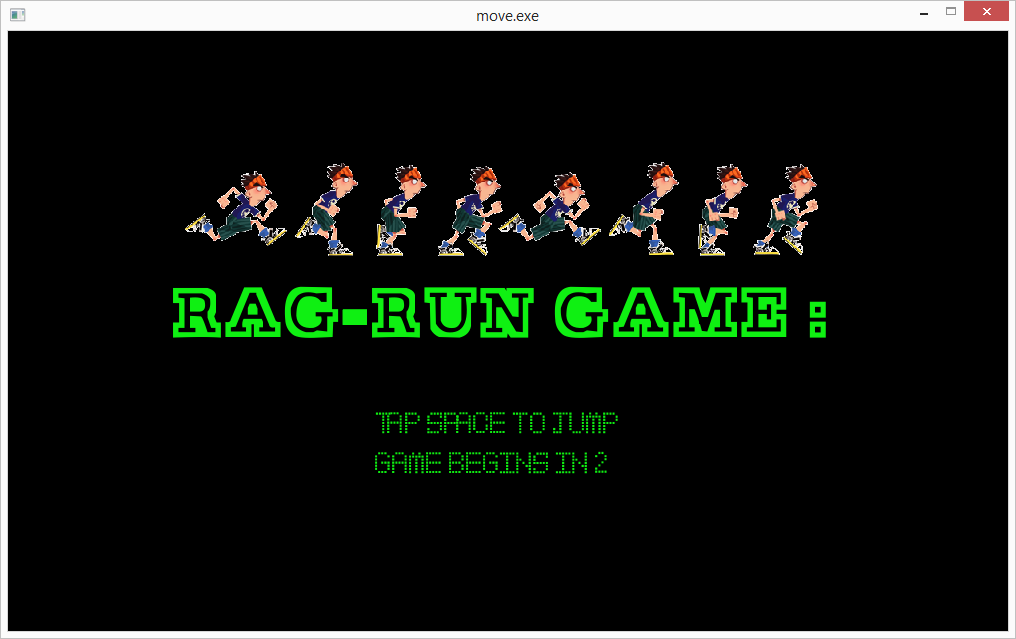
# CS103 – Computer Programming – Spring 2015

**Semester Project**

**RAG-RUN**



|  |  |
| --- | --- |
| **Developed by:** |  |
|  | **Sohaib Hashmi – Section A – 14k-2121**  **Talha Asif – Section A – 14k-2246**  **Hamza Masood – Section A – 12k-2198** |
| **Supervised by:** |  |
|  | **Ms. Fareha Sheikh** |

**Department of Computer Science**

**National University of Computer & Emerging Sciences (NU-FAST)**

**Table of Content**

|  |  |
| --- | --- |
| Introduction | 3 |
| Need for the Project | 3 |
| Uniqueness of the Project | 3 |
| Implemented Classes | 3-4 |
| UML Class Diagram | 4-6 |
| Header Files | 7 |
| Algorithm | 7-9 |
| Flow Chart | 9-11 |
| Screen Shots  Source code | 11-14  15 |
| Future Enhancements | 15 |
| References | 15 |

**INTRODUCTION:**

This application is a game called RAG-RUN brought to you by TMH Productions. It’s a simple game with a simple idea built for entertainment purpose only. This game helps the user to relax him/herself during his busy routine as it is played with a SINGLE KEY!

**NEED FOR THE PROJECT:**

As gaming industry is getting better and better and more games are being made on 3D platforms, the classical world and mighty galaxy of 2D games is being slowly sidelined. But the world still needs to know the beginning of the gaming world and therefore it still needs to play 2D games, which requires new 2D games to be added in this mighty galaxy and here we are, adding our small black hole into this galaxy. Our game is made for entertainment purpose and to provide relaxation to its users.

Furthermore new 3D games require more powerful PCs or consoles which requires a huge spending which is not possible for all of us. Install our modern yet simple game which is free of cost (till now at least :P) and keep yourself and your children happy.

**UNIQUENESS OF THE PROJECT:**

Every game bring some uniqueness with it and here is our game based on the nightmare of every freshmen student i.e. RAGGING. Never done before, our game is based on the idea of ragging and as the students of FAST we have added quite some flavor in it. Our game enables the user to run away from the senior as far as he can.

**IMPLEMENTED CLASSES:**

* Dimension: Its purpose is to pass height and width of the game to other classes and functions.
* Credits: This class is used to display the credits at the end of the game.
* Mainpage: The purpose of this class is to display the title and instruction of the game before the start of it.
* Background: This is class is made to display our background and make our parallax background work.
* Comet: This class is very important regarding our game. It draws, starts and updates the position of the hurdles in our game.
* Collide: This class serves the purpose of checking the collision of user with the hurdles.

**UML CLASS DIAGRAM:**

Playgame

Rungame() : void

Mainpage

+ background() : void

+ select() : void

linked

linked

linked

Highscore

-Score : int

-Highcore : int

+ showscore() : void

+ save(int ) : void

Exit

Displayexit() : void

Exit() : void

Map

+ backmap() : void

+ foremap() : void

User

# user : speed

+Displayuser() : void

+Moveuser() : void

+Jumpuser(int) : void

Hurdles

+ Box() : void

+car() : void

+ banana() : void

Computer

#computer

+ displaycomp() : void

+ movecomp() : void

+ jumpcomp(int) : void

Speed

# Time : double

# spd\_user : int

# spd\_comp : int

+ spduser(int time ,int spd\_user) : int

+ spdcomp(int time, int spd\_comp) : int

Aggregation to speed

inheritance to highscore

Score

# scr : speed

-Count : int

Setscore(int scr) : void

Getscore() : int

savescr(int gs) : void

dispplayscr() : void

**HEADER FILES:**

1. allegro5\allegro.h to include allegro5 file
2. allegro5\allegro\_image.h to include allegro image file
3. allegro5\allegro\_font.h for fonts to work on allegro
4. allegro5\allegro\_ttf.h to include ttf type fonts
5. allegro5\allegro\_primitives.h graphics primitives which are used in game

**ALGORITHM:**

Well there are a lot of algorithms being used in our game for different purposes but our team is not going to reveal all of them, coz a good chef never reveals the secret ingredient of his special recipe :P.

1. The main algorithm in our game is of the hurdles.

* DrawComet() is the function which draws hurdles in our game using allegro graphics

void Comet::DrawComet(Comet comets[], int size)

{

for(int i = 0; i < size; i++)

{

if(comets[i].live)

{

al\_draw\_rectangle(comets[i].x,comets[i].y,comets[i].x+50,comets[i].y+50,al\_ map\_rgb(210,105,30),4);

al\_draw\_filled\_rectangle(comets[i].x+3,comets[i].y+3,comets[i].x+47,comets[i].y+47,al\_map\_rgb(139,69,19));

}

}

}

* Then comes our function of updating our comet which determines the position of our hurdles.

void Comet::UpdateComet(Comet comets[], int size)

{

for(int i = 0; i < size; i++)

{

if(comets[i].live)

{

comets[i].x -= comets[i].speed;

if(comets[i].x < 0)

comets[i].live = false;

}

}

}

2. Another important part of our algorithm is to check the collision of the user with hurdles.

* CollideImage() is the function which checks whether our user has collided with the image or not.

void Collide::CollideImage(Comet comets[],int cSize,ALLEGRO\_BITMAP \*image2[],int count,ALLEGRO\_DISPLAY \*display)

{

for(int i = 0; i < cSize; i++)

{

if(comets[i].live)

{

if(((comets[i].x-comets[i].boundx < 195 && comets[i].x-comets[i].boundx > 180) && HEIGHT - y < 250 ) ||  (comets[i].x-comets[i].boundx < 195 && comets[i].x-comets[i].boundx > 180) && y > 440)

{

comets[i].live = false;

al\_rest(2.0);

al\_clear\_to\_color(al\_map\_rgb(0,0,0));

al\_init\_font\_addon();

al\_init\_ttf\_addon();

ALLEGRO\_FONT \*fontover = al\_load\_font("mypager.ttf", 80, 0);

ALLEGRO\_FONT \*fontclose = al\_load\_font("mypager.ttf", 30, 0);

al\_draw\_textf(fontover, al\_map\_rgb(255, 0, 255), WIDTH/4 , 270, 0, "GAME OVER");

al\_draw\_textf(fontclose, al\_map\_rgb(255, 0, 255), WIDTH/3 , 380, 0, "YOUR SCORE : %i ", (count/6)-1);

al\_flip\_display();

al\_rest(3.0);

al\_clear\_to\_color(al\_map\_rgb(0,0,0));

credits end;

end.showcredit(display);

al\_rest(2.0);

exit(0);

}

}

}

}

This game is also responsible to display the game over screen if collision has occurred.

3. Another important algorithm in our game is of making our user jump. It is done in accordance with the timer of our allegro.

if(al\_key\_down(&keystate, ALLEGRO\_KEY\_SPACE) && jump)

{

vely = -jumpspeed;

jump = false;

}

if(!jump)

vely += gravity;

else

vely = 0;

X += velx;

Y += vely;

Collide col(X,Y);

col.CollideImage(comets,NUM\_COMETS,image2,count,display);

jump = (Y >= 415 );

if(jump)

Y = 415 ;

**FLOWCHART:**

START

TITLE PAGE

IF

TIME > 5 SEC

NO

YES

User Starts

Background starts

Show background and user.

IF USER PRESS SPACE

NO

YES

IF COLLISSION = TRUE

USER JUMPS

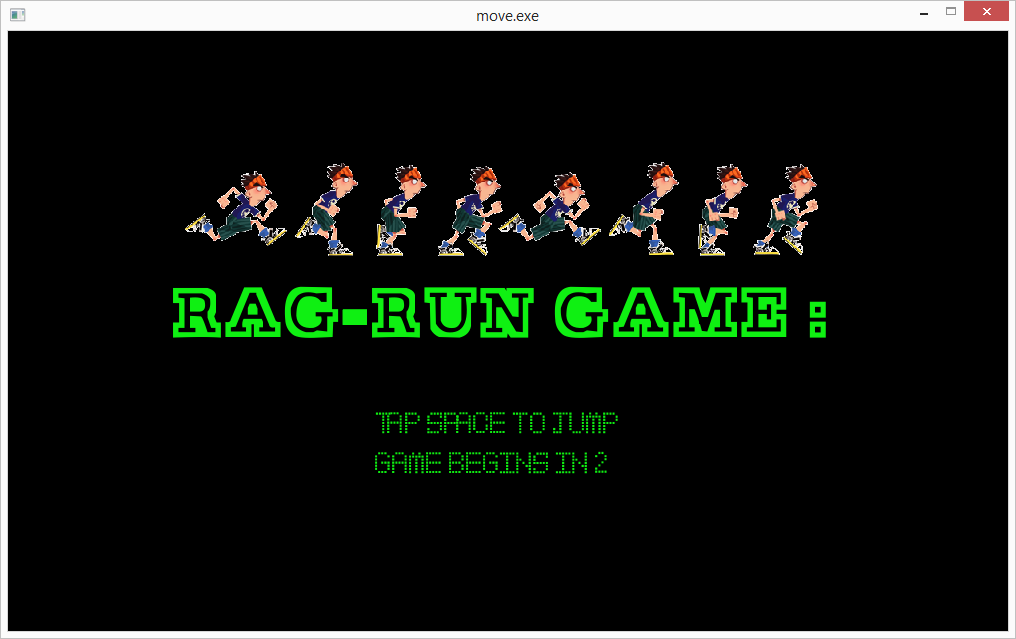
NO

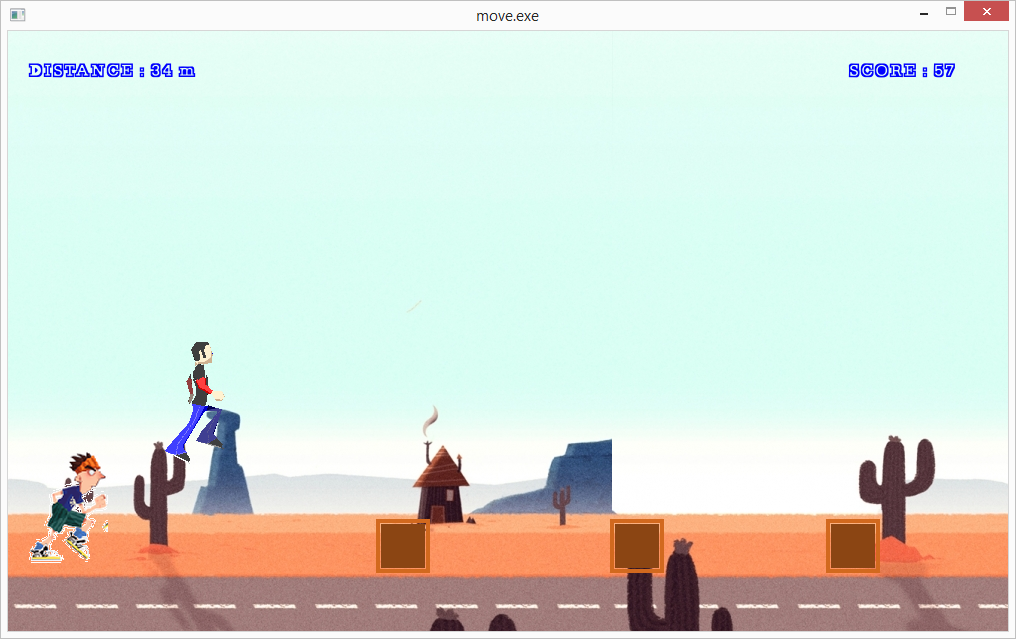
YES

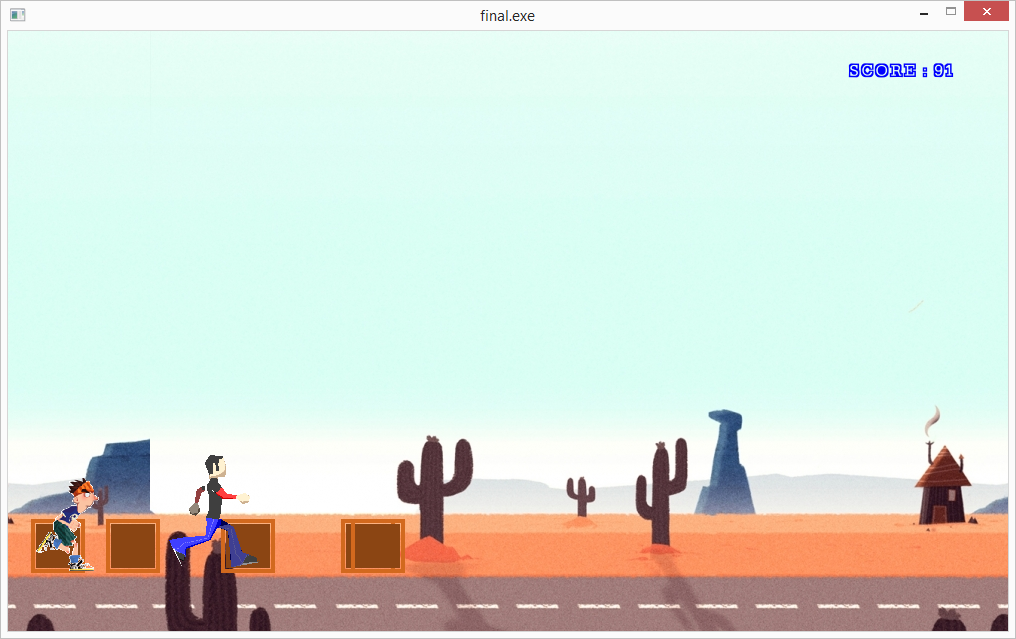
PAUSE SCREEN

SHOW GAMEOVER SCREEN, SHOW CREDITS

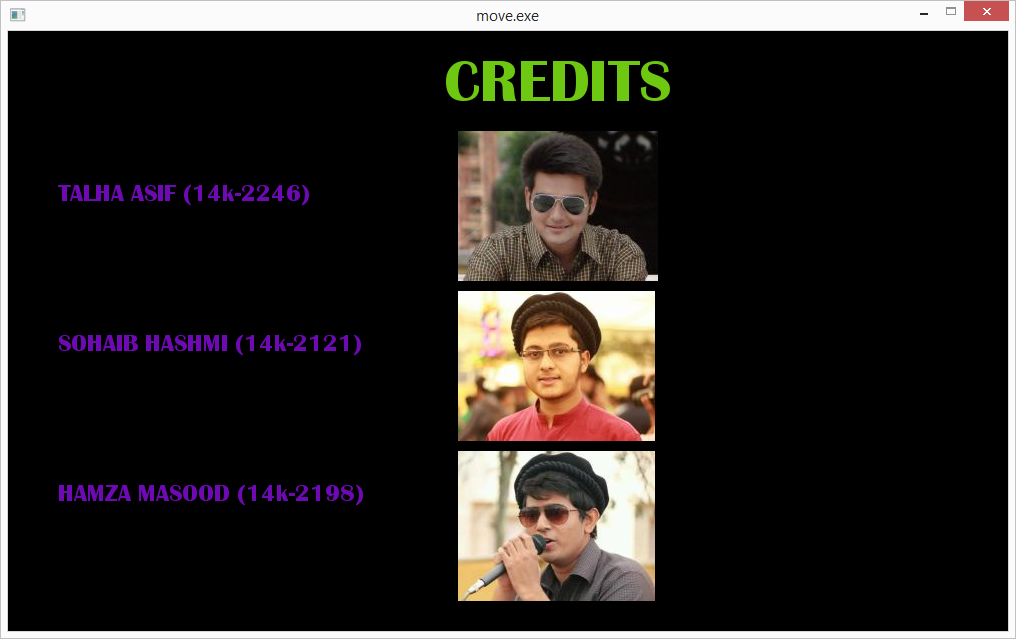
**SCREENSHOTS:**











**SOURCE FILE:**

* Dimension.h
* Credits.h
* Mainpage.h
* Background.h
* Comet.h
* Collide.h

**FUTURE ENHANCEMENTS:**

There’s a room of improvement in everything such is the case with our game.

We may add some characters in it including some coins that the user may collect along the way. We may also add some special powers in it. We also want to add sound but allegro 5 is having a problem these days as soon as the updated version of allegro 5 is released we’ll do the thing.

**REFERENCE:**

To understand allegro 5 visit youtube channel of Mike Greig