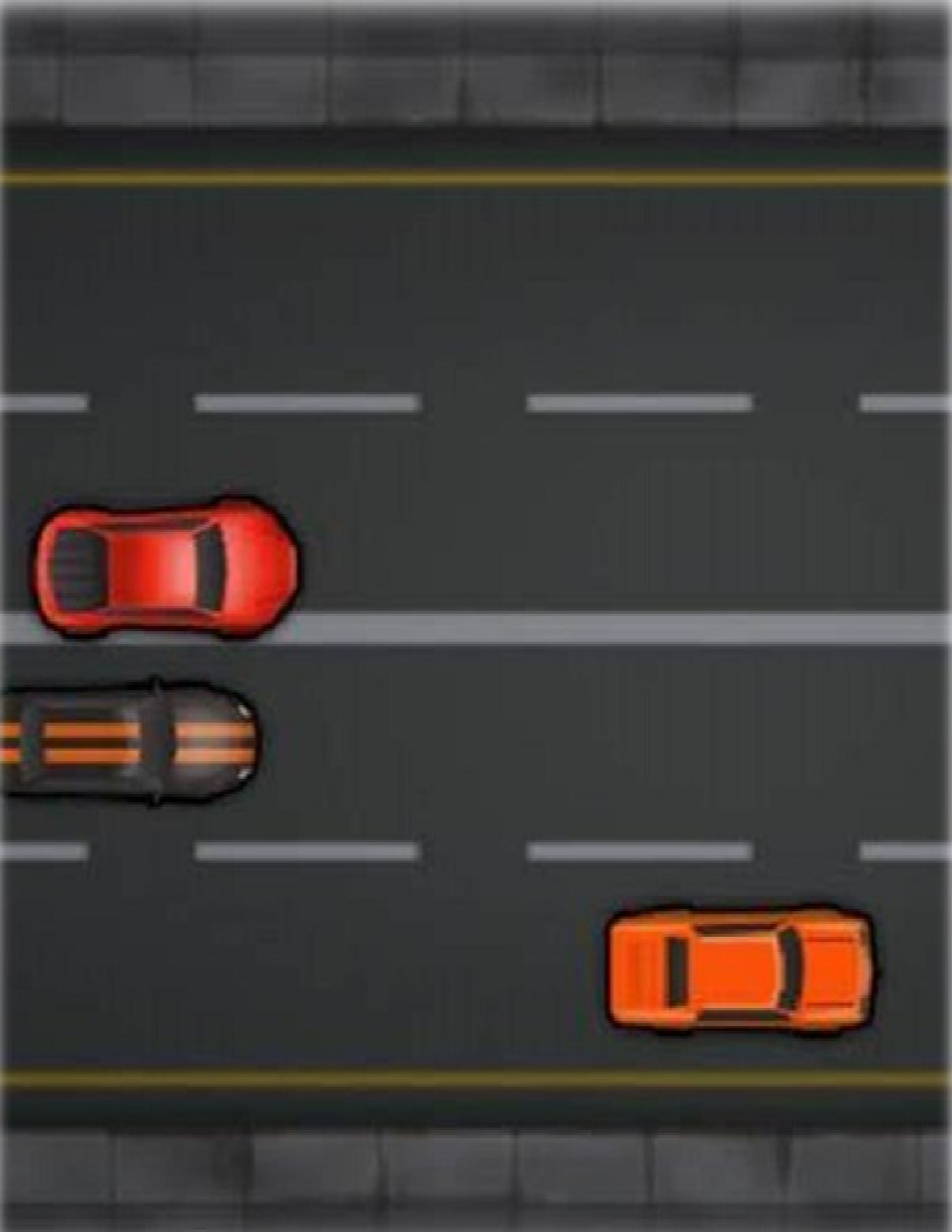

بسم الله الرحمن الرحیم

پروژه ماشین بازی

درس برنامه سازی پیشرفته

امیرحسین حنیفی

مقدمه

Car gameدر این پروژه قصد داریمبه سبک کنسول های قدیمی،بازی

. را پیاده سازی کنیم

شرح پروژه

در این بازی با استفاده ازمفاهیم شی گرایی و کالسبه صورت متداولماشین هایی با موقعیت هایمتفاوت (رندم) وارد صفحه میشوند کهسرعت انها نیز متفاوت است وشما
. باید بدون اصابت کردن،آن ها را رد کنید
توجه کنید که بایدترفند هایی پیاده کنید کهامکانبسته شدن راه توسط ماشین های
. دیگر وجود نداشته باشد

```
#ifndef CARGAME_HPP
#define CARGAME_HPP

#include <iostream>
#include <windows.h>

#define SCREEN_WIDTH 90
#define SCREEN_HEIGHT 26
#define WIN_WIDTH 70
class CarGame {
private:
    HANDLE console;
    COORD CursorPosition;

    int enemyY[3];
    int enemyX[3];
    int enemyFlag[3];
    char car[4][4] = { ' ','*','*',' ',
                       '*','*','*','*',
                       ' ','*','*',' ',
                       '*','*','*','*' };

    int carPos;
    int score;

public:
    CarGame();

    void gotoxy(int x, int y);

    void setcursor(bool visible, DWORD size);

    void drawBorder();

    void genEnemy(int ind);

    void drawEnemy(int ind);

    void eraseEnemy(int ind);

    void resetEnemy(int ind);

    void drawCar();

    void eraseCar();

    int collision();

    void gameover();

    void updateScore();

    void instructions();

    void play();
```

```
    void menu();
};


#endif // CARGAME_HPP


#include <iostream>
#include <fstream>

int main() {
    CarGame carGame;
    carGame.menu();

    return 0;
}



#include<iostream>
#include<conio.h>
#include<dos.h>
#include <windows.h>
#include <time.h>
#include "carGame.hpp"

using namespace std;


    CarGame::CarGame() {
        console = GetStdHandle(STD_OUTPUT_HANDLE);
        carPos = WIN_WIDTH / 2;
        score = 0;
    }

    void CarGame:: gotoxy(int x, int y) {
        CursorPosition.X = x;
        CursorPosition.Y = y;
        SetConsoleCursorPosition(console, CursorPosition);
    }

    void CarGame:: setcursor(bool visible, DWORD size) {
        if (size == 0)
            size = 20;

        CONSOLE_CURSOR_INFO lpCursor;
        lpCursor.bVisible = visible;
        lpCursor.dwSize = size;
        SetConsoleCursorInfo(console, &lpCursor);
    }

    void CarGame::drawBorder() {
        for (int i = 0; i < SCREEN_HEIGHT; i++) {
            for (int j = 0; j < 17; j++) {
                gotoxy(0 + j, i); cout << "*";
                gotoxy(WIN_WIDTH - j, i); cout << "*";
            }
        }
        for (int i = 0; i < SCREEN_HEIGHT; i++) {
            gotoxy(SCREEN_WIDTH, i); cout << "*";
        }
    }

    void CarGame:: genEnemy(int ind) {
        enemyX[ind] = 17 + rand() % (33);
    }

    void CarGame:: drawEnemy(int ind) {
        if (enemyFlag[ind] == true) {
            gotoxy(enemyX[ind], enemyY[ind]);   cout << "****";
            gotoxy(enemyX[ind], enemyY[ind] + 1); cout << " ** ";
            gotoxy(enemyX[ind], enemyY[ind] + 2); cout << "****";
            gotoxy(enemyX[ind], enemyY[ind] + 3); cout << " ** ";
        }
    }
```

```
void CarGame:: eraseEnemy(int ind) {
    if (enemyFlag[ind] == true) {
        gotoxy(enemyX[ind], enemyY[ind]); cout << "    ";
        gotoxy(enemyX[ind], enemyY[ind] + 1); cout << "    ";
        gotoxy(enemyX[ind], enemyY[ind] + 2); cout << "    ";
        gotoxy(enemyX[ind], enemyY[ind] + 3); cout << "    ";
    }
}

void CarGame:: resetEnemy(int ind) {
    eraseEnemy(ind);
    enemyY[ind] = 1;
    genEnemy(ind);
}

void CarGame:: drawCar() {
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            gotoxy(j + carPos, i + 22); cout << car[i][j];
        }
    }
}

void CarGame:: eraseCar() {
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            gotoxy(j + carPos, i + 22); cout << " ";
        }
    }
}

int CarGame:: collision() {
    if (enemyY[0] + 4 >= 23) {
        if (enemyX[0] + 4 - carPos >= 0 && enemyX[0] + 4 - carPos < 9) {
            return 1;
        }
    }
    return 0;
}

void CarGame:: gameover() {
    system("cls");
    cout << endl;
    cout << "\t\t--------------------------" << endl;
    cout << "\t\t-------- Game Over -------" << endl;
    cout << "\t\t--------------------------" << endl << endl;
    cout << "\t\tPress any key to go back to menu.";
    getch();
}

void CarGame:: updateScore() {
    gotoxy(WIN_WIDTH + 7, 5); cout << "Score: " << score << endl;
}

void CarGame:: instructions() {
    system("cls");
    cout << "Instructions";
    cout << "\n----------------";
    cout << "\n Avoid Cars by moving left or right. ";
    cout << "\n\n Press 'a' to move left";
    cout << "\n Press 'd' to move right";
    cout << "\n Press 'escape' to exit";
    cout << "\n\nPress any key to go back to menu";
    getch();
}

void CarGame:: play() {
    carPos = -1 + WIN_WIDTH / 2;
    score = 0;
    enemyFlag[0] = 1;
    enemyFlag[1] = 0;
    enemyY[0] = enemyY[1] = 1;

    system("cls");
```

```
    drawBorder();
    updateScore();
    genEnemy(0);
    genEnemy(1);

    gotoxy(WIN_WIDTH + 7, 2); cout << "Car Game";
    gotoxy(WIN_WIDTH + 6, 4); cout << "----------";
    gotoxy(WIN_WIDTH + 6, 6); cout << "----------";
    gotoxy(WIN_WIDTH + 7, 12); cout << "Control ";
    gotoxy(WIN_WIDTH + 7, 13); cout << "-------- ";
    gotoxy(WIN_WIDTH + 2, 14); cout << " A Key - Left";
    gotoxy(WIN_WIDTH + 2, 15); cout << " D Key - Right";

    gotoxy(18, 5); cout << "Press any key to start";
    getch();
    gotoxy(18, 5); cout << "                      ";

    while (1) {
        if (kbhit()) {
            char ch = getch();
            if (ch == 'a' || ch == 'A') {
                if (carPos > 18)
                    carPos -= 4;
            }
            if (ch == 'd' || ch == 'D') {
                if (carPos < 50)
                    carPos += 4;
            }
            if (ch == 27) {
                break;
            }
        }

        drawCar();
        drawEnemy(0);
        drawEnemy(1);
        if (collision() == 1) {
            gameover();
            return;
        }
        Sleep(50);
        eraseCar();
        eraseEnemy(0);
        eraseEnemy(1);

        if (enemyY[0] == 10)
            if (enemyFlag[1] == 0)
                enemyFlag[1] = 1;

        if (enemyFlag[0] == 1)
            enemyY[0] += 1;

        if (enemyFlag[1] == 1)
            enemyY[1] += 1;

        if (enemyY[0] > SCREEN_HEIGHT - 4) {
            resetEnemy(0);
            score++;
            updateScore();
        }
        if (enemyY[1] > SCREEN_HEIGHT - 4) {
            resetEnemy(1);
            score++;
            updateScore();
        }
    }
}

void CarGame:: menu() {
    do {
        system("cls");
        gotoxy(10, 5); cout << " -------------------------- ";
        gotoxy(10, 6); cout << " |        Car Game        | ";
        gotoxy(10, 7); cout << " --------------------------";
```

```
        gotoxy(10, 9); cout << "1. Start Game";
        gotoxy(10, 10); cout << "2. Instructions";
        gotoxy(10, 11); cout << "3. Quit";
        gotoxy(10, 13); cout << "Select option: ";
        char op = getche();

        if (op == '1') play();
        else if (op == '2') instructions();
        else if (op == '3') exit(0);

    } while (1);
}
```

خروجی :