



**SANJAY GHODAWAT INSTITUTE**

Approved by A.I.C.T.E. New Delhi, and Recognized by DTE Mumbai, Govt. of Maharashtra

**SANJAY GHODAWAT INSTITUTE**

**ATIGRE**



**Third semester**

**(Year 2025-26)**

**Micro Project**

**Computer Graphics (313001)**

**Title of this Project –“ 2D CAR RACING “**

**Program:** Computer Science and Engineering (CW3K)

**Members of the Group**

<b>1.Piyush Lohar</b>	<b>Roll No:-33</b>
<b>2.Aryan Awale</b>	<b>Roll No:-26</b>
<b>3.Aditya Kamble</b>	<b>Roll No:-40</b>
<b>4.Shree Kshrisagar</b>	<b>Roll No:-43</b>



**SANJAY GHODAWAT INSTITUTE**

Approved by A.I.C.T.E. New Delhi, and Recognized by DTE Mumbai, Govt. of Maharashtra

**Maharashtra State Board of Technical Education, Mumbai**

**CERTIFICATE**

This is to certify that the Micro project work entitled

**“ 2D CAR RACING ”**

Has been successfully completed by

**In fulfilment for the**

**Diploma in Computer Science & Engineering**

**Maharashtra State Board of technical education**

**During the academic year 2025-26 under the guidance of**

**Ms. K. C. Bohara**

**Mr. S. V. chavan**

**Project Guide**

**H.O. D**

**Dr. V. V. Giri**

**Principal**



**SANJAY GHODAWAT INSTITUTE**

Approved by A.I.C.T.E. New Delhi, and Recognized by DTE Mumbai, Govt. of Maharashtra

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

Program Code: CW3K

Course Name: Computer Graphics

Course Code: 313001

Project Title: 2D CAR RACING

Sr. No.	Roll No.	Name of Student	Enrollment No.	Exam Seat No.	Micro Project Marks		Total
					G	I	
1	33	Piyush Lohar	24213440494				
2	26	Aryan Awale	24213440486				
3	40	Aditya Kamble	24213440501				
4	43	Shree Kshrisagar	24213440505				

**Ms. K. C. Bohara****Course faculty & Signature**

**MAHARASHTRA STATE BOARD OF TECHNICAL  
EDUCATION, MUMBAI****MICRO PROJECT****Progress Report / Weekly Report****Name of the Project:**2D CAR RACING**Course:** Computer Graphics (313001)**Program:** Computer Science and Engineering (CW3K)**Roll No:**33, 26, 40, 43

Week No.	Date	Duration in Hrs.	Work / Activity Performed	Sign of the faculty
1.	05-08-2025	2 <sup>1/2</sup>	Studied basics of computer graphics and understood project requirement	
2.	14-08-2025	3	Design game logic and flow for car and race	
3.	19-08-2025	2	Implement graphics initialization and basics shapes using C graphics	
4.	25-08-2025	2	Developed car movement and keyboard controls	
5.	07-09-2025	3 <sup>1/2</sup>	Added car object, collision detection, and time logic	
6.	21-09-2025	2	Tested game for errors, fixed bugs, and improved performance	
7.	02-10-2025	1	Prepared project report, conclusion, and final documentation	



## Teacher Evaluation Sheet for Micro Project

**Course Title and Code:** - Computer Graphics (313001)

**Title of the Project:** - 2D CAR RACING

**Group No:** - 09

CO 1:	Create and manipulate digital animations using computer graphics techniques
CO 2:	Apply visual and geometric concepts to design animated components
CO 3:	Use animation software to represent complex ideas in a visual format
CO 5:	Develop basic programming solutions for animated projects or graphics
CO 6:	Present project results clearly through digital visualizations and presentations

**Addressed by the Micro Project:**



## **Acknowledgement**

I would like to express my special thanks of gratitude to my Respected Mam, Ms. K. C. Bohara and HOD Mr. S. V. Chavan for providing guidance and support throughout the process of completing this project and helped me in doing a lot of research and I came to know about so many new things. I am also thankful to my principal Dr. V. V. Giri for allowing me the opportunity to explore and work on this project in the college environment.

Secondly, I would like to thank my parents and classmates who helped me a lot in finalizing this project within the limited time frame.



## TABLE OF CONTENTS

<b>Sr. No.</b>	<b>Content</b>	<b>Page No.</b>
<b>1.</b>	<b>Abstract</b>	<b>1</b>
<b>2.</b>	<b>Introduction</b>	<b>2</b>
<b>3.</b>	<b>Objective</b>	<b>3</b>
<b>4.</b>	<b>Algorithm</b>	<b>4</b>
<b>4.</b>	<b>Tags used in webpage</b>	<b>5</b>
<b>5.</b>	<b>Technology used</b>	<b>6</b>
<b>6.</b>	<b>Code</b>	<b>7,8,9,10,11</b>
<b>7.</b>	<b>Flowchart</b>	<b>12</b>
<b>8.</b>	<b>Output</b>	<b>13,14</b>
<b>9.</b>	<b>Conclusion</b>	<b>15</b>
<b>10.</b>	<b>Reference</b>	<b>16</b>

## **Abstract:**

This project presents a 2D Car Racing Game implemented in C using the graphics.h (WinBGIm) library. The game simulates a racing environment with three cars: one controlled by the player and two computer-controlled opponents. The race occurs on a predefined track surrounded by a cartoon-styled audience, enhancing visual appeal. The player controls the car using arrow keys: Left/Right to move horizontally, Up to accelerate, and Down to brake. Opponents move forward automatically with random lane adjustments and speed variations, creating unpredictability. The first car to cross the finish line is declared the winner. Upon winning, a celebration animation with fireworks is displayed.

### **Key features include:**

- Detailed car and track rendering,
- Audience visualization with cartoon figures,
- Dynamic opponent behavior,
- Real-time speed display,
- Interactive control system, and
- Celebration effects on victory.

This project demonstrates computer graphics programming concepts such as object drawing, animation, event handling, and randomness to create an engaging and interactive racing game.



## **Introduction:**

This project is a 2D Car Racing Game developed using the C programming language with the graphics.h (WinBGIm) library. The aim of the project is to simulate a simple racing environment where the player competes against two computer-controlled opponent cars on a racing track.

### **The game provides:**

1. A detailed car design with headlights, taillights, wheels, and roof.
2. A track with boundaries, finish line, and audience graphics to make the race visually engaging.
3. Keyboard controls for the player (arrow keys for movement, acceleration, and braking).
4. Opponent cars with random speed and position changes, creating a competitive and unpredictable racing experience.
5. A celebration screen with animated fireworks when a winner is declared.

This project demonstrates the use of basic graphics programming concepts such as shapes, colors, animation, keyboard handling, and randomization. It is designed to give players an interactive and fun racing experience while showcasing how computer graphics can be applied to develop simple 2D games.



## Objective:

The main objective of this project is to design and implement a 2D Car Racing Game using C graphics that provides an interactive and engaging experience for the player.

### The program aims to:

- Simulate a racing environment with a track, audience, and multiple competing cars.
- Allow user control of the player's car through keyboard inputs (acceleration, braking, and lane shifting).
- Incorporate computer-controlled opponents with random movement and varying speeds to create competition and unpredictability.
- Detect the winner by determining which car crosses the finish line first.
- Enhance user engagement through visual elements such as detailed car designs, animated audience, and celebration fireworks.
- Demonstrate practical application of computer graphics concepts including drawing, animation, event handling, and randomization in C programming.

### Algorithm:-

Step 1: Start the program.  
Step 2: Initialize graphics mode and random seed.  
Step 3: Create and initialize cars:  
Player's car (position, color, speed).  
Opponent cars (positions, colors, speed).  
Step 4: Define the finish line Y-coordinate.  
Step 5: Repeat until any car crosses the finish line:  
Clear screen.  
Draw track, audience, and controls.  
Display finish line.  
Update opponent car positions (forward and slight left-right movement).  
Restrict cars within track boundaries.  
Update player's car:  
Automatic forward movement.  
Small random drift.  
Keep inside track boundaries.  
Draw all cars.  
Check winner:  
If player crosses finish → Winner = 1.  
If opponent1 crosses finish → Winner = 2.  
If opponent2 crosses finish → Winner = 3.  
Randomly vary opponent car speeds.  
Show speed info on screen.  
Apply keyboard controls:  
← (left arrow): Move player left.  
→ (right arrow): Move player right.  
↑ (up arrow): Accelerate forward.  
↓ (down arrow): Brake (slow down).  
ESC: Exit game.  
Step 6: When a winner is detected → stop the loop.  
Step 7: Display celebration screen with fireworks for the winner.  
Step 8: Wait for key press and close graphics window.  
Step 9: End program.



## Tags used in webpage

### 1. Header Files (Tags for Libraries)

#include <graphics.h> – for graphics functions (WinBGI).

#include <conio.h> – for getch(), kbhit() input functions.

#include <stdlib.h> – for rand(), srand().

#include <stdio.h> – for printf().

#include <time.h> – for random seed (time(NULL)).

### 2. User-Defined Structure

```
typedef struct { int x, y; int color; int speed; } Car;
```

Defines a Car object with position, color, and speed.

### 3. Functions Defined in Code

drawDetailedCar(Car c) – draws a car with body, roof, wheels, lights.

drawCartoonPerson(...) – draws a simple cartoon audience person.

drawAudience() – draws multiple spectators on both sides of the track.

drawTrack() – draws track boundary, markings, and finish line.

drawControls() – shows game instructions.

celebrate(int winner) – displays winning message with fireworks animation.

main() – entry point where game logic and loop are implemented.

### 4. Graphics Functions (Tags from graphics.h)

initgraph() – initializes graphics mode.

closegraph() – closes graphics window.

cleardevice() – clears screen.

setcolor() – sets drawing color.

setfillstyle() – sets fill pattern and color.

bar() – draws filled rectangle.

rectangle() – draws rectangle outline.

arc() – draws an arc.

line() – draws line.

fillellipse() – draws filled ellipse.

outtextxy() – prints text at a given position.

settextstyle() – sets font style/size.

putpixel() – sets color at a pixel.

delay() – pauses execution (used for animation).

### 5. Console & Input Functions

getch() – waits for a key press.

kbhit() – checks if a key was pressed.

### 6. Utility Functions

rand() – generates random numbers (for opponents & fireworks).

srand() – seeds random generator.



### **Technology used**

1. DELL OPTIPLEX 3020
2. WINDOWS 10
3. ASUS
4. DEV C++
5. GRAPHICS LIBRARY



## Code:-

```
#include <graphics.h>
#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
typedef struct {
    int x, y;    // top-left of car bounding box
    int color;   // body color
    int speed;   // forward speed (pixels/frame)
} Car;

void drawDetailedCar(Car c) {
    int x = c.x, y = c.y;
    setfillstyle(SOLID_FILL, DARKGRAY);
    bar(x+5, y+30, x+45, y+36);
    setcolor(c.color);
    setfillstyle(SOLID_FILL, c.color);
    bar(x, y+10, x+50, y+30);    // main body
    setfillstyle(SOLID_FILL, LIGHTGRAY);
    bar(x+10, y-5, x+40, y+12);    // roof
    setfillstyle(SOLID_FILL, BLUE);
    bar(x+12, y-3, x+38, y+8);
    setfillstyle(SOLID_FILL, YELLOW);
    fillellipse(x+4, y+18, 4, 3);    // left headlight
    setfillstyle(SOLID_FILL, RED);
    fillellipse(x+46, y+14, 4, 3);    // taillight
    setcolor(BLACK);
    setfillstyle(SOLID_FILL, BLACK);
    fillellipse(x+12, y+34, 6, 6);
    fillellipse(x+38, y+34, 6, 6);
    setfillstyle(SOLID_FILL, DARKGRAY);
    bar(x+36, y-6, x+44, y-4);
}

void drawCartoonPerson(int cx, int cy, int headColor, int shirtColor, int pantColor, int raiseHands) {
    setfillstyle(SOLID_FILL, headColor);
    fillellipse(cx, cy, 8, 8);
    setcolor(BLACK);
    putpixel(cx-2, cy-1, BLACK);
    putpixel(cx+2, cy-1, BLACK);
    setcolor(BLACK);
    arc(cx, cy+3, 200, 340, 4);
}
```



```
bar(cx-8, cy+8, cx+8, cy+24);
setfillstyle(SOLID_FILL, pantColor);
bar(cx-8, cy+24, cx+8, cy+36);
setcolor(BLACK);
line(cx-4, cy+36, cx-4, cy+42);
line(cx+4, cy+36, cx+4, cy+42);
setcolor(BLACK);
if (raiseHands) {
    line(cx-8, cy+12, cx-16, cy-4);
    line(cx+8, cy+12, cx+16, cy-4);
    setfillstyle(SOLID_FILL, headColor);
    fillellipse(cx-16, cy-4, 3, 3);
    fillellipse(cx+16, cy-4, 3, 3);
} else {
    line(cx-8, cy+12, cx-16, cy+20);
    line(cx+8, cy+12, cx+16, cy+20);
}
}
void drawAudience() {
    int y;
    int idx = 0;
    for (y = 70; y < 430; y += 36) {
        int skin = (idx % 3 == 0) ? LIGHTRED : ((idx % 3 == 1) ? BROWN : LIGHTBLUE);
        int shirt = (idx % 4 == 0) ? YELLOW : ((idx % 4 == 1) ? RED : ((idx % 4 == 2) ? GREEN : MAGENTA));
        int pant = (idx % 2 == 0) ? BLUE : DARKGRAY;
        drawCartoonPerson(40, y, skin, shirt, pant, (idx % 2));
        int skin2 = ((idx+1) % 3 == 0) ? LIGHTRED : (((idx+1) % 3 == 1) ? BROWN : LIGHTBLUE);
        int shirt2 = ((idx+1) % 4 == 0) ? CYAN : (((idx+1) % 4 == 1) ? RED : (((idx+1) % 4 == 2) ? BLUE :
MAGENTA));
        int pant2 = ((idx+1) % 2 == 0) ? GREEN : DARKGRAY;
        drawCartoonPerson(600, y, skin2, shirt2, pant2, ((idx+1) % 2));
        idx++;
    }
}
void drawTrack() {
    setcolor(WHITE);
    rectangle(60, 30, 580, 430);
    setcolor(LIGHTGRAY);
    rectangle(120, 90, 520, 370);
    setcolor(WHITE);
    for (int x = 130; x <= 510; x += 30) {
        line(x, 200, x+12, 200);
        line(x, 260, x+12, 260);
    }
    setcolor(WHITE);
```



```
arc(120, 90, 180, 270, 30);
arc(520, 90, 270, 360, 30);
arc(120, 370, 90, 180, 30);
arc(520, 370, 0, 90, 30);
for (int i = 120; i <= 520; i += 12) {
    if (((i - 120) / 12) % 2 == 0) setcolor(WHITE); else setcolor(BLACK);
    line(i, 60, i, 80);
}
setcolor(YELLOW);
settextstyle(DEFAULT_FONT, HORIZ_DIR, 2);
outtextxy(220, 5, "CGR: 2D Car Racing");
}
void drawControls() {
    setcolor(LIGHTGREEN);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(10, 450, "Controls: Left <- Right -> Up = Accelerator Down = Brake (ESC to quit)");
}
void celebrate(int winner) {
    cleardevice();
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 3);
    setcolor(YELLOW);
    char msg[80];
    sprintf(msg, "Car %d WINS!", winner);
    outtextxy(180, 30, msg);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(180, 80, "Celebration fireworks... press any key to exit");
    for (int b = 0; b < 60; ++b) {
        int cx = 80 + rand() % 480;
        int cy = 100 + rand() % 260;
        int petals = 6 + rand() % 8;
        int col = 1 + rand() % 15;
        setcolor(col);
        for (int p = 0; p < petals; ++p) {
            int rx = 6 + rand() % 40;
            int ry = 6 + rand() % 40;
            int ex = cx + (rand() % rx) - rx/2;
            int ey = cy + (rand() % ry) - ry/2;
            line(cx, cy, ex, ey);
            fillellipse(ex, ey, 2 + rand()%4, 2 + rand()%4);
        }
        delay(80);
    }
    getch();
}
```





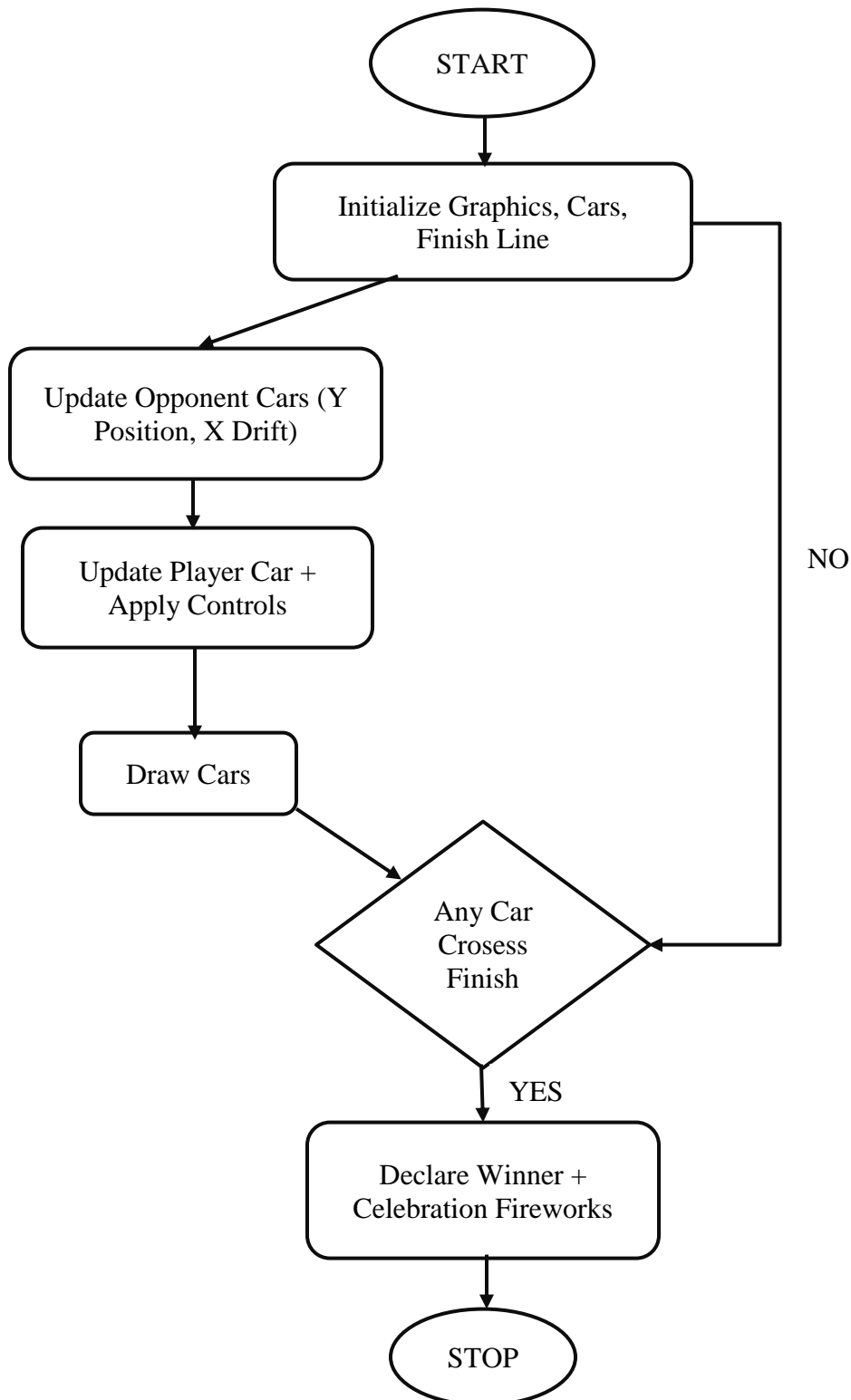
```
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, ""); // Assumes WinBGI is installed
    srand((unsigned)time(NULL));
    Car player = {200, 360, RED, 2};
    Car opp1 = {320, 390, BLUE, 2};
    Car opp2 = {440, 420, GREEN, 2};
    int finishY = 80; // Y coordinate threshold of finish line
    int winner = 0;
    int ch;
    while (1) {
        cleardevice();
        drawTrack();
        drawAudience();
        drawControls();
        setcolor(WHITE);
        outtextxy(260, 50, "FINISH");
        opp1.y -= opp1.speed;
        opp2.y -= opp2.speed;
        opp1.x += (rand() % 7) - 3;
        opp2.x += (rand() % 7) - 3;
        if (opp1.x < 140) opp1.x = 140;
        if (opp1.x > 440) opp1.x = 440;
        if (opp2.x < 140) opp2.x = 140;
        if (opp2.x > 440) opp2.x = 440;
        player.y -= player.speed;
        player.x += (rand() % 5) - 2;
        if (player.x < 140) player.x = 140;
        if (player.x > 440) player.x = 440;
        if (player.y < 40) player.y = 40;
        if (player.y > 420) player.y = 420;
        drawDetailedCar(player);
        drawDetailedCar(opp1);
        drawDetailedCar(opp2);
        if (player.y <= finishY) { winner = 1; break; }
        if (opp1.y <= finishY) { winner = 2; break; }
        if (opp2.y <= finishY) { winner = 3; break; }
        if ((rand() % 100) < 4) opp1.speed = 1 + rand() % 3;
        if ((rand() % 100) < 4) opp2.speed = 1 + rand() % 3;
        char tmp[64];
        sprintf(tmp, "You: %d", player.speed);
        outtextxy(10, 30, tmp);
        sprintf(tmp, "Blue: %d", opp1.speed);
        outtextxy(540, 30, tmp);
        sprintf(tmp, "Green: %d", opp2.speed);
```



```
outtextxy(540, 14, tmp);
delay(40);
if (kbhit()) {
    ch = getch();
    if (ch == 0 || ch == 0xE0) { // arrow key prefix
        ch = getch();
        switch (ch) {
            case 75: // left
                player.x -= 18;
                if (player.x < 140) player.x = 140;
                break;
            case 77: // right
                player.x += 18;
                if (player.x > 440) player.x = 440;
                break;
            case 72: // up - accelerate
                player.speed = 4;
                // small boost forward immediately
                player.y -= 12;
                if (player.y < 40) player.y = 40;
                break;
            case 80: // down - brake
                player.speed = 1;
                player.y += 12;
                if (player.y > 420) player.y = 420;
                break;
            default:
                break;
        }
    } else {
        // ESC to quit
        if (ch == 27) { closegraph(); return 0; }
    }
}
celebrate(winner);
closegraph();
return 0;
}
```



## Flowchart:-





# SANJAY GHODAWAT INSTITUTE

Approved by A.I.C.T.E. New Delhi, and Recognized by DTE Mumbai, Govt. of Maharashtra

## Output





**SANJAY GHODAWAT INSTITUTE**

Approved by A.I.C.T.E. New Delhi, and Recognized by DTE Mumbai, Govt. of Maharashtra



## **Conclusion:**

The 2D Car Racing Game successfully demonstrates the use of C language and computer graphics programming to create an interactive and entertaining application. Through the implementation of cars, racing tracks, animated audience, and celebration effects, the project highlights how fundamental graphics functions can be combined with logic, randomization, and user input handling to simulate a real-time game environment.

### **The project meets its objectives by:**

1. Providing player control using keyboard inputs.
2. Introducing opponent cars with dynamic behavior for competition.
3. Ensuring smooth animations and boundary restrictions.
4. Displaying a celebration sequence for the winning car.

This project not only enhances understanding of graphics.h (BGI/WinBGIm) library and event-driven programming in C but also develops problem-solving skills in areas such as animation, real-time control, and game logic.



## Reference:

1. <https://www.google.com/search?client=firefox-b-e&q=draw+.io>
2. <https://www.google.com/search?client=firefox-b-e&q=algorithm>
3. <https://www.google.com/search?client=firefox-b-e&q=lovepdf>
4. <https://www.google.com/search?client=firefox-b-e&q=shiksha>