**End War**



Session: 2022 – 2026

**Submitted by:**

Shahneela Iqbal 2022-CS-159

**Submitted to:**

Dr. Muhammad Awais Hassan

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Contents

[Short Description 3](#_Toc130345704)

[Game Characters Description 3](#_Toc130345705)

[Enemies 3](#_Toc130345706)

[Game Objects Description 4](#_Toc130345707)

[Rules & Interactions 4](#_Toc130345708)

[Goal of the Game 5](#_Toc130345709)

[Wireframes 5](#_Toc130345710)

[Data Structures 7](#_Toc130345711)

[Functions Prototypes 8](#_Toc130345712)

[Code 8](#_Toc130345713)

## Short Description

Tom is a player. He wants to explore the world. He is adventurous who wants to go on as many places as possible where there are hidden treasures and rewards. While on this journey, he has to face many obstacles and enemies who are determined to put him down with their evil intentions and tactics. The enemies disturb him and make it difficult for the Tom to do it. But he is very determined and he never loses hope. He always tries to do it.

He has to cross difficult roads with heavy traffic going left and right. Once he starts crossing them, he needs to beat fast paced vehicles as well as slow moving cars that are coming from both directions left and right. So, he has to look for both directions while crossing them that can crush him. There are different other obstacles that fall from above in vertical motion which makes it hard for him to avoid. He can pick the treasures along the way using his rope launcher or if he moves over them.

After many tries he succeeded in picks all the treasures and cross the boundary wall. If Tom dies 3 times in the game then he will lose the game. Tom inspires the player of new generation to never give up and always keep on try and one day you will achieve your goals.

## Game Characters Description

**Player**

There is one human player in the Game.

**Tom:**

Tom is the main character in the game and is known for his adventurous nature. He is inquisitive and loves to explore the world, always wanted to go outside and explore. He has a rope launcher that he holds dear because it has always helped him through difficult times. He can pick up different objects with it. Tom is brave, determined, and has a never-say-die spirit. He is the hero of the game, admired for his bravery and determination in the face of danger.

## Enemies

There are 2 enemies in the game.

**Cheetah:**

Cheetah is one of the four enemies in the game and is known for his aggressive speed. It is always in a rush. It always tries to hit the Tom. It is most difficult enemy to face.

**Turtle:**

Turtle is another enemy and is known for its slow speed. Tom takes it very easy.

## Game Objects Description

Following are the Objects in the Game

**Boundary Walls:**

There are two boundary walls in the game. One is the from which Tom starts his game and the other boundary wall is the one which Tom has to cross whiling going across the enemies.

**Treasures:**

Star is the components when Tom picks them 1 point adds to his score.

## Rules & Interactions

Tom can cross the road and picks all the star with the fire on the star. If he collides with the enemies his lives will decrease. If he picks the treasures and also cross the front boundary wall his score will increase.

## Goal of the Game

The goal of the game is to picks all the treasures and cross the front boundary wall while avoiding the enemies.

## Wireframes



Figure 1: Main Menu

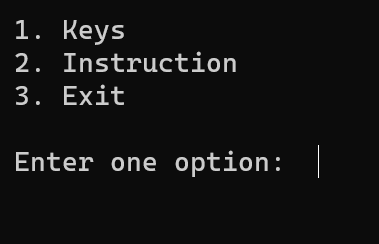


Figure 2: Option

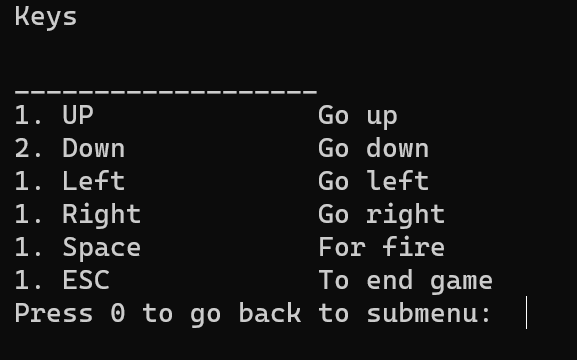


Figure 3:Keys

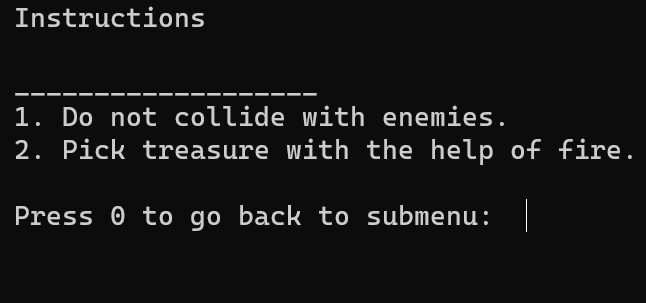
Figure 4: Instructions



Figure 5: Exit

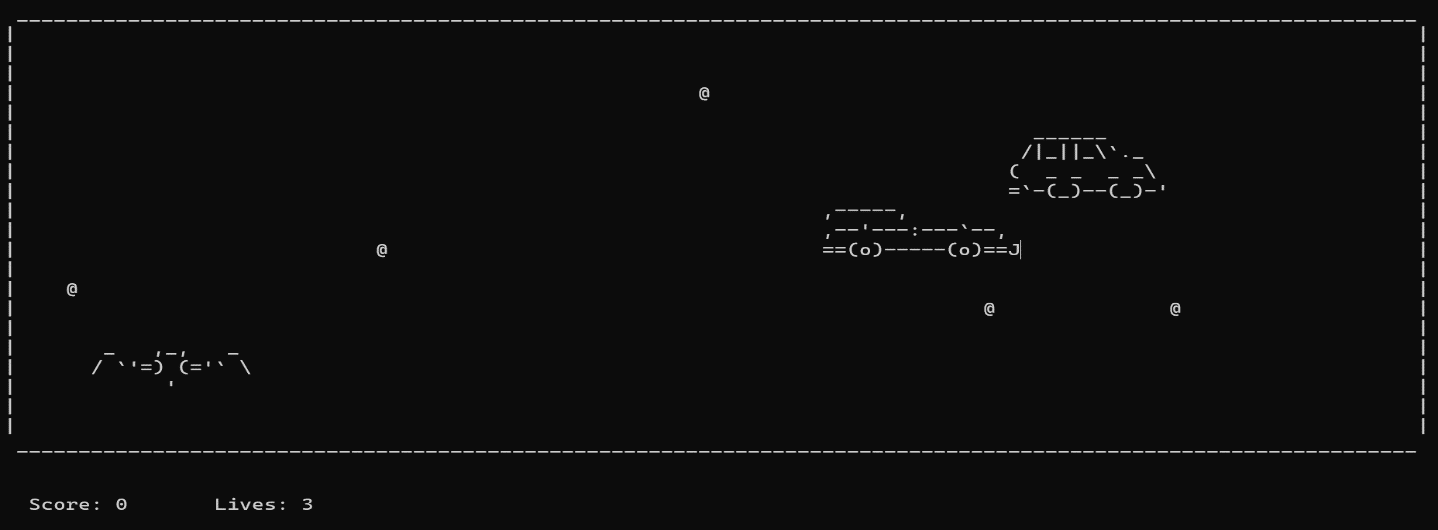


Figure 6: Game

## Data Structures

int fruitsX[5] = {10, 50, 65, 35, 100};

int fruitsY[5] = {5, 20, 15, 18, 1};

char tom1[15] = {' ', ' ', '\_', ' ', ' ', ' ', ',', '\_', ',', ' ', ' ', ' ', '\_', ' ', ' '};

char tom2[15] = {' ', '/', ' ', '`', '\'', '=', ')', ' ', '(', '=', '\'', '`', ' ', '\\', ' '};

char tom3[15] = {'/', '.', '-', '.', '-', '.', '\\', ' ', '/', '.', '-', '.', '-', '.', '\\'};

char tom4[15] = {' ', ' ', ' ', ' ', ' ', ' ', ' ', '\'', ' ', ' ', ' ', ' ', ' ', ' ', ' '};

char cheetah1[16] = {',', '-', '-', '-', '-', '-', ',', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '};

char cheetah2[16] = {',', '-', '-', '\'', '-', '-', '-', ':', '-', '-', '-', '`', '-', '-', ','};

char cheetah3[16] = {'=', '=', '(', 'o', ')', '-', '-', '-', '-', '-', '(', 'o', ')', '=', '=', 'J'};

char turtle1[13] = {' ', ' ', '\_', '\_', '\_', '\_', '\_', '\_', ' ', ' ', ' ', ' ', ' '};

char turtle2[13] = {' ', '/', '|', '\_', '|', '|', '\_', '\\', '`', '.', '\_', ' ', ' '};

char turtle3[13] = {'(', ' ', ' ', '\_', ' ', '\_', ' ', ' ', '\_', ' ', '\_', '\\', ' '};

char turtle4[13] = {'=', '`', '-', '(', '\_', ')', '-', '-', '(', '\_', ')', '-', '\''};

## Functions Prototypes

void **header**();

int **menu**();

int **option**();

int **keys**();

int **instruction**();

void **road**();

void **moveBullet**();

void **generateBullet**();

void **removeBulletFromArray**(int index);

void **gotoxy**(int x, int y);

void **moveCheetah**();

void **printCheetah**();

void **eraseCheetah**();

void **moveTurtle**();

void **printTurtle**();

void **eraseTurtle**();

void **printTom**();

char **getCharAtxy**(short int x, short int y);

void **generateFruits**();

void **removeFruits**();

void **printFruits**();

void **printScore**();

void **bulletCollision**();

void **printBullet**(int x, int y);

void **eraseBullet**(int x, int y);

void **moveTomUp**();

void **moveTomDown**();

void **moveTomLeft**();

void **moveTomRight**();

void **eraseTom**();

## Code

#include <iostream>

#include <windows.h>

#include <time.h>

#include <cstdlib>

using namespace **std**;

int enemyX = 4;

int enemyY = 10;

int tomX = 6;

int tomY = 18;

int TurX = 101;

int TurY = 6;

int bulletX[100];

int bulletY[100];

int bulletCount = 0;

int score = 0;

int lives = 3;

**string** enemyDirection = "Right";

**string** turtleDirection = "Left";

int timer = 0;

int turTimer = 0;

int fruitsX[5] = {10, 50, 65, 35, 100};

int fruitsY[5] = {5, 20, 15, 18, 1};

char tom1[15] = {' ', ' ', '\_', ' ', ' ', ' ', ',', '\_', ',', ' ', ' ', ' ', '\_', ' ', ' '};

char tom2[15] = {' ', '/', ' ', '`', '\'', '=', ')', ' ', '(', '=', '\'', '`', ' ', '\\', ' '};

char tom3[15] = {'/', '.', '-', '.', '-', '.', '\\', ' ', '/', '.', '-', '.', '-', '.', '\\'};

char tom4[15] = {' ', ' ', ' ', ' ', ' ', ' ', ' ', '\'', ' ', ' ', ' ', ' ', ' ', ' ', ' '};

char cheetah1[16] = {',', '-', '-', '-', '-', '-', ',', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '};

char cheetah2[16] = {',', '-', '-', '\'', '-', '-', '-', ':', '-', '-', '-', '`', '-', '-', ','};

char cheetah3[16] = {'=', '=', '(', 'o', ')', '-', '-', '-', '-', '-', '(', 'o', ')', '=', '=', 'J'};

char turtle1[13] = {' ', ' ', '\_', '\_', '\_', '\_', '\_', '\_', ' ', ' ', ' ', ' ', ' '};

char turtle2[13] = {' ', '/', '|', '\_', '|', '|', '\_', '\\', '`', '.', '\_', ' ', ' '};

char turtle3[13] = {'(', ' ', ' ', '\_', ' ', '\_', ' ', ' ', '\_', ' ', '\_', '\\', ' '};

char turtle4[13] = {'=', '`', '-', '(', '\_', ')', '-', '-', '(', '\_', ')', '-', '\''};

*// Functions declaration*

void **header**();

int **menu**();

int **option**();

int **keys**();

int **instruction**();

void **road**();

void **moveBullet**();

void **generateBullet**();

void **removeBulletFromArray**(int index);

void **gotoxy**(int x, int y);

void **moveCheetah**();

void **printCheetah**();

void **eraseCheetah**();

void **moveTurtle**();

void **printTurtle**();

void **eraseTurtle**();

void **printTom**();

char **getCharAtxy**(short int x, short int y);

void **generateFruits**();

void **removeFruits**();

void **printFruits**();

void **printScore**();

void **bulletCollision**();

void **printBullet**(int x, int y);

void **eraseBullet**(int x, int y);

void **moveTomUp**();

void **moveTomDown**();

void **moveTomLeft**();

void **moveTomRight**();

void **eraseTom**();

**main**()

{

**system**("cls");

**header**();

    int menuFun = **menu**();

    if (menuFun == 2)

    {

        while (true)

        {

            int optionFun = **option**();

            if (optionFun == 1)

            {

                int keyFun = **keys**();

                if (keyFun == 0)

                {

                    int optionFun = **option**();

                }

            }

            else if (optionFun == 2)

            {

                int instructionFun = **instruction**();

                if (instructionFun == 0)

                {

                    int optionFun = **option**();

                }

            }

            else if (optionFun == 3)

            {

**system**("cls");

                int menuFun = **menu**();

            }

        }

    }

    if (menuFun == 3)

    {

        cout **<<** "You have successfully exit the game";

    }

    if (menuFun == 1)

    {

**system**("cls");

**road**();

**generateFruits**();

**printFruits**();

**printCheetah**();

**printTurtle**();

**printTom**();

        while (true)

        {

            if (**GetAsyncKeyState**(**VK\_LEFT**))

            {

**moveTomLeft**();

            }

            if (**GetAsyncKeyState**(**VK\_RIGHT**))

            {

**moveTomRight**();

            }

            if (**GetAsyncKeyState**(**VK\_UP**))

            {

**moveTomUp**();

            }

            if (**GetAsyncKeyState**(**VK\_DOWN**))

            {

**moveTomDown**();

            }

            if (**GetAsyncKeyState**(**VK\_SPACE**))

            {

**generateBullet**();

            }

**printFruits**();

**moveBullet**();

**printScore**();

            if (timer == 1)

            {

**moveCheetah**();

                timer = 0;

            }

            if (turTimer == 3)

            {

**moveTurtle**();

                turTimer = 0;

            }

            timer++;

            turTimer++;

**Sleep**(30);

            if (lives<=0)

            {

                break;

            }

        }

**system**("cls");

        cout **<<** "Your Total Score is " **<<** score **<<** **endl**;

    }

}

void **header**()

{

    cout **<<** " \_\_\_\_\_\_\_                    \_\_\_\_\_\_\_\_\_                                    \_\_\_\_\_\_\_\_     " **<<** **endl**;

    cout **<<** "|            |        |    |          \\    |         |        \_-\_       |        \\    " **<<** **endl**;

    cout **<<** "|            |  \\     |    |           |   |         |      /     \\     |         |   " **<<** **endl**;

    cout **<<** "|\_\_\_\_\_\_\_     |   \\    |    |           |   |         |     |       |    |\_\_\_\_\_\_\_\_/    " **<<** **endl**;

    cout **<<** "|            |    \\   |    |           |   |    /\\   |     |  ---  |    |\\           " **<<** **endl**;

    cout **<<** "|            |     \\  |    |           |   |   /  \\  |     |       |    |  \\         " **<<** **endl**;

    cout **<<** "|            |      \\ |    |          |    |  /    \\ |     |       |    |    \\       " **<<** **endl**;

    cout **<<** "|\_\_\_\_\_\_\_     |        |    |\_\_\_\_\_\_\_\_\_/     |         |     |       |    |      \\     " **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** **endl**;

}

int **menu**()

{

    int option;

    cout **<<** "Menu." **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" **<<** **endl**;

    cout **<<** "1. Start" **<<** **endl**;

    cout **<<** "2. Option" **<<** **endl**;

    cout **<<** "3. Exit" **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** "Enter one option:  ";

    cin **>>** option;

    return option;

}

int **option**()

{

**system**("cls");

    int option;

    cout **<<** "1. Keys" **<<** **endl**;

    cout **<<** "2. Instruction" **<<** **endl**;

    cout **<<** "3. Exit" **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** "Enter one option:  ";

    cin **>>** option;

    return option;

}

int **keys**()

{

    int option;

**system**("cls");

    cout **<<** "Keys" **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" **<<** **endl**;

    cout **<<** "1. UP              Go up" **<<** **endl**;

    cout **<<** "2. Down            Go down" **<<** **endl**;

    cout **<<** "1. Left            Go left" **<<** **endl**;

    cout **<<** "1. Right           Go right" **<<** **endl**;

    cout **<<** "1. Space           For fire" **<<** **endl**;

    cout **<<** "1. ESC             To end game" **<<** **endl**;

    cout **<<** "Press 0 to go back to submenu:  ";

    cin **>>** option;

}

int **instruction**()

{

    int option;

**system**("cls");

    cout **<<** "Instructions" **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" **<<** **endl**;

    cout **<<** "1. Do not collide with enemies." **<<** **endl**;

    cout **<<** "2. Pick treasure with the help of fire." **<<** **endl**;

    cout **<<** **endl**;

    cout **<<** "Press 0 to go back to submenu:  ";

    cin **>>** option;

    return option;

}

void **road**()

{

    cout **<<** " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** "|                                                                                                                 |" **<<** **endl**;

    cout **<<** " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ " **<<** **endl**;

}

void **printCheetah**()

{

**gotoxy**(enemyX, enemyY);

    for (int index = 0; index < 16; index++)

    {

        cout **<<** cheetah1[index];

    }

**gotoxy**(enemyX, enemyY + 1);

    for (int index = 0; index < 16; index++)

    {

        cout **<<** cheetah2[index];

    }

**gotoxy**(enemyX, enemyY + 2);

    for (int index = 0; index < 16; index++)

    {

        cout **<<** cheetah3[index];

    }

}

void **printTurtle**()

{

**gotoxy**(TurX, TurY);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** turtle1[index];

    }

**gotoxy**(TurX, TurY + 1);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** turtle2[index];

    }

**gotoxy**(TurX, TurY + 2);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** turtle3[index];

    }

**gotoxy**(TurX, TurY + 3);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** turtle4[index];

    }

}

void **printTom**()

{

**gotoxy**(tomX, tomY);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** tom1[index];

    }

**gotoxy**(tomX, tomY + 1);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** tom2[index];

    }

**gotoxy**(tomX, tomY + 2);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** tom3[index];

    }

**gotoxy**(tomX, tomY + 2);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** tom4[index];

    }

}

void **eraseCheetah**()

{

**gotoxy**(enemyX, enemyY);

    for (int index = 0; index < 16; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(enemyX, enemyY + 1);

    for (int index = 0; index < 16; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(enemyX, enemyY + 2);

    for (int index = 0; index < 16; index++)

    {

        cout **<<** " ";

    }

}

void **eraseTurtle**()

{

**gotoxy**(TurX, TurY);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(TurX, TurY + 1);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(TurX, TurY + 2);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(TurX, TurY + 3);

    for (int index = 0; index < 13; index++)

    {

        cout **<<** " ";

    }

}

void **moveCheetah**()

{

    if (enemyDirection **==** "Left")

    {

        char next = **getCharAtxy**(enemyX - 1, enemyY);

        if (next == ' ' || next == '@')

        {

**eraseCheetah**();

            enemyX--;

**printCheetah**();

        }

        if (next == '|')

        {

            enemyDirection **=** "Right";

        }

    }

    if (enemyDirection **==** "Right")

    {

        char next = **getCharAtxy**(enemyX + 16, enemyY);

        if (next == ' ' || next == '@')

        {

**eraseCheetah**();

            enemyX++;

**printCheetah**();

        }

        if (next == '|')

        {

            enemyDirection **=** "Left";

        }

    }

}

void **moveTurtle**()

{

    if (turtleDirection **==** "Left")

    {

        char next = **getCharAtxy**(TurX - 1, TurY);

        if (next == ' ' || next == '@')

        {

**eraseTurtle**();

            TurX--;

**printTurtle**();

        }

        if (next == '|')

        {

            turtleDirection **=** "Right";

        }

    }

    if (turtleDirection **==** "Right")

    {

        char next = **getCharAtxy**(TurX + 15, TurY);

        if (next == ' ' || next == '@')

        {

**eraseTurtle**();

            TurX++;

**printTurtle**();

        }

        if (next == '|')

        {

            turtleDirection **=** "Left";

        }

    }

}

void **moveTomLeft**()

{

    char next = **getCharAtxy**(tomX - 1, tomY);

    if (next == ' ')

    {

**eraseTom**();

        tomX = tomX - 1;

**printTom**();

    }

    else{

**eraseTom**();

**printTom**();

        if (next != '|')

        {

**eraseTom**();

            lives--;

            tomY = 18;

        }

    }

}

void **moveTomRight**()

{

    char next = **getCharAtxy**(tomX + 17, tomY);

    if (next == ' ')

    {

**eraseTom**();

        tomX = tomX + 1;

**printTom**();

    }

    else{

**eraseTom**();

**printTom**();

        if (next != '|')

        {

**eraseTom**();

            lives--;

            tomY = 18;

        }

    }

}

void **moveTomUp**()

{

    char next = **getCharAtxy**(tomX, tomY - 1);

    if (next == ' ')

    {

**eraseTom**();

        tomY = tomY - 1;

**printTom**();

    }

    else{

**eraseTom**();

**printTom**();

        if (next != '\_')

        {

**eraseTom**();

            lives--;

            tomY = 18;

        }

    }

}

void **moveTomDown**()

{

    char next = **getCharAtxy**(tomX+9, tomY + 4);

    if (next == ' ')

    {

**eraseTom**();

        tomY = tomY + 1;

**printTom**();

    }

    else{

**eraseTom**();

**printTom**();

        if (next != '\_')

        {

**eraseTom**();

            lives--;

            tomY = 18;

        }

    }

}

void **eraseTom**()

{

**gotoxy**(tomX, tomY);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(tomX, tomY + 1);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(tomX, tomY + 2);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** " ";

    }

**gotoxy**(tomX, tomY + 3);

    for (int index = 0; index < 15; index++)

    {

        cout **<<** " ";

    }

}

void **gotoxy**(int x, int y)

{

**COORD** coordinates;

    coordinates.X = x;

    coordinates.Y = y;

**SetConsoleCursorPosition**(**GetStdHandle**(**STD\_OUTPUT\_HANDLE**), coordinates);

}

char **getCharAtxy**(short int x, short int y)

{

**CHAR\_INFO** ci;

**COORD** xy = {0, 0};

**SMALL\_RECT** rect = {x, y, x, y};

**COORD** coordBufSize;

    coordBufSize.X = 1;

    coordBufSize.Y = 1;

    return **ReadConsoleOutput**(**GetStdHandle**(**STD\_OUTPUT\_HANDLE**), &ci, coordBufSize, xy, &rect) ? ci.Char.AsciiChar : ' ';

}

void **generateBullet**()

{

    bulletX[bulletCount] = tomX + 7;

    bulletY[bulletCount] = tomY;

**gotoxy**(tomX + 7, tomY);

    cout **<<** ".";

    bulletCount++;

}

void **moveBullet**()

{

    for (int x = 0; x < bulletCount; x++)

    {

        char next = **getCharAtxy**(bulletX[x], bulletY[x] - 1);

        if (next != ' ')

        {

**eraseBullet**(bulletX[x], bulletY[x]);

**removeBulletFromArray**(x);

            if (next == '@')

            {

                score++;

**removeFruits**();

**generateFruits**();

                if (score%10 == 0)

                {

**srand**(**time**(0));

**eraseCheetah**();

**eraseTurtle**();

                    enemyY = 5 + **rand**()%10;

                    TurY = enemyY - (4 + **rand**()%5);

                }

            }

        }

        else

        {

**eraseBullet**(bulletX[x], bulletY[x]);

            bulletY[x] = bulletY[x] - 1;

**printBullet**(bulletX[x], bulletY[x]);

        }

    }

}

void **removeBulletFromArray**(int index)

{

    for (int x = index; x < bulletCount - 1; x++)

    {

        bulletX[x] = bulletX[x + 1];

        bulletY[x] = bulletY[x + 1];

    }

    bulletCount--;

}

void **printBullet**(int x, int y)

{

**gotoxy**(x, y);

    cout **<<** ".";

}

void **eraseBullet**(int x, int y)

{

**gotoxy**(x, y);

    cout **<<** " ";

}

void **generateFruits**(){

**srand**(**time**(0));

    for (int i=0; i<5; i++){

        fruitsX[i] = **rand**() % 100;

        fruitsY[i] = **rand**() % 15;

    }

}

void **printFruits**(){

    for (int i=0; i<5; i++){

**gotoxy**(fruitsX[i]+5,fruitsY[i]+1);

        cout **<<** "@";

    }

}

void **removeFruits**()

{

    for (int i=0; i<5; i++){

**gotoxy**(fruitsX[i]+5,fruitsY[i]+1);

        cout **<<** " ";

    }

}

void **printScore**(){

**gotoxy**(1,25);

    cout **<<** " Score: "**<<** score **<<** "\t Lives: "**<<** lives **<<endl**;

}