**Simple car game for beginners**

PRANJAL GUTA(24MIM10028)-Ideation and logic developing

SAKSHAM AGRAWAL(24BCE10492)-Managing system specific function(windows.h,.....) and implementation

DEV GUPTA(24BAI10577)-Managing UI Handling and execution

**CODE**

#include <iostream>

#include <windows.h>

#include <conio.h>

#include <time.h>

#define SCREEN\_WIDTH 110

#define SCREEN\_HEIGHT 35

#define WIN\_WIDTH 70

using namespace std;

HANDLE console = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD CursorPosition;

int enemyY[3];

int enemyX[3];

int enemyFlag[3];

char car[4][4] = {

' ','+','+',' ',

'+','+','+','+',

' ','+','+',' ',

'+','+','+','+'

};

int carPos = WIN\_WIDTH / 2;

int score = 0;

void gotoxy(int x, int y) {

CursorPosition.X = x;

CursorPosition.Y = y;

SetConsoleCursorPosition(console, CursorPosition);

}

void drawBorder() {

for (int i = 0; i < SCREEN\_HEIGHT; i++) {

gotoxy(0, i); cout << "|";

gotoxy(WIN\_WIDTH, i); cout << "|";

}

}

void genEnemy(int ind) {

enemyX[ind] = 18 + rand() % (WIN\_WIDTH - 36);

}

void drawEnemy(int ind) {

if (enemyFlag[ind]) {

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 eraseEnemy(int ind) {

if (enemyFlag[ind]) {

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 resetEnemy(int ind) {

eraseEnemy(ind);

enemyY[ind] = 1;

genEnemy(ind);

}

void 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 eraseCar() {

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

gotoxy(j + carPos, i + 22); cout << " ";

}

}

}

int collision() {

for (int i = 0; i < 3; i++) {

if (enemyFlag[i] && enemyY[i] + 3 >= 22) {

if (enemyX[i] >= carPos && enemyX[i] <= carPos + 3) {

return 1;

}

}

}

return 0;

}

void gameover() {

system("cls");

cout << "\n\t\t----------------------------" << endl;

cout << "\t\t---------GAME OVER----------" << endl;

cout << "\t\t----------------------------" << endl;

cout << "\t\tPress any key to go back to the menu.";

\_getch();

}

void updateScore() {

gotoxy(WIN\_WIDTH + 5, 5); cout << "Score: " << score;

}

void play() {

carPos = WIN\_WIDTH / 2;

score = 0;

enemyFlag[0] = 1;

enemyFlag[1] = 0;

enemyFlag[2] = 0;

enemyY[0] = enemyY[1] = enemyY[2] = 1;

system("cls");

drawBorder();

updateScore();

genEnemy(0);

genEnemy(1);

genEnemy(2);

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

\_getch();

gotoxy(18, 5); cout << " ";

while (1) {

if (GetAsyncKeyState(VK\_LEFT) && carPos > 18) carPos -= 4;

if (GetAsyncKeyState(VK\_RIGHT) && carPos < 50) carPos += 4;

if (GetAsyncKeyState(VK\_ESCAPE)) break;

drawCar();

for (int i = 0; i < 3; i++) drawEnemy(i);

if (collision()) { gameover(); return; }

Sleep(100);

eraseCar();

for (int i = 0; i < 3; i++) eraseEnemy(i);

for (int i = 0; i < 3; i++) {

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

if (enemyY[i] > SCREEN\_HEIGHT - 4) {

resetEnemy(i);

score++;

updateScore();

}

}

}

}

int main() {

srand((unsigned)time(NULL));

while (1) {

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. Quit";

gotoxy(10, 12); cout << "Select option: ";

char op = \_getch();

if (op == '1') play();

else if (op == '2') return 0;

}

}