Ahsanullah University of Science & Technology

CSE1200 Software Development Project

Dept. of CSE

Section: B

Group: B2

Group Members:

- **190204093**

MD FARDIN JAMAN ARONOCK

- **190204103**

MD. NURUL YOUSUF KHAN

- **190204105**

MD. SYMUM HOSSAIN

- **190204108**

ASHRAFUL ALAM ABID

CSE1200 LAB PROJECT REPORT

TriGGo

Description:

Our project is basically a game project where we built a game named **TriGGo.** While making this game we used **iGraphics** and also we used **Visual Studio 2013** as a compiler. **TriGGo** is a challenging game. Our game contains two level. The difficulties increases with the scores and levels more and more. As a human being we face many obstacles and we tries to defeat each and every obstacles and our game is quite similar to the practical life like while playing **TriGGo** you will have to cross obstacles without any crashes and defeat your enemies. If you managed to reach the destination you will get the opportunity to make the high score on the high score board.

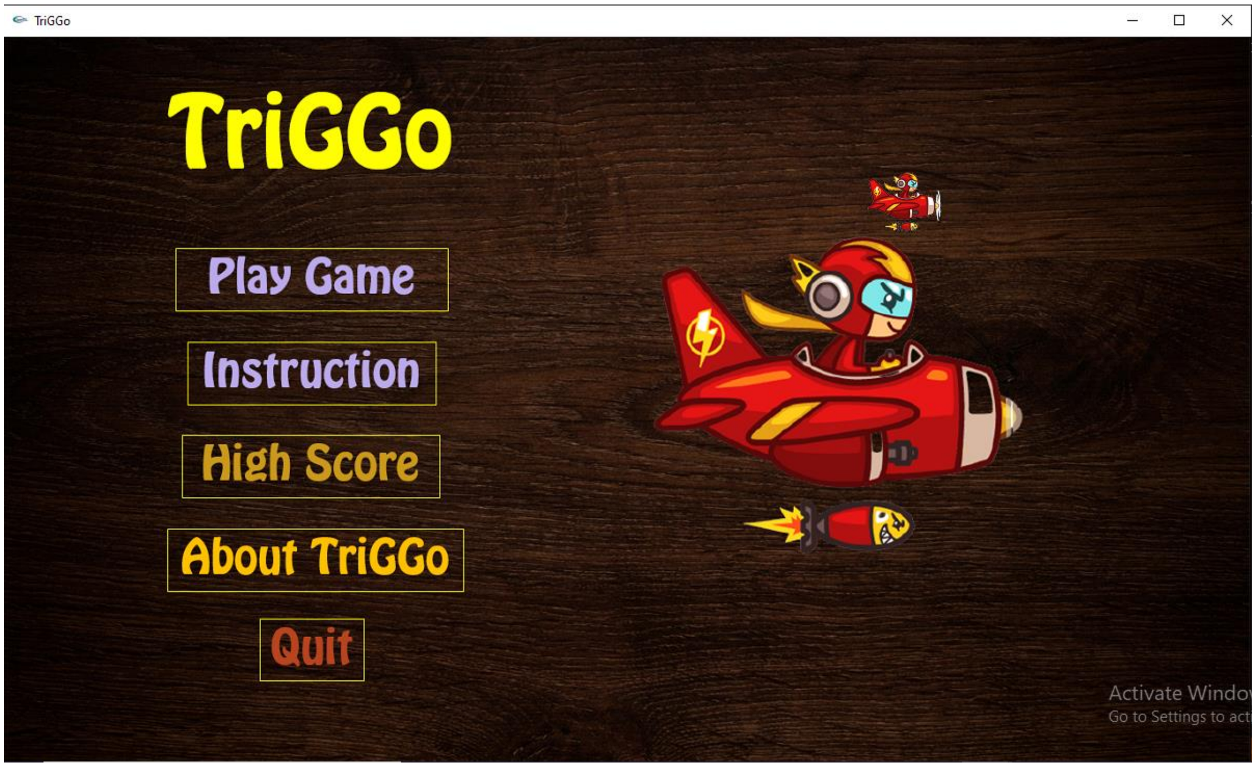
Controls:

- Press "B" or "b" for bullets in level 1
- Press pg up, pg down to operate TriGGo in level 1
- Press pg right, pg left, pg down, pg up (arrow keys) to operate TriGGo in level 2.
- Press "P" or "p" to Pause the game
- Press "R" or "r" to Resume the game
- Press "M" or "m" to operate the music system On or Off
- Pres Mouse left key to click options.

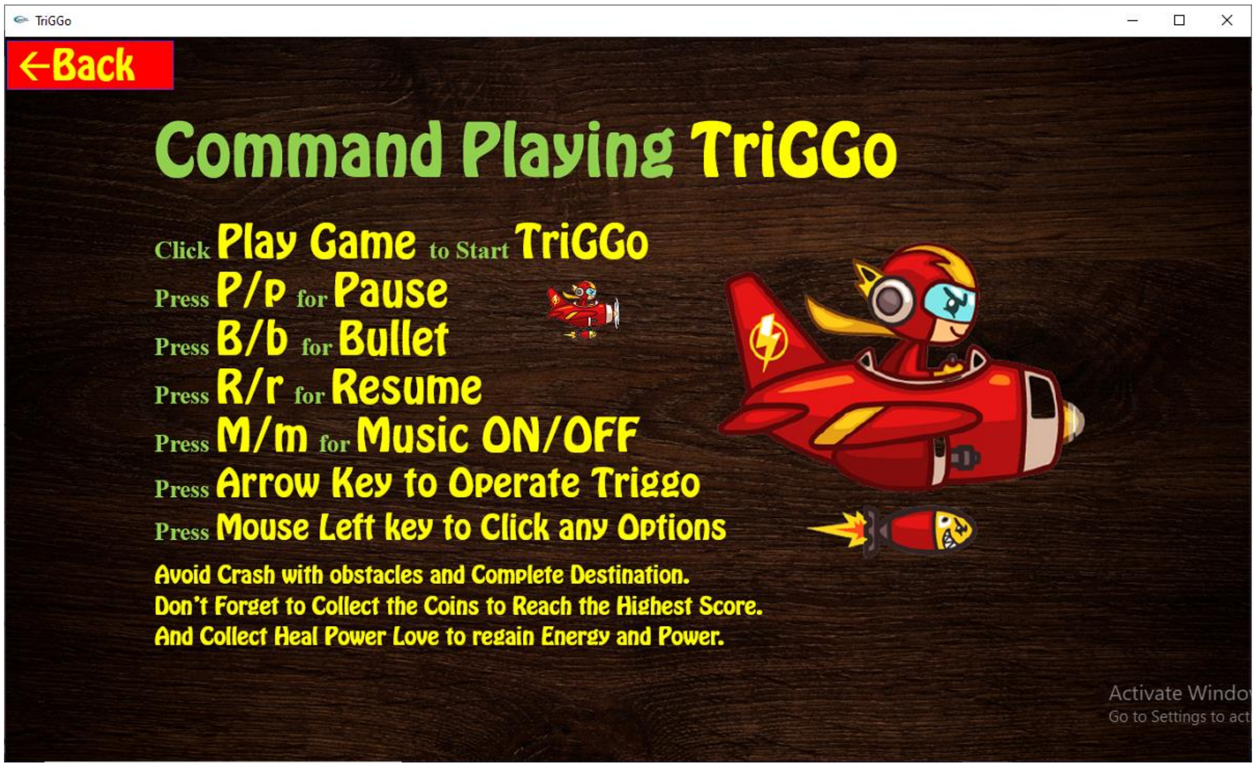
Here the game story is to rescue TriGGo's friend who was lost from his home. So the player who will play the game will have to help TriGGo to find his friend.

Game details through couples of snaps is given below with Open Window, Instruction, About TriGGo, High Score, Quit option and also the main game screen with Level 1 and Level 2 and besides the Game Over screen snap is also given from where the player can return back to the open window.

Open Window:



Instruction:


TriGGo

←Back

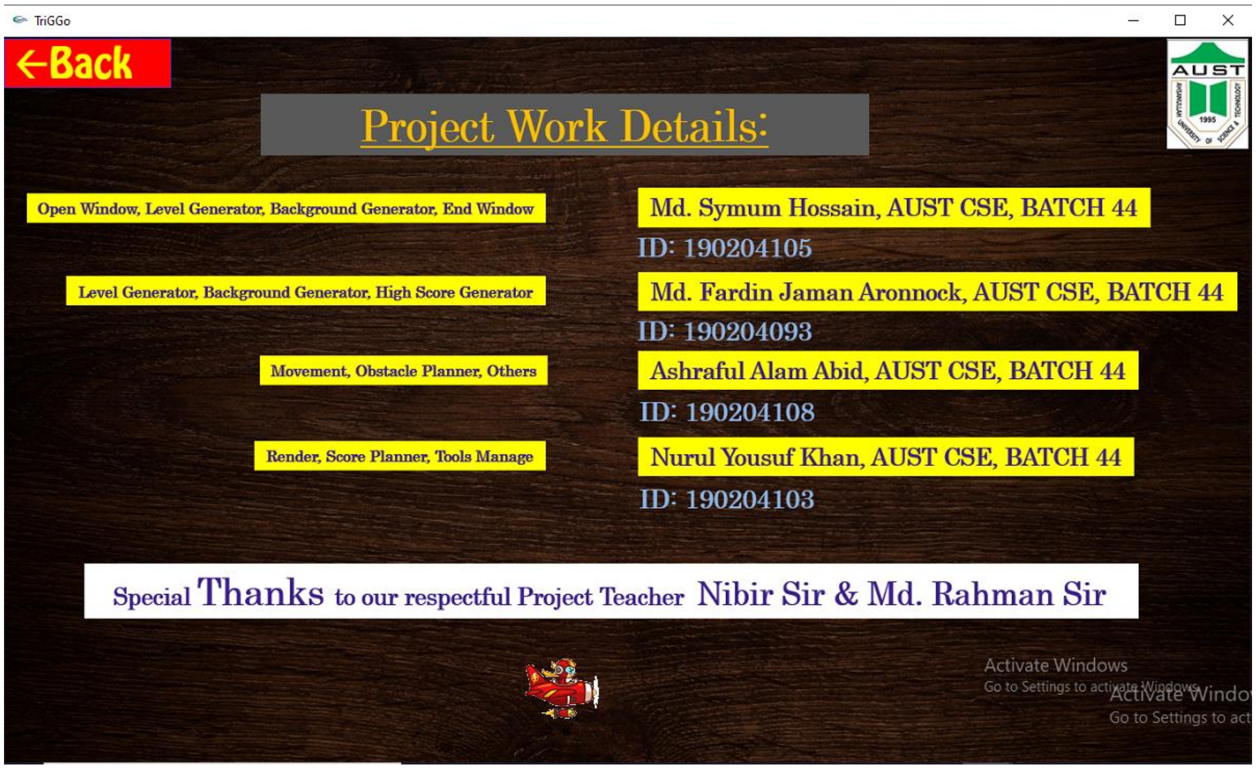
Command Playing TriGGo

Click Play Game to Start TriGGo
Press P/p for Pause
Press B/b for Bullet
Press R/r for Resume
Press M/m for Music ON/OFF
Press Arrow Key to Operate Triggo
Press Mouse Left key to Click any Options

Avoid Crash with obstacles and Complete Destination.
Don't Forget to Collect the Coins to Reach the Highest Score.
And Collect Heal Power Love to regain Energy and Power.

Activate Windo
Go to Settings to act

About TriGGo:



TriGGo

←Back

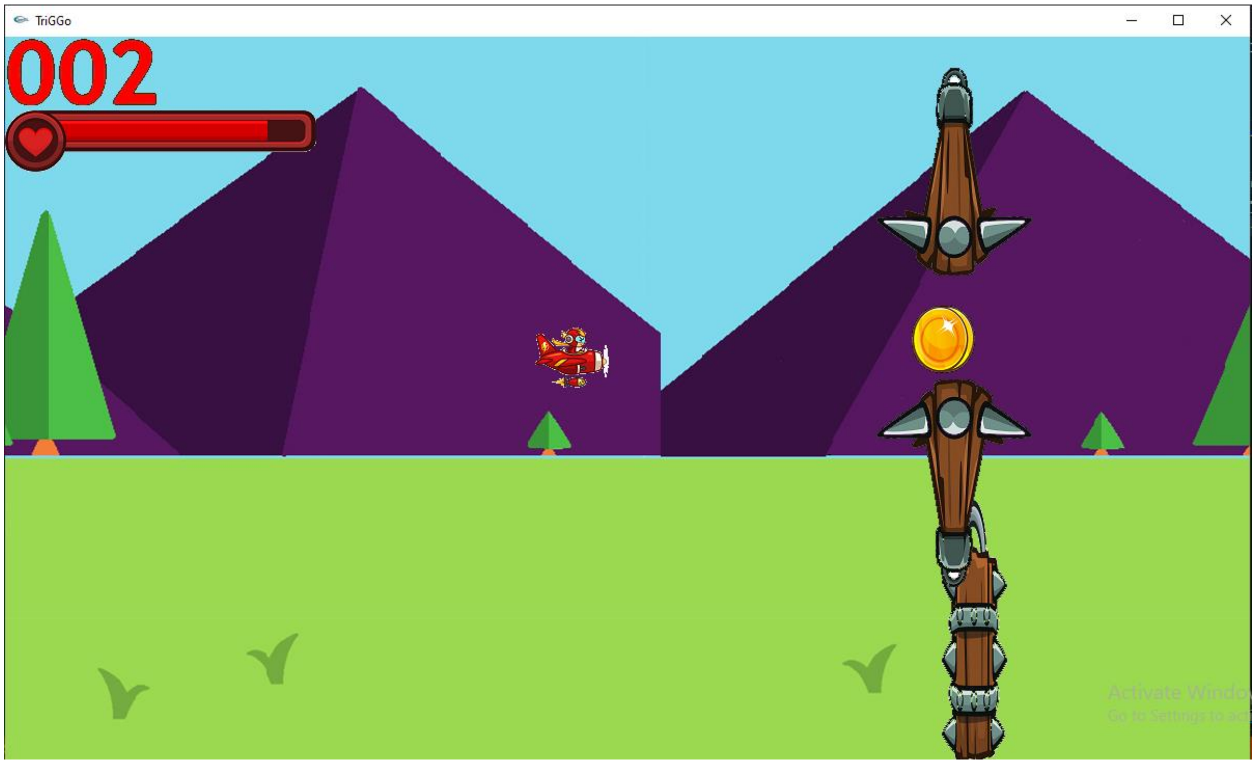
Project Work Details:

Open Window, Level Generator, Background Generator, End Window	Md. Symum Hossain, AUST CSE, BATCH 44 ID: 190204105
Level Generator, Background Generator, High Score Generator	Md. Fardin Jaman Aronnock, AUST CSE, BATCH 44 ID: 190204093
Movement, Obstacle Planner, Others	Ashraful Alam Abid, AUST CSE, BATCH 44 ID: 190204108
Render, Score Planner, Tools Manage	Nurul Yousuf Khan, AUST CSE, BATCH 44 ID: 190204103

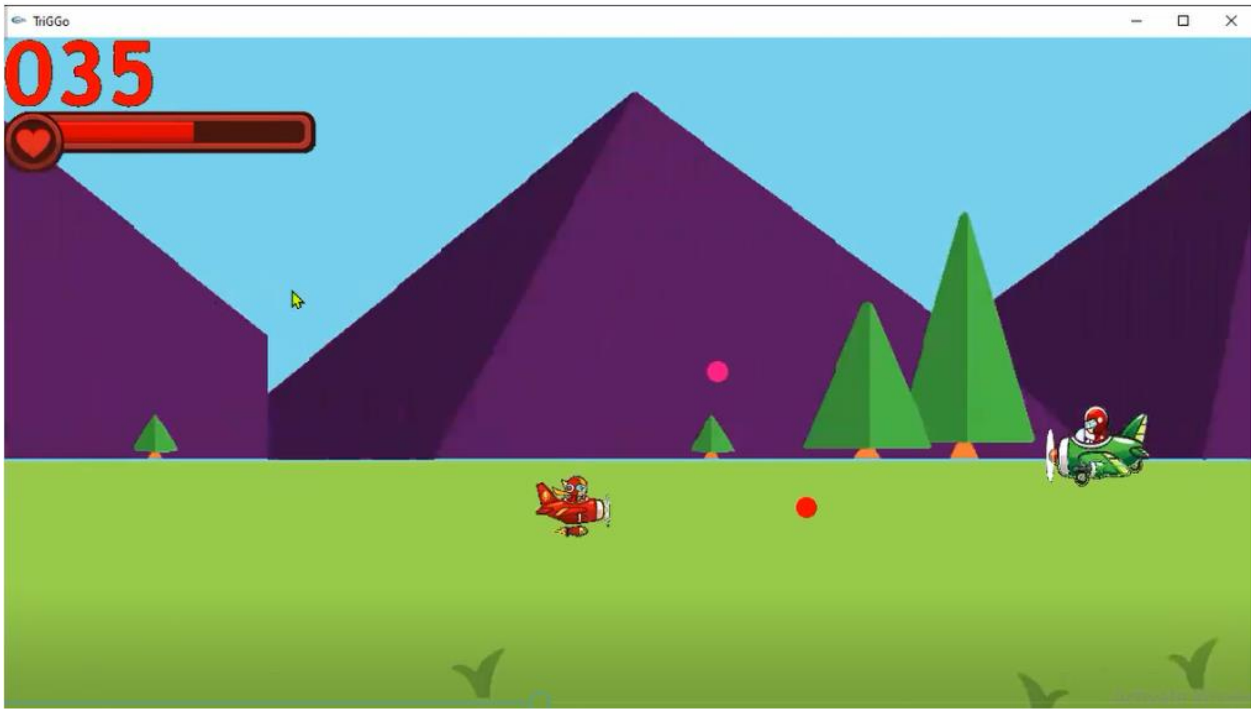
Special Thanks to our respectful Project Teacher Nibir Sir & Md. Rahman Sir

Activate Windows
Go to Settings to activate Windows.

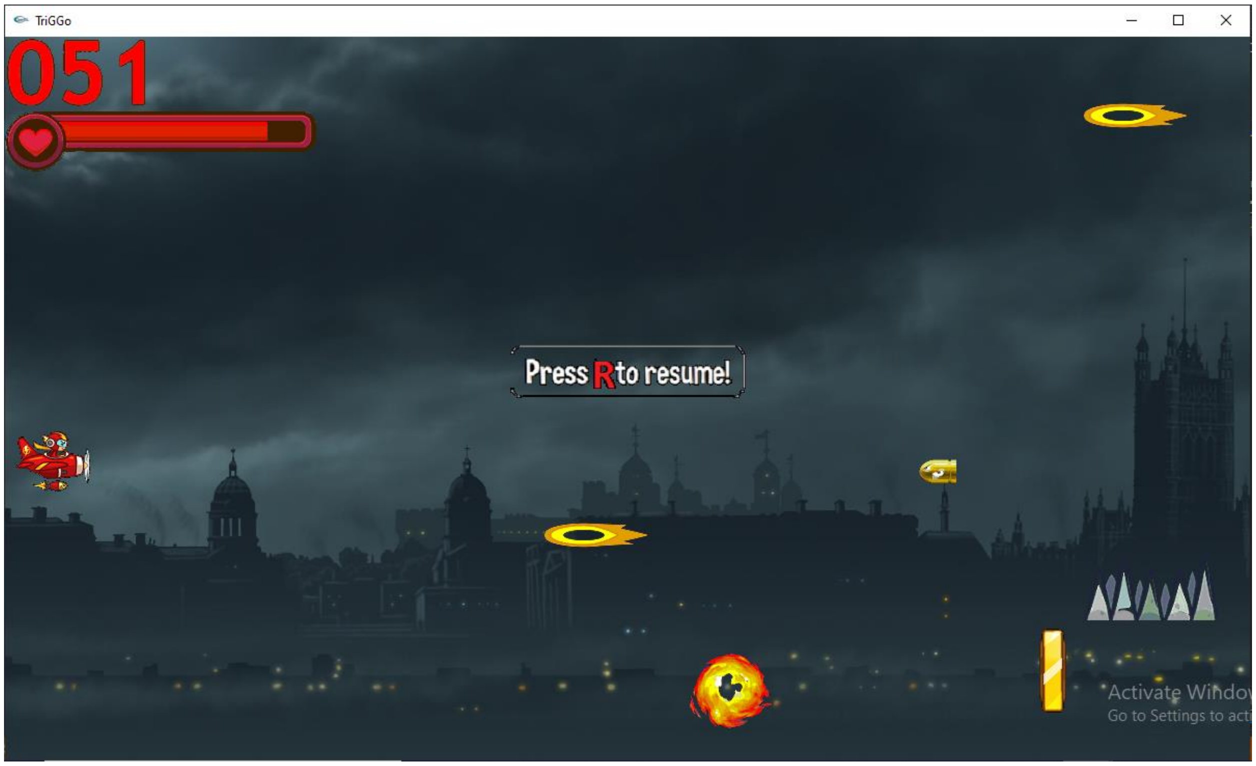
Start Game: (Level-1)



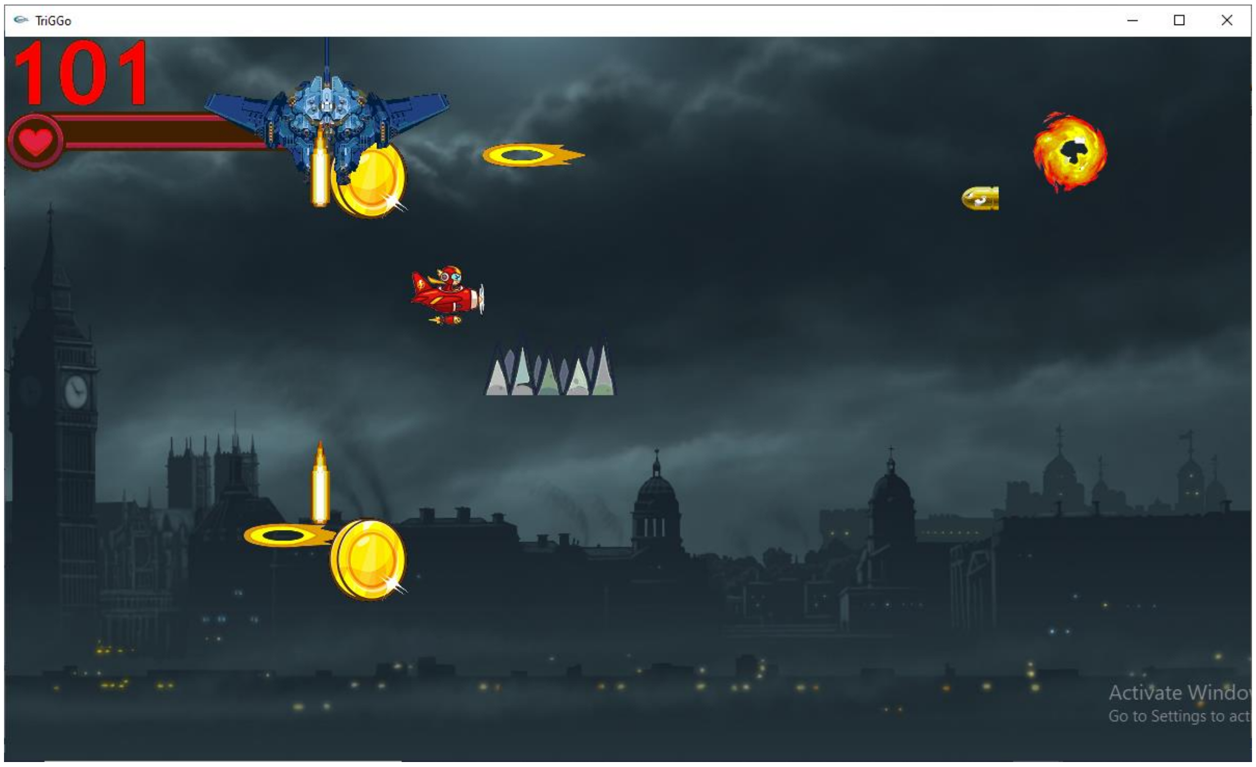
Level 1 with Boss:



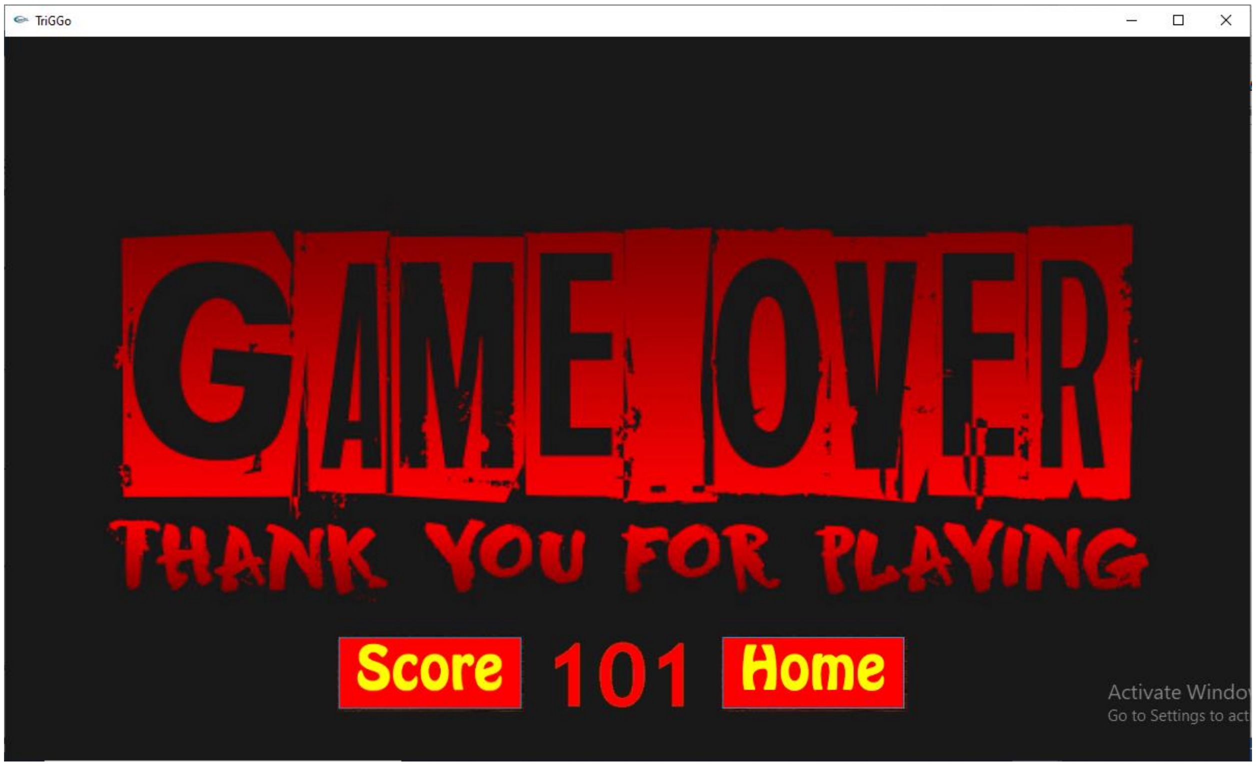
Level 2: (Game Pause)



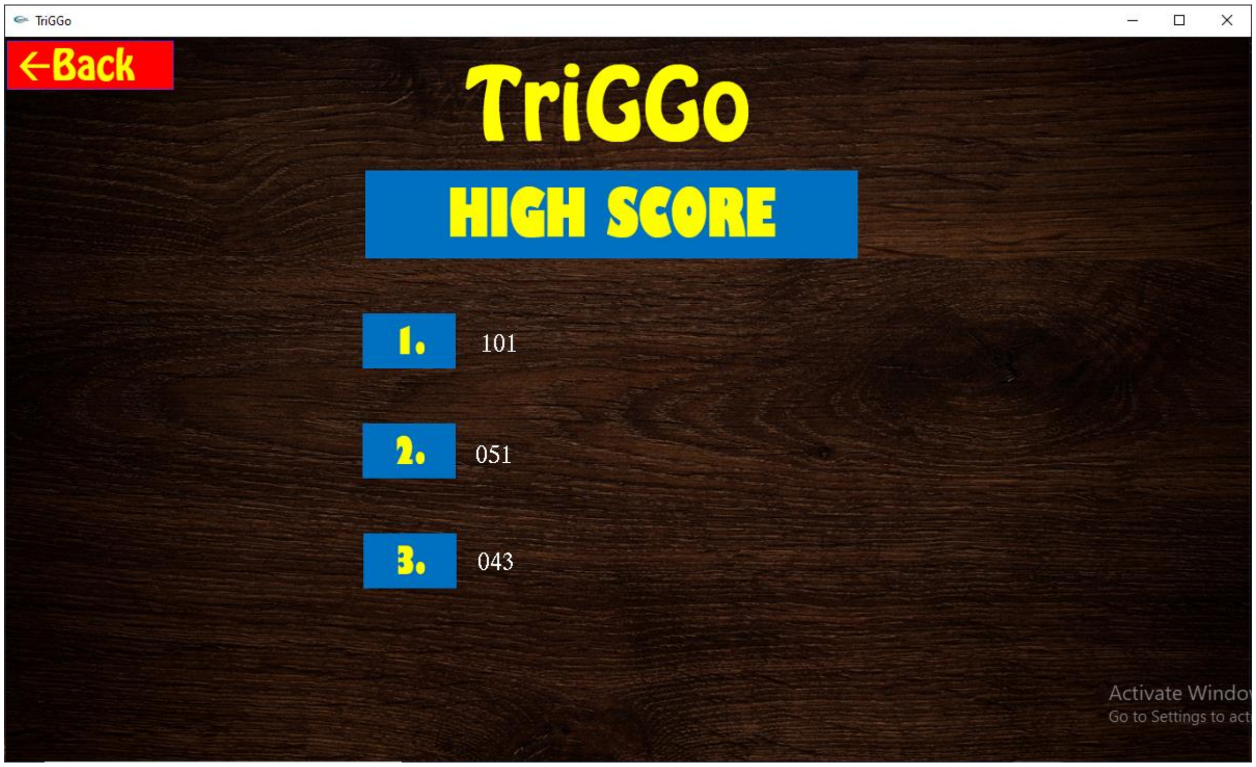
Level 2 with Boss:



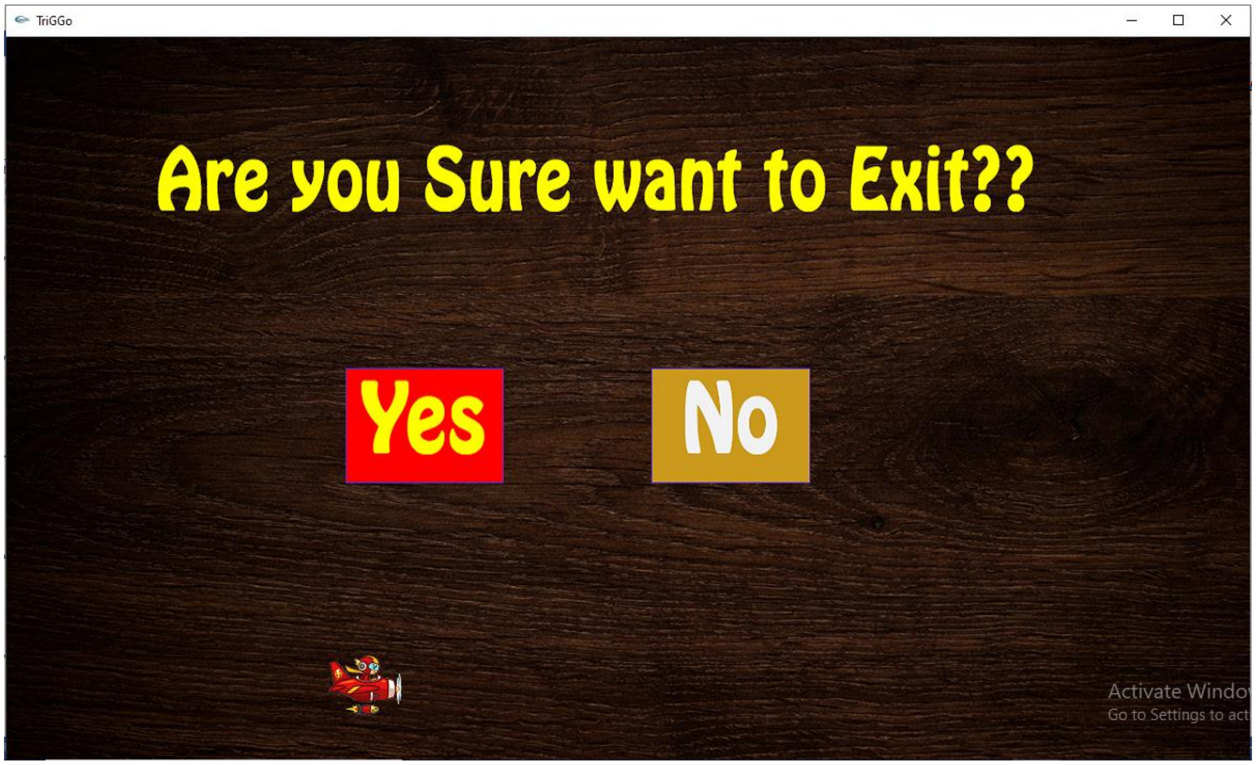
Game over Window:



High Score Board:



Quit:



Used functions and its purpose:

<u>Built-in Functions:</u>

PlaySound(): This built-in function used to play sound in project. We can play not only continuous sound but also, we can play one time music system.

iSetTimer(): Scheduled a task to perform after some pre-specified time interval. The specified function will be called again and again automatically after the specified time interval milisec. There can be at most 10 timers in our program. Once started these timers cannot be stopped. But they can be paused or resumed.

iInitialize(): Creates a window of specified height and width size and also a title.

main(): It is the main function from where the code starts to run its operation.

iSpecialKeyboard(): It is called whenever user hits special keys like function keys, home, end, pg up, pg down, arrows etc. you will have to use appropriate constants to detect them. A list is given: [GLUT_KEY_LEFT, GLUT_KEY_UP, GLUT_KEY_RIGHT, GLUT_KEY_DOWN, GLUT_KEY_PAGE UP, GLUT_KEY_PAGE DOWN, GLUT_KEY_HOME, GLUT_KEY_END, GLUT_KEY_INSERT etc.]

iKeyboard(): It is called whenever the user hits a key in keyboard. Here, key- holds the ASCII value of the key pressed.

iMouse(): This called when the user presses/releases the mouse. (mx, my) is the position where the mouse pointer is. To specify the point we used this.

iMouseMove(): We used this because it is called when the user presses and drags the mouse. (mx, my) is the position where the mouse pointer is.

iPassiveMouseMove(): this is called when the user move the mouse. (mx, my) is the position where the mouse pointer is. We used this to specify the point.

void iShowBMP(int x, int y, char filename[]): Shows a 24-bit .bmp file on screen. We used this to place our pictures.

Void ishowBMP2(int x, int y, char filename[],0 or 255): Shows a 24-bit .bmp file/ 256-bit .bmp file on screen. It can hide black and red layout of the .bmp file. We used this to place our pictures too.

mciSendString(): The mciSendString function sends a command string to an MCI device. Thedevice that the command is sent to is specified in the command string. We used this to operate a onetime sound when another sound is playing on the background.

void iText(GLdouble x, GLdouble y, char *str, void* font=GLUT_BITMAP_TIMES_ROMAN_24):

Displays a string on screen. We used it to show the high score to our high score board.

fopen(): The C library function FILE *fopen(const char *filename, const char *mode) opens the filename pointed to, by filename using the given mode. To read or write high score we used this function.

fscanf():The C library function int fscanf(FILE *stream, const char *format, ...) reads formatted input from a stream. To construct high score accurately we used this.

fprintf():The C library function int fprintf(FILE *stream, const char *format, ...) sends formatted output to a stream. To construct high score accurately we used this.

fclose(): The C library function int fclose(FILE *stream) closes the stream. All buffers are flushed.

iDraw(): This function is called again and again by the system. We can draw multiple activity by using this function. One of the most important function is iDraw(). We used it to construct our game accurately.

iClear(): We used this function to clears the screen.

void iPauseTimer(int index): Pauses the timer. The timer is de-activated. We used to pause our background rendering.

void iResumeTimer(int index): Resumes the timer. The timer is activated again. To resume our paused background rendering we used it.

iStart(): It is used to start project by calling main function.

rand(): It is used to generate random number by which we specified the direction of obstacles randomly.

Own Created Functions:

change() and backGroundChange() : used to operate the rendering of background.

imgposfunction() and backGroundPosition(): used to operate the background of level one and level two.

flag0(): Used to operate the opening window.

flag105(): Used to operate the high score window.

flag2(): Used to operate the quit option of the game.

flag5(): Used to store and see the details of TriGGo or to operate About TriGGo options.

flag6(): Used to operate the playing command window or the game instruction options.

gameOover(): To operate game over window we used this.

gamescreen(): We used this function to control the main playing level screen we used this.

void readScore(): We used this function to read high score from the text file.

Problems we faced:

We faced many difficulties as it was our first software development project. The main issues was the communication and discussing gap with the project members as we worked online. As iGraphics is not that much renowned so that the number of tutorials and solutions in the internet is a very minimum. And also there was a couples of small issues like interruptions of game speed because it takes a very high space and impact on our RAM. Debugging is also a bit different and not that easy.

We noticed that igraphics project runtime speed depends on processor and same project reacts differently on two different processor. In Intel cori7 10 gen processor our project running little faster where it's little slow down on cori5 8gen processor. As a result, obstacle and background rendering use in game running is faster in corei7.

How we solved problems:

We tried our best to resolve the discussing and communication gap. If we couldn't find the suitable solution in the internet writing **iGraphics** then we tried to search from the **OpenGL** as we got to know that iGraphics is quite similar to the **OpenGL** and it's much more renowned and broader which was very impactful to our project work.

To fixed the speed issues and rendering problem we used iSetTimer() to reduce difference reaction on this two type of processor. Again, we set flexible obstacle change rate as a variable and test it both processors. So finally, we find out that two different processor have different processing ability. We can reduce it but can't be fixed 100% accurately but we minimized.

Is there any way to improve the application?

Yes, the application can be improved more using widespread use of OpenGL parts instead of iGraphics. Also if we could use PlaySound function to operate multiple sound at a same moment then the code could be easier to written. Also the application can be improved by adding one or two levels more. There is some restrictions using iGraphics like we can't initialize any object in negative x or y axis. Also if we could manage to find the function to input name onto our screen then the score board could be more beautiful. If we could get ride out of these issues then the game could be more perfect and accurate.