V.E.S. INSTITUTE OF TECHNOLOGY

Computer Graphics and Virtual Reality

Mini Project Report on

ROYAL MINT OF SPAIN 2D MODEL USING TURBO-C

Submitted in partial fulfillment of the requirements of

Second Year Computer Engineering

by

KUNAL KOTKAR (32)

SHREEJA NANDA (39)

DEPARTMENT OF COMPUTER ENGINEERING

V.E.S INSTITUTE OF TECHNOLOGY

2019 – 2020

ROYAL MINT OF SPAIN 2D MODEL

ABSTRACT:

As stated in the title, we have created a famous architectural structure of Spain, La Casa De Papel i.e. The Royal Mint of Spain using the graphics.h header library included in Turbo C. This project showcases the use of certain functions in this library and creating a 2-Dimensional model of this famous institute. The project includes a delayed transition of all parts of the model so as to give it an animated effect, adding to the refinement of this model.

REQUIREMENTS:

A) **SOFTWARE USED:** Turbo C by Akki

B) **HARDWARE USED:** Dell Inspiron 7570 Intel Core i7 8th Gen Laptop with 4GB Graphics

INTRODUCTION:

As with any software project, it is possible to utilize a vast array of existing libraries and solutions to get done. Yet, we laid emphasis on making this entire project ourselves instead of depending on external agents as a way to showcase our skills and enhance our own learning.

For the following project, the following major elements were needed for a successful effort:

1) A programming language well known to both of us.
2) An accurate procedural model of the architecture.
3) Functional delay in showcasing every part of the model.

FIG: PROPOSED MODEL OF THE ROYAL MINT OF SPAIN

A) LIBRARIES USED:

1) graphics.h: graphics.h is a header that provides access to a simple graphics library that makes it possible to draw lines, rectangles, ovals, arcs, polygons, images, and strings on a graphical window.

2) stdlib.h: stdlib.h is the header of the general-purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions, and others.

3) stdio.h: Input and Output operations can also be performed in C++ using the C Standard Input and Output Library (cstdio, known as stdio.h in the C language). This library uses what are called *streams* to operate with physical devices such as keyboards, printers, terminals or any other type of files supported by the system. Streams are an abstraction to interact with these in a uniform way; All streams have similar properties independently of the individual characteristics of the physical media they are associated with.

4) conio.h: The conio.h header file used in C programming language contains functions for console input/output. Some of its most commonly used functions are clrscr, getch,

getche, kbhit, etc. They can be used to clear screen, change the color of text and background, move text, check whether a key is pressed or not and perform other tasks.

B) EXPLANATION:

Using basic and known functions of graphics.h library, this entire project has been created. Initially, a base rectangle was created using the **rectangle** function of the graphics.h library. This rectangle was drawn to act as the foundation and to demarcate the entire structure size on the graphics screen. A **top-down approach** was used in making this structure. The **rectangle** function has been widely used throughout the model because of its quadratic symmetry.

After the creation of the base, we started with the **central structure**. The same **top-down approach** was used in creating it. Starting from the **top central window-like structure**, this entire part has been created using the **rectangle** function, with altering dimensions. The **pillars** were created by using a repetitive **for-loop** so as to be constructed symmetrically throughout the central base. The **stairs** were made to appear cascadingly using another **for-loop** using increasing dimensions of them equally from both sides, using the **external pillar bases** as a reference.

Now, for the **floors**, we used a **top-down bisectional approach**, wherein we divided the construction of the floors into two halves i.e to the **left** and **right** of the **central structure,** so as to maintain exact symmetry of the monument. Each floor was made of different dimensions, and divided by **outlines and inlines** to make it look realistic.

The **windows** were constructed using the same approach as used in **floors** i.e. the **top-down bisectional approach across the central structure.** Each floor has separate dimensions of windows. These were created by using repetitive **for-loops** on both sides of the **central structure.**

The **coloring** of the structure was added at the end after the entire structure appears on the screen to add to the flair of the monument. **floodfill, fillpoly, and setfillstyle** functions are widely used for coloring the structure. Each floor appears to be colored in a **top-down** approach with the central structure colored at the end.

Finally, **LA CASA DE PAPEL** appears on the screen. **outtextxy and settextstyle** functions were used for the same.

IMPLEMENTATION:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
int xmid, ymid;

//center//
void centerstruct()
{
   int poly[8];
   rectangle(xmid-95,ymid-70,xmid+95,ymid-20);// biggest rect
   rectangle(xmid-90,ymid-80,xmid+90,ymid-70);//  upper rect
   line(xmid-115,ymid-65,xmid-115,ymid-20); //central line
   line(xmid+115,ymid-65,xmid+115,ymid-20);//   central line
   delay(200);
   rectangle(xmid-20,ymid-50,xmid+20,ymid-30);// center gray rect
   setfillstyle(7,LIGHTGRAY);
   delay(200);
   floodfill(xmid-10,ymid-35,WHITE);
   rectangle(xmid-125,ymid-20,xmid+125,ymid-15); //  below base rect
   rectangle(xmid-115,ymid-15,xmid+115,ymid);  // above pillar
rectangle(xmid-60,ymid+105,xmid+60,ymid+140);   //door
   poly[0]=xmid-60;
   poly[1]=ymid+105;
   poly[2]=xmid+60;
   poly[3]=ymid+105;
   poly[4]=xmid+60;
   poly[5]=ymid+140;
   poly[6]=xmid-60;
   poly[7]=ymid+140;
   setfillstyle(2,DARKGRAY);
   fillpoly(4,poly);
}

//pillars//
void pillars()
{
```

```
int i=0,j=0;
int poly1[8],poly2[8];
rectangle(xmid-95,ymid+82,xmid+96,ymid+87);  //hori incenter
delay(200);
setfillstyle(1,DARKGRAY);
        poly1[0]=xmid-95;
        poly1[1]=ymid+82;
        poly1[2]=xmid+96;
        poly1[3]=ymid+82;
        poly1[4]=xmid+96;
        poly1[5]=ymid+87;
        poly1[6]=xmid-95;
        poly1[7]=ymid+87;
        fillpoly(4,poly1);

for(j=0;j<4;j++)                //cascading pillars
{
        rectangle(xmid-110+i,ymid,xmid-95+i,ymid+140);
        rectangle(xmid-90+i,ymid,xmid-75+i,ymid+140);
        setfillstyle(6,WHITE);
        poly1[0]=xmid-110+i;
        poly1[1]=ymid;
        poly1[2]=xmid-95+i;
        poly1[3]=ymid;
        poly1[4]=xmid-95+i;
        poly1[5]=ymid+140;
        poly1[6]=xmid-110+i;
        poly1[7]=ymid+140;
        fillpoly(4,poly1);
        poly2[0]=xmid-90+i;
        poly2[1]=ymid;
        poly2[2]=xmid-75+i;
        poly2[3]=ymid;
        poly2[4]=xmid-75+i;
        poly2[5]=ymid+140;
        poly2[6]=xmid-90+i;
        poly2[7]=ymid+140;
        fillpoly(4,poly2);
        i=i+62;
```

```
        delay(200);
    }
    rectangle(xmid-112,ymid+140,xmid-93,ymid+150);  //left pillar base
    rectangle(xmid+94,ymid+140,xmid+113,ymid+150);  //right pillar base
}

//staircase//
void stairs()
{
    int j=0,i=0,k=0;
    rectangle(xmid-93,ymid+140,xmid+94,ymid+146);          //upper platform
    rectangle(xmid-93,ymid+146,xmid+94,ymid+148);  // stair1
    rectangle(xmid-93,ymid+148,xmid+94,ymid+150); //stair 2
    for(j=0;j<12;j++)                          //cascading stairs
    {
        rectangle(xmid-122-i,ymid+150+k,xmid+123+i,ymid+152+k);
        i=i+5;
        k=k+2;
        delay(200);
    }
}

//top floor
void f4()
{
    rectangle(xmid-310,ymid-75,xmid-90,ymid-70);  //topmost outline l
    rectangle(xmid+310,ymid-75,xmid+90,ymid-70);            //topmost outline r
    rectangle(xmid-315,ymid-70,xmid-95,ymid-65); //top inline l
    rectangle(xmid+315,ymid-70,xmid+95,ymid-65); //top inline r
    line(xmid-310,ymid-65,xmid-310,ymid-18);         //vert l out
    line(xmid+310,ymid-65,xmid+310,ymid-18);        //vert r out
    line(xmid-305,ymid-65,xmid-305,ymid-18);         //vert l in
    line(xmid+305,ymid-65,xmid+305,ymid-18);        //vert r in
}

//windowtopfloor
void winf4()
{       int i=0,j=0;
    for(j=0;j<5;j++)
```

```
   {
         rectangle(xmid-275+i,ymid-50,xmid-255+i,ymid-30);
         rectangle(xmid-272+i,ymid-47,xmid-258+i,ymid-33);
         line(xmid-265+i,ymid-47,xmid-265+i,ymid-33);
         i=i+30;
   }
   i=0;
   for(j=0;j<5;j++)
   {
         rectangle(xmid+275-i,ymid-50,xmid+255-i,ymid-30);
         rectangle(xmid+272-i,ymid-47,xmid+258-i,ymid-33);
         line(xmid+265-i,ymid-47,xmid+265-i,ymid-33);
         i=i+30;
   }
}

//big floor
void f3()
{
   line(xmid-315,ymid-18,xmid-125,ymid-18); //top hori l
   line(xmid-315,ymid-13,xmid-115,ymid-13); //bottom hori l
   line(xmid-315,ymid-18,xmid-315,ymid-13);   //vert l
   line(xmid+315,ymid-18,xmid+125,ymid-18);       //top h r
   line(xmid+315,ymid-13,xmid+115,ymid-13);       //bottom h r
   line(xmid+315,ymid-18,xmid+315,ymid-13);       // vert r
   line(xmid-310,ymid-13,xmid-310,ymid+80); //vert l out
   line(xmid+310,ymid-13,xmid+310,ymid+80); //vert r out
   line(xmid-305,ymid-13,xmid-305,ymid+80); // vert l in
   line(xmid+305,ymid-13,xmid+305,ymid+80); //vert r in
}

//window big floor
void winf3()
{    int i=0,j=0;
   for(j=0;j<4;j++)
   {
         rectangle(xmid-280+i,ymid,xmid-250+i,ymid+65);
         rectangle(xmid-277+i,ymid+3,xmid-253+i,ymid+62);
         line(xmid-265+i,ymid+3,xmid-265+i,ymid+62);
```

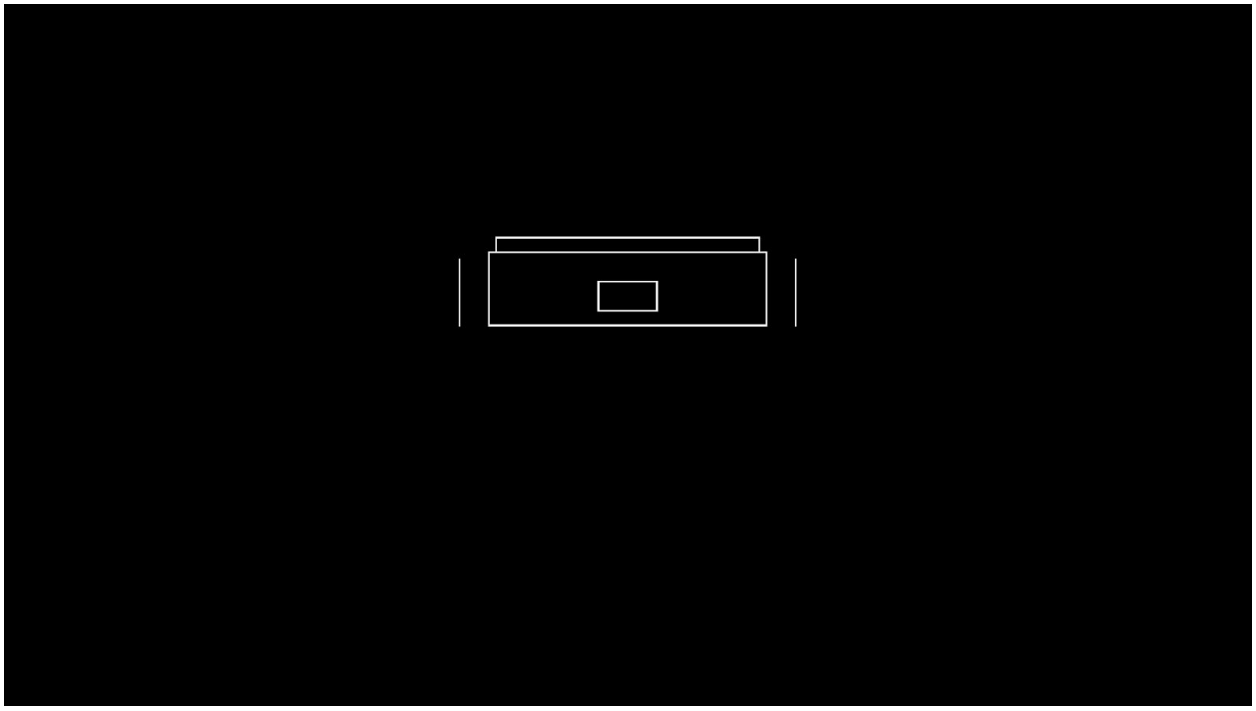
```
        i=i+40;
    }
    i=0;
    for(j=0;j<4;j++)
    {
        rectangle(xmid+280-i,ymid,xmid+250-i,ymid+65);
        rectangle(xmid+277-i,ymid+3,xmid+253-i,ymid+62);
        line(xmid+265-i,ymid+3,xmid+265-i,ymid+62);
        i=i+40;
    }
}

//first floor
void f2()
{
        rectangle(xmid-315,ymid+80,xmid-110,ymid+85);  //hori r
        rectangle(xmid+315,ymid+80,xmid+111,ymid+85);  //hori l
        rectangle(xmid-310,ymid+85,xmid-305,ymid+140);  //vert l
        rectangle(xmid+310,ymid+85,xmid+305,ymid+140);  //vert r

}

//window first floor
void winf2()
{       int i=0,j=0;
    for(j=0;j<5;j++)
    {
        rectangle(xmid-275+i,ymid+100,xmid-255+i,ymid+126);
        rectangle(xmid-272+i,ymid+103,xmid-258+i,ymid+123);
        line(xmid-265+i,ymid+103,xmid-265+i,ymid+123);
        i=i+30;
    }
    i=0;
    for(j=0;j<5;j++)
    {
        rectangle(xmid+275-i,ymid+100,xmid+255-i,ymid+126);
        rectangle(xmid+272-i,ymid+103,xmid+258-i,ymid+123);
        line(xmid+265-i,ymid+103,xmid+265-i,ymid+123);
        i=i+30;
```

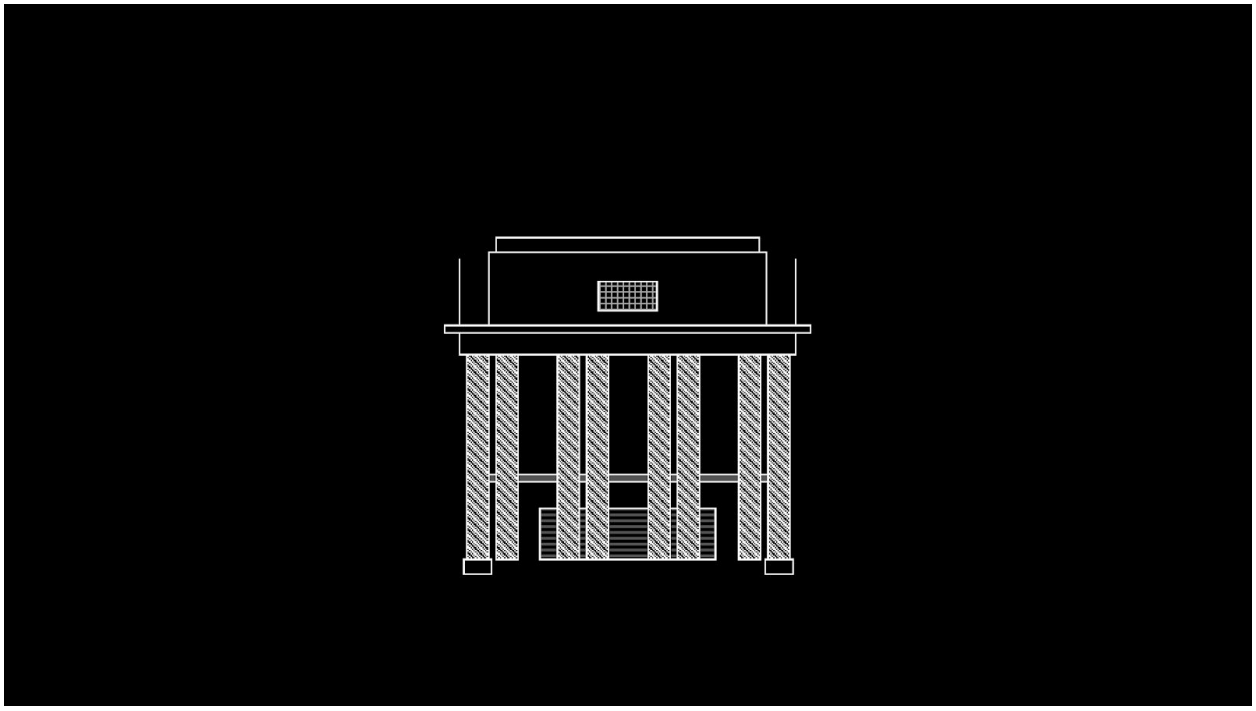
```
    }

}

//bottom floor
void f1()
{
        rectangle(xmid-315,ymid+140,xmid-112,ymid+145);  //hori r
        rectangle(xmid+315,ymid+140,xmid+113,ymid+145);  //hori l
        rectangle(xmid-310,ymid+145,xmid-305,ymid+170);  //vert l
        rectangle(xmid+310,ymid+145,xmid+305,ymid+170);  //vert r
        line(0,ymid+170,xmid*2,ymid+170); //bottom line
}

//colour fill
void fillfloor()
{
   setfillstyle(1,DARKGRAY);
   floodfill(xmid-300,ymid-60,WHITE);  //top l
   floodfill(xmid+300,ymid-60,WHITE);  //top r
   floodfill(xmid-300,ymid+150,WHITE);              //bottom l
   floodfill(xmid+300,ymid+150,WHITE);              // bottom r
   floodfill(xmid-300,ymid,WHITE);                  //big l
   floodfill(xmid+300,ymid,WHITE);                  // big r
   floodfill(xmid-300,ymid+90,WHITE);      //first l
   floodfill(xmid+300,ymid+90,WHITE);  //first r
   floodfill(xmid-250,ymid-73,WHITE); //terrace l
   floodfill(xmid+250,ymid-73,WHITE); //terrace r

}

void fillcenter()
{
   setfillstyle(1,LIGHTGRAY);
   floodfill(xmid-100,ymid-60,WHITE);       //l
   floodfill(xmid,ymid-60,WHITE);           //c
   floodfill(xmid+100,ymid-60,WHITE);       //r
   floodfill(xmid,ymid-10,WHITE); // above pillar
   floodfill(xmid,ymid+143,WHITE);  //platform above stairs
```

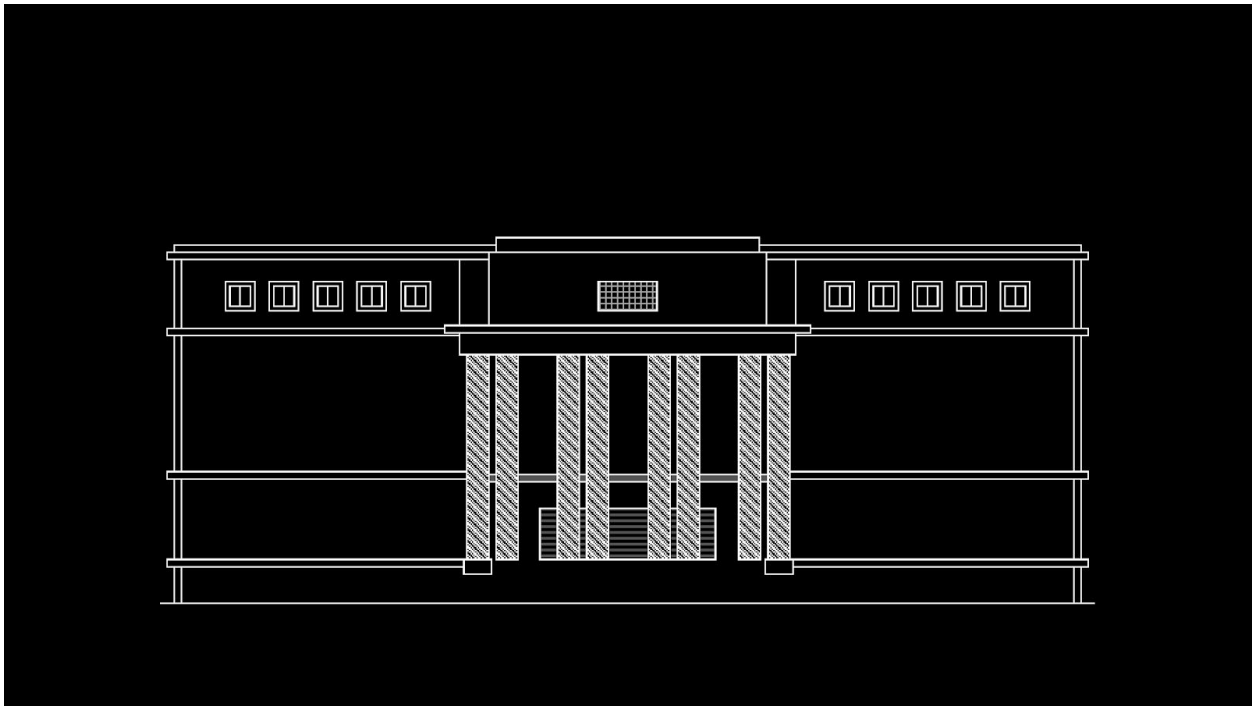
```
    floodfill(xmid,ymid-75,WHITE);    //topmost
    floodfill(xmid-100,ymid+145,WHITE);  // left pillar base
    floodfill(xmid+100,ymid+145,WHITE);   // right pillar base
}
int main(void)
{
   int gdriver = DETECT, gmode, errorcode;
   initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");
   errorcode = graphresult();

   if (errorcode != grOk)
   {
          printf("Graphics error: %s\n", grapherrormsg(errorcode));
          printf("Press any key to halt:");
          getch();
          exit(1);
   }

   xmid=getmaxx()/2;
   ymid=getmaxy()/2;

   centerstruct();
   delay(200);
   pillars();
   delay(200);
   f4();
   delay(200);
   f3();
   delay(200);
   f2();
   delay(200);
   f1();
   delay(200);
   winf4();
   delay(200);
   winf3();
   delay(200);
   winf2();
   delay(200);
```

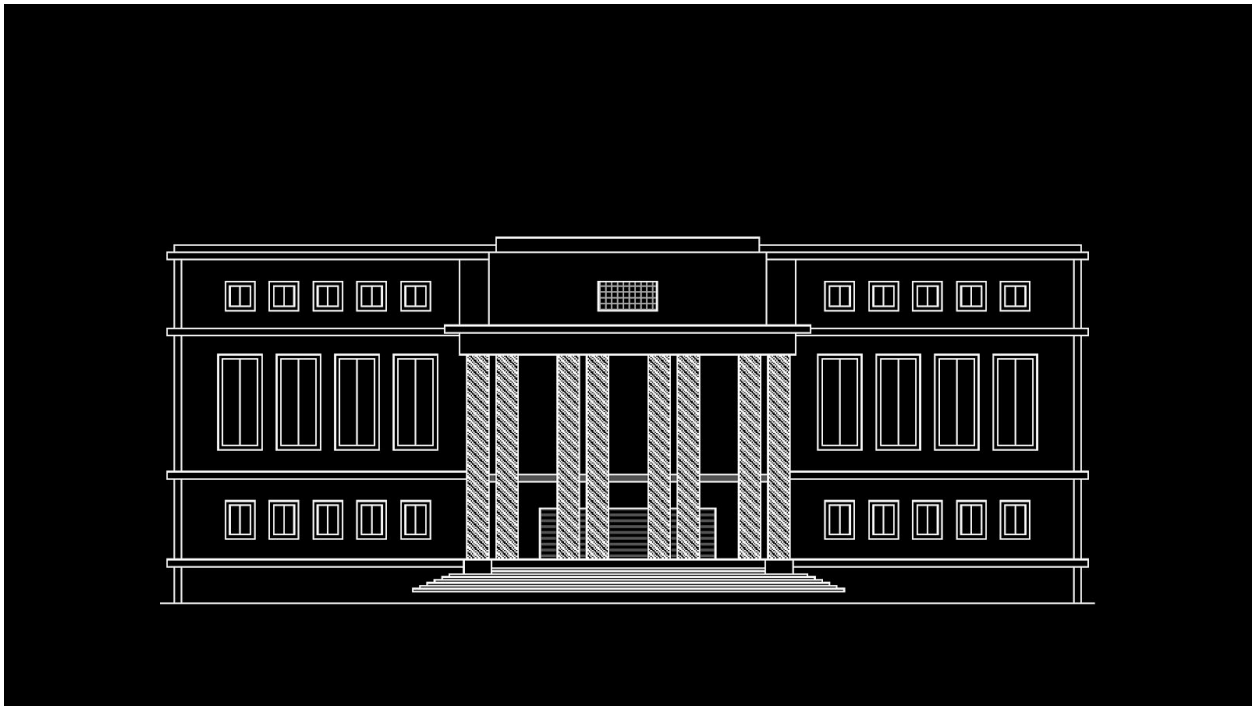
```
  stairs();
  delay(200);
  fillcenter();
  delay(200);
  fillfloor();
  delay(200);

  settextstyle(8,0,5);
  setcolor(RED);
  outtextxy(110,40,"LA CASA DE PAPEL");
  getch();
  cleardevice();


 setcolor(13);
 rectangle(1,1,639,479);
 rectangle(15,15,624,464);
 setfillstyle(6,11);
 floodfill(2,2,13);
 settextstyle(7,0,3);
 setcolor(CYAN);
 outtextxy(150,100,"COMPUTER GRAPHICS PROJECT");
 outtextxy(150,200,"MADE BY :KUNAL AND SHREEJA") ;
 outtextxy(175,300,"CLASS: D7B \t\t BATCH:B");
 outtextxy(175,400,"ROLL NO: 32 & 39");
 getch();
 closegraph();
 return 0;
}
```

SNAPSHOTS:

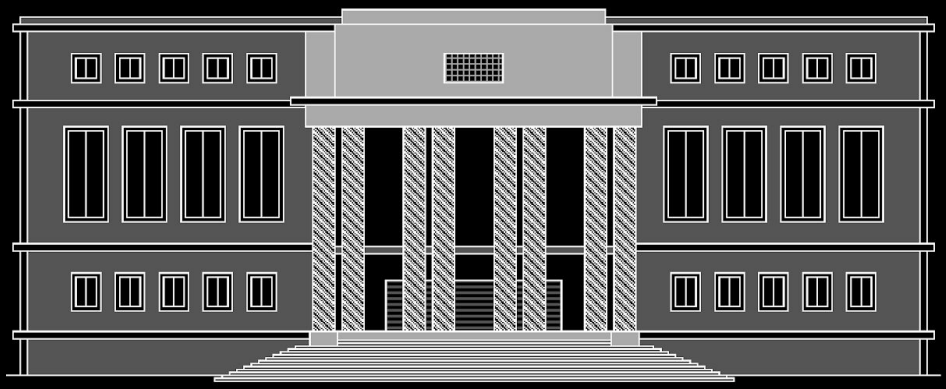








LA CASA DE PAPEL



COMPUTER GRAPHICS PROJECT

MADE BY :KUNAL AND SHREEJA

CLASS: D7B BATCH:B

ROLL NO: 32 & 39

CONCLUSION:

Creating this project gave us an insight into the vastness of Computer Graphics as a subject and its numerous applications on a daily basis. The project, although seemingly looks quite easy, yet quite a few complications were faced on account of its intricate symmetry. A few details were dropped out as compared to the original model, yet the efforts to create an authentic structure have been displayed quite successfully.