**CONTENTS**

* **Abstract of the project**
* **Introduction of the project**
* **Implementation**
* **Source Code**
* **Results & Output of the project**
* **Conclusion**
* **references**

**ABSTRACT**

Main aim of this Mini Project is to illustrate 3D Car Animation using Turbo C++ Computer graphics. The concepts of Turbo C++ glut library and C++ has been used to create  3D Car Animation. We demonstrates all the animation of 3D car including standalone car, driving in day/night, wheel effect, fog effect, animate in different directions,  change of colors.

There is user interaction in this CG program. We have used input devices like mouse and key board to interact with program.

3D computer graphics or three-dimensional computer graphics (in contrast with 2D computer graphics) are graphics that use a three dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering 2D images. Such images may be stored for viewing later or displayed in real time.

3D computer graphics rely on many of the same algorithms as 2D computer vector graphics in the wire frame model and 2D computer raster graphics in the final rendered display. In computer graphics software, 2D applications may use 3D techniques to achieve effects such as lighting, and 3D may use 2D rendering techniques.

3D computer graphics are often referred to as 3D models. Apart from the rendered graphic, the model is contained within the graphical data

**INTRODUCTION**

In 3D computer graphics, 3D modeling is the process of developing a mathematical coordinate-based representation of any surface of an object (inanimate or living) in three dimensions via specialized software by manipulating edges, vertices, and polygons in a simulated 3D space.[1][2][3]

Three-dimensional (3D) models represent a physical body using a collection of points in 3D space, connected by various geometric entities such as triangles, lines, curved surfaces, etc.[4] Being a collection of data (points and other information), 3D models can be created manually, algorithmically (procedural modeling), or by scanning.[5][6] Their surfaces may be further defined

3D models are now widely used anywhere in 3D graphics and CAD but their history predates the widespread use of 3D graphics on personal computers.[10]

In the past, many computer games used pre-rendered images of 3D models as sprites before computers could render them in real-time. The designer can then see the model in various directions and views, this can help the designer see if the object is created as intended to compared to their original vision. Seeing the design this way can help the designer or company figure out changes or improvements needed to the product.

**IMPLEMENTATIONS**

**Hardware Requirements**

This project consist of the following Hradware Materials :

HP laptop Core i5 with 8GB RAM & 512GB SSD And it includes AMD radeon graphics card.

**Operating System**

For this project I am using Windows 11 operating system.

**Software Requirements**

**Turbo C3**

Turbo C is a discontinued integrated development environment (IDE) and compiler for the C programming language from Borland. First introduced in 1987, it was noted for its integrated development environment, small size, fast compile speed, comprehensive manuals and low price.

**Functions Used In Program**

In this program I am using various functions they are follows :

* Line( ) function is used to draw a line from a point (x1,y1) to point (x2,y2).
* Initgraph( )function initialize the graphics system by loading graphics driver from disk and putting system into graphics mode.
* Settextstyle( ) function used to change the way in which text appears.
* Outtextxy( ) function Displays the text or string at a specified point (x,y) on the screen.
* Setcolor( ) function Used to set yhe current drawing color to the new color.
* Ellipse ( ) Ellipse is defined as the locus of a point in a plane which moves in a plane in such a manner that the ratio of its distance from a fixed point called focus in the same plane to its distance from a fixed straight line called directrix is always constant, which should always be less than unity
* Putpixel( ) function The header file graphics. h contains putpixel() function which plots a pixel at location (x, y) of specified color.
* Setfillstyle( ) function which sets the current fill pattern and fill color. floodfill() function is used to fill an enclosed area. Current fill pattern and fill color is used to fill the area.
* Floodfill( ) floodfill() function is used to fill an enclosed area. Current fill pattern and fill color is used to fill the area
* delay ( ) The delay() function in C is used to stop the execution of the program for some period of time. Parameters: It accepts a time in milliseconds to stop the execution of the program to that period of time.
* Int main( ) The delay() function in C is used to stop the execution of the program for some period of time. Parameters: milliseconds to stop the execution of the program to that period of time.

**SOURCE CODE**

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<stdlib.h>

#include<process.h>

void main()

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

setbkcolor(0);

int t;

while(1)

{

settextstyle(2,0,5);

outtextxy(100,10,"CODING BY OM ");

outtextxy(100,25,"Press L,H,T,P");

outtextxy(100,40,"Press 1 for Quite");

as:

setcolor(13);

ellipse(380,127,20,152,130,35);

line(490,109,560,142);

line(560,142,569,142);

line(569,142,582,102);

line(582,102,620,92);

line(593,132,617,125);

line(617,124,627,96);

line(620,92,628,97);

line(472,86,602,96);

line(501,113,575,121);

line(443,77,475,80);

line(443,77,432,93);

line(475,80,472,85);

line(593,132,593,137);

line(593,137,600,141);

line(600,141,600,185);

line(600,185,608,192);

line(608,192,608,192);

line(608,234,586,253);

line(263,112,363,127);

line(193,160,263,112);

line(193,160,220,170);

line(220,170,280,180);

line(280,180,320,185);

line(320,185,363,127);

line(340,194,460,169);

line(460,169,519,152);

ellipse(512,144,300,30,10,10);

ellipse(467,143,28,100,50,30);

line(510,128,521,138);

line(435,116,440,100);

line(339,194,372,144);

ellipse(454,208,87,123,128,95);

line(372,144,384,128);

int b,x,y;

line(365,298,524,264);

line(365,298,330,310);

line(330,310,323,310);

ellipse(162,221,135,190,90,40);

line(96,193,140,174);

line(140,174,160,168);

line(160,168,192,161);

ellipse(75,246,95,190,18,18);

line(57,251,57,286);

ellipse(181,178,232,263,200,137);

ellipse(195,180,256,286,200,137);

ellipse(191,171,228,247,200,100);

ellipse(231,198,234,275,200,80);

ellipse(196,167,228,246,200,90);

ellipse(231,184,234,276,200,80);

ellipse(191,200,228,246,200,90);

ellipse(228,218,234,276,200,80);

ellipse(258,268,180,220,200,40);

ellipse(178,296,244,355,16,10);

ellipse(238,249,227,250,200,60);

ellipse(302,281,320,77,26,45);

ellipse(290,277,65,162,40,45);

ellipse(278,288,144,212,300);

ellipse(302+260,229,328,87,26,45);

ellipse(290+280-7,277-50+2,90,162,40,45);

ellipse(278+270,288-50,144,215,27,45);

b=0;

int v=0;

ellipse(302+250+v,227+b,295,90,29,41);

ellipse(302+234+v,231+b,245,306,50,40);

ellipse(302+248+v,229+b,0,360,21,30);

ellipse(302+247+v,229+b,0,360,8,10);

setfillstyle(6,11);

line(546+v,201+b,546+v,220+b);

line(551+v,201+b-2,551+v,220+b);

line(546+v,238+b,546+v,257+b);

line(551+v,238+b+2,551+v,257+b+2);

line(530+v,225+b,541+v,225+b);

line(530+v,230+b,541+v,230);

line(557+v,225+b,570+v,225+b);

line(557+v,230+b,570+v,230+b);

line(563+v,206+b,552+v,222+b);

line(534+v,246+b,543+v,232+b);

line(566+v,210+b,556+v,223+b);

line(536+v,250+b,544+v,238+b);

line(536+v,207+b,546+v,222+b);

line(532+v,213+b,542+v,224+b);

line(556+v,235+b,566+v,247+b);

line(551+v,237+b,563+v,253+b);

v=-260;

b=56;

ellipse(302+233+v,221+b,260,60,49,51);

ellipse(302+243+v,224+b,0,360,28,35);

ellipse(300+245+v,223+b,0,360,10,12);

ellipse(285+249+v,239+b,210,260,30,33);

b=45;

v=v-4;

line(546+v,201+b,546+v,220+b+2);

line(551+v,201+b,551+v,220+b+2);

b=b+8;

line(546+v,238+b,546+v,257+b+4);

line(551+v,238+b,551+v,257+b+4);

v=v-2;

line(530+v-6,225+b,541+v,225+b);

line(530+v-6,230+b,541+v,230+b);

v=v+5;line(557+v,225+b,570+v+3,225+b);

line(557+v-1,230+b,570+v+3,230+b);

b=b-5;

v=v-5;

line(565+v+3,206+b,552+v+4,222+b-2);

b=b+15;

line(534+v,246+b,543+v+3,232+b-5);

b=b-10;

line(566+v+7,210+b-5,566+v+4,220+b);

line(536+v-5,250+b,544+v-2,238+b-4);

line(536+v,207+b-8,545+v,222+b-5);

line(531+v,212+b-8,542+v,224+b-2);

line(556+v,235+b,566+v+3,247+b+5);

line(551+v,237+b,563+v+2,253+b+3);

ellipse(199,250,144,345,18,8);

line(185,245,206,230);

ellipse(223,234,340,110,8,5);

line(230,237,217,252);

line(206,230,220,229);

line(90,223,152,236);

line(152,236,137,254);

line(90,223,90,242);

ellipse(240,270,104,136,100,60);

ellipse(185,237,120,160,100,60);

ellipse(80,221,357,134,10,10);

line(152,236,168,228);

line(435,116,440,171);

//////////////////////////////////hp//////////////////////////

line(134,185,196,160);

line(214,212,318,185);

ellipse(166,247,99,330,8,8);

ellipse(171,243,310,129,7,7);

putpixel(174,250,13);

putpixel(173,251,13);

putpixel(164,239,13);

putpixel(165,238,13);

setcolor(13);

line(1,430,639,300);

line(1,445,639,315);

line(1,210,93,194);

line(1,195,194,158);

line(520,90,639,71);

line(478,86,639,56);

int c=0;

line(10,194+c,10,208+c);

line(40,189+c,40,204+c);

line(70,183+c,70,198+c);

line(100,176+c,100,190+c);

line(130,170+c,130,177+c);

line(160,166+c,160,168+c);

line(190,160+c,190,161+c);

line(190+330,78+c,190+330,89+c);

line(190+360,72+c,190+360,85+c);

line(190+390,67+c,190+390,81+c);

line(190+420,62+c,190+420,76+c);

line(190+449,57+c,190+449,71+c);

c=236;

line(10,192+c,10,208+c);

line(40,189+c-2,40,204+c-3);

line(70,183+c-3,70,198+c-3);

line(100,176+c-2,100,190-2);

line(130,170+c-2,130,177+c+5);

line(160,166+c-3,160,168+c+8);

line(190,160+c-4,190,161+c+9);

line(190+30,156+c-5,190+30,170+c-5);

line(190+30+30,156+c-12,190+30+30,170+c-12);

line(190+90,156+c-18,190+90,170+c-17);

line(190+120,156+c-25,190+120,170+c-25);

line(190+150,156+c-30,190+150,170+c-30);

line(190+180,156+c-37,190+180,170+c-36);

line(190+210,156+c-42,190+210,170+c-42);

line(190+240,156+c-48,190+240,170+c-48);

line(190+270,156+c-55,190+270,170+c-54);

line(190+300,156+c-61,190+300,170+c-61);

line(190+330,78+c+10,190+330,89+c+13);

line(190+360,72+c+11,190+360,85+c+13);

line(190+390,67+c+10,190+390,81+c+10);

line(190+420,62+c+8,190+420,76+c+10);

line(190+449,57+c+8,190+449,71+c+8);

setcolor(12);

line(1,310,25,306);

line(6,318,30,315);

line(1,310,6,318);

line(25,306,30,314);

setcolor(12);

line(605,310-128,629,306-128);

line(610,318-128,634,315-128);

line(605,310-128,610,318-128);

line(629,306-128,634,314-128);

setcolor(12);

line(46,302,70,298);

line(51,310,75,307);

line(46,302,51,310);

line(70,298,75,306);

setfillstyle(1,0);

floodfill(64,303,12);

setfillstyle(1,14);

floodfill(14,314,12);

floodfill(617,183,12);

setfillstyle(1,0);

floodfill(14,314,12);

floodfill(617,183,12);

setfillstyle(1,14);

floodfill(64,303,12);

t=getch();

if(t=='1')

exit(0);

if(t=='h')

{

sound(710);

delay(500);

nosound();

}

if(t=='t')

{

while(!kbhit())

{

setfillstyle(1,0);

floodfill(536,213,13);

floodfill(563,213,13);

floodfill(561,244,13);

floodfill(538,244,13);

floodfill(274,295,13);

floodfill(294,295,13);

floodfill(274,265,13);

floodfill(294,265,13);

floodfill(548,250,13);

floodfill(548,214,13);

floodfill(533,228,13);

floodfill(563,228,13);

floodfill(262,281,13);

floodfill(308,281,13);

floodfill(284,251,13);

floodfill(284,295,13);

setfillstyle(1,random(12));

floodfill(200,250,13);

delay(10);

//setfillstyle(1,11);

floodfill(170,250,13);

floodfill(80,230,13);

}

setfillstyle(1,0);

floodfill(200,250,13);

delay(10);

floodfill(170,250,13);

floodfill(80,230,13);

}

if(t=='l')

{

while(!kbhit())

{

delay(120);

setfillstyle(6,0);

floodfill(536,213,13);

floodfill(563,213,13);

floodfill(561,244,13);

floodfill(538,244,13);

floodfill(274,295,13);

floodfill(294,265,13);

setfillstyle(1,0);

floodfill(64,303,12);

setfillstyle(9,0);

floodfill(81-40+5,419+7,13);

floodfill(151-40,409+7,13);

floodfill(211-40,397+7,13);

floodfill(271-40,380+7,13);

floodfill(331-40,368+7,13);

floodfill(396-40,355+7,13);

floodfill(450-40,345+7,13);

floodfill(510-40,335+7,13);

floodfill(570-40,325+7,13);

floodfill(630-40,312+7,13);

floodfill(50,197,13);

floodfill(110,177,13);

floodfill(166,165,13);

floodfill(527,86,13);

floodfill(587,71,13);

setfillstyle(6,14);

floodfill(548,250,13);

floodfill(548,214,13);

floodfill(533,228,13);

floodfill(563,228,13);

floodfill(262,281,13);

floodfill(308,281,13);

floodfill(284,251,13);

floodfill(284,295,13);

setfillstyle(9,10);

floodfill(19,429,13);

floodfill(81,429,13);

floodfill(151,409,13);

floodfill(211,397,13);

floodfill(271,380,13);

floodfill(331,368,13);

floodfill(396,355,13);

floodfill(450,345,13);

floodfill(510,335,13);

floodfill(570,325,13);

floodfill(630,312,13);

///////////////////////////////////

floodfill(20,197,13);

floodfill(80,187,13);

floodfill(133,174,13);

floodfill(517,86,13);

floodfill(557,81,13);

floodfill(627,70,13);

setfillstyle(1,14);

floodfill(14,314,12);

floodfill(617,183,12);

setfillstyle(10,4);

floodfill(302+248,230,13);

floodfill(302+248+v,230+b,13);

setfillstyle(6,11);

floodfill(200,250,13);

floodfill(170,250,13);

floodfill(80,230,13);

delay(120);

setfillstyle(6,0);

floodfill(548,250,13);

floodfill(548,214,13);

floodfill(533,228,13);

floodfill(563,228,13);

floodfill(262,281,13);

floodfill(308,281,13);

floodfill(284,251,13);

floodfill(284,295,13);

setfillstyle(9,0);

floodfill(19,429,13);

floodfill(81,419,13);

floodfill(151,409,13);

floodfill(211,397,13);

floodfill(271,380,13);

floodfill(331,368,13);

floodfill(396,355,13);

floodfill(450,345,13);

floodfill(510,335,13);

floodfill(570,325,13);

floodfill(630,312,13);

floodfill(20,197,13);

floodfill(80,187,13);

floodfill(133,174,13);

floodfill(517,86,13);

floodfill(557,81,13);

floodfill(627,70,13);

//////////////////////////////////

setfillstyle(1,0);

floodfill(14,314,12);

floodfill(617,183,12);

setfillstyle(6,10);

floodfill(536,213,13);

floodfill(563,213,13);

floodfill(561,244,13);

floodfill(538,244,13);

floodfill(274,295,13);

floodfill(294,295,13);

floodfill(274,265,13);

floodfill(294,265,13);

setfillstyle(9,14);

floodfill(81-40+5,419+7,13);

floodfill(151-40,409+7,13);

floodfill(211-40,397+7,13);

floodfill(271-40,380+7,13);

floodfill(331-40,380+7,13);

floodfill(396-40,355+7,13);

floodfill(450-40,345+7,13);

floodfill(570-40,325+7,13);

floodfill(630-40,312+7,13);

floodfill(50,197,13);

floodfill(110,177,13);

floodfill(166,165,13);

floodfill(527,86,13);

floodfill(587,71,13);

setfillstyle(1,14);

floodfill(64,303,12);

setfillstyle(9,4);

floodfill(302+248,230,13);

floodfill(302+248+v,230+b,13);

delay(20);

setfillstyle(1,14);

floodfill(200,250,13);

floodfill(170,250,13);

floodfill(80,230,13);

delay(20);

setfillstyle(1,0);

floodfill(200,250,13);

floodfill(170,250,13);

floodfill(80,230,13);

} }

if(t=='p')

{

int n=0;

while(!kbhit())

{

if(n<=60)

n++;

setcolor(0);

rectangle(1+1,-10,90-1,-12+n);

delay(14);

setcolor(9);

rectangle(1,-10,90,-10+n);

if(n==60)

{

outtextxy(10,10,"THANK YOU");

outtextxy(10,30,"-----ITS OMI");

delay(400);

}

}

}

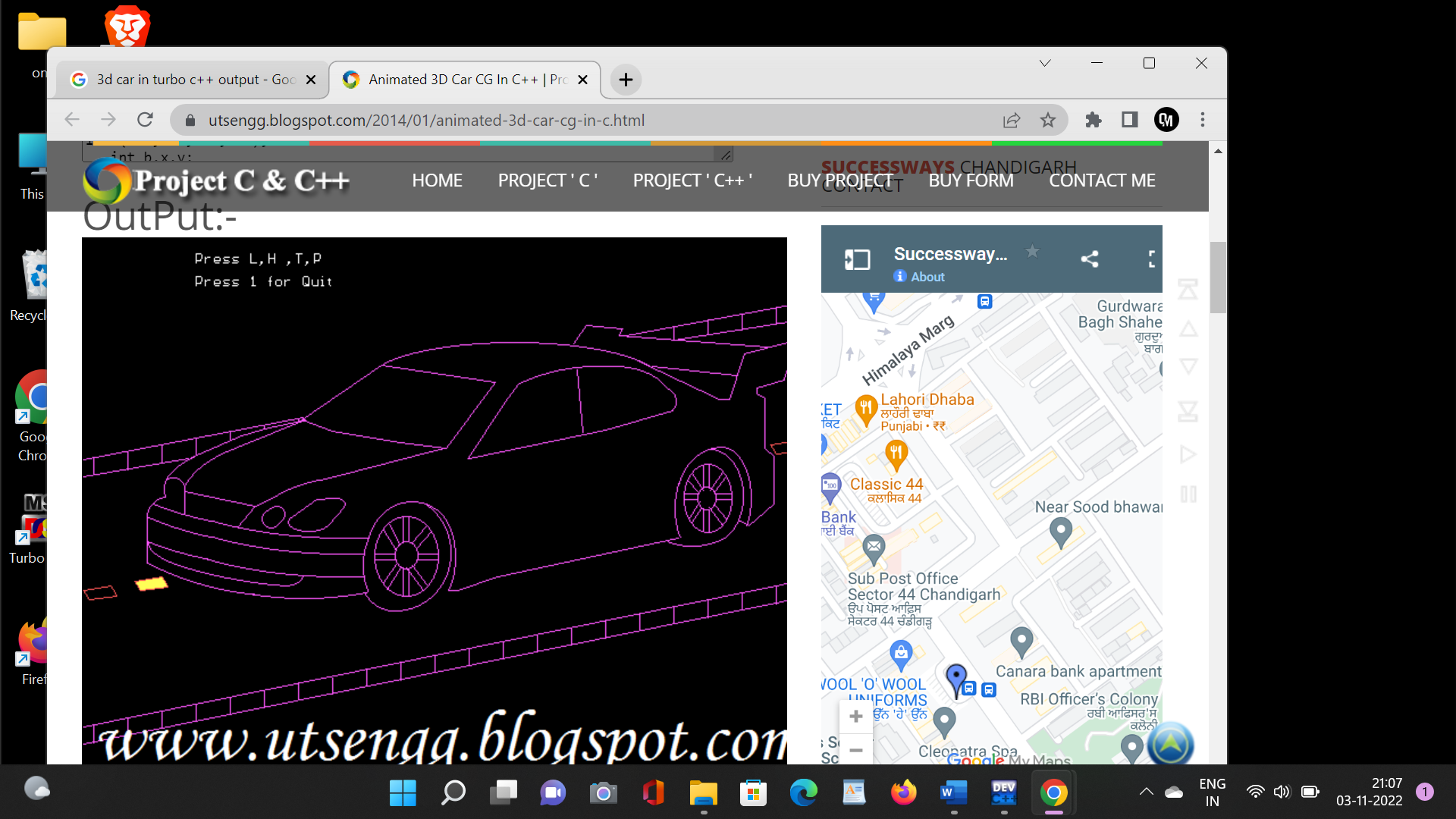
getch();

}

}

**RESULTS & OUTPUT**

**Output of the programm :**

****

****

**CONCLUSION**

In this chapter, it has show how animation was begun in the early day of animation. It also discussed about 2D and 3D animation briefly and some example of 2D and 3D animation films.

The development from 2D to 3D animation is driven by the invention of many programs such as Maya and 3D Max. These programs made creating 3D characters and 3D animated projects possible. With the popularity of movies like Toy Story and Shriek, the genre of 3D animated films has established its place in 21st century culture.

more, there's a demand for 3D animation over and beyond what animation houses and movie studios are currently producing.

From the example of 3D animation story given on the previous sub chapter, it shows that most of the 3D animation today less focuses on giving moral value to the end users. This is why, through this project animator try to create a 3D animation story which conveyed a positive moral value so that the end user might get some education while watching this 3D animation.

Thus based on the concept and technique that have been studied, it will be applied in developing this project to create a short animation story titled Hari yang malang dalam hidupku.

**REFERENCES**

1. <https://www.researchgate.net/publication/344285614_A_PROJECT_ON_3D_CAR_SIMULATION>
2. <https://github.com/iamgovindthakur/3D-Car-Animation-Using-OpenGL/blob/master/3D%20car%20project%20report.pdf>
3. <https://www.openglprojects.in/2021/06/computer-graphics-mini-project-3d-car.html>
4. <https://www.shutterstock.com/search/car-3d-abstract>
5. <https://cplusplus.happycodings.com/computer-graphics/code28.html>
6. <https://www.youtube.com/watch?v=vNerxKSLb5w>
7. <https://www.youtube.com/watch?v=f1gyphQMGe0>