Project 2

Zombie Attack

Name: _____ Time spent: _____min

Deliverables

via EMAIL

3:00 pm W15 – C2 – no exceptions

2 separate attached files (no code in the email body)

Flow Diagram

Code – in xyz.c file

This project involves creating a game where the user (the zombie) attempts to capture a human (the human) who is wandering randomly around a fenced-in field (the field).

The field is an NxM array with the border elements (fences) marked with:

　　　　{ or } on the left and right side respectively

　　　　– on the top and bottom

　　　　z indicating the zombie position

　　　　h indicating the human position

　　　　N = 25, M = 60, set with #define

The zombie is an array or structure:

　　　　holding its x-y position

　　　　moved by the h(left), j(up), k(down), l(right) keys

The human is an array or structure:

　　　　holding its x-y position

　　　　moved at random in the left/up/down/right direction

The game starts with a splash screen indicating how the game is played

At the start:

       The human is placed in the field at random

       The zombie is placed in the field at random

Each loop of the game:

       The keyboard is checked for a key press

           if detected – the location of the zombie is updated

       A check for a win is performed

           both human and zombie in the same location?

       The location of the human is updated

           A short time delay is implemented (50ms – 200ms)

If a win is detected:

       The human/zombie characters are replaced with an X

       A winner message is placed near the middle of the field

       A request to replay or quit is made

When the zombie or human locations are updated:

       They must remain inside the fenced-in portion of the field

Window printing function

```c
#include <windows.h>
```

Use the following function to place the cursor in the desired location – simply add it to your program file

```c
void set_cursor_position(int row, int col){
    HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
    fflush(stdout);
    COORD coord = { (SHORT)col, (SHORT)row };
    SetConsoleCursorPosition(hOut, coord);
    return;
}// end  set_cursor_position
```

Use printf(…) to print a single character or a text string to the new location

Keyboard checking code

```
#include "conio.h"      // Note this is not portable but works for Windows
```

Use the following code to check if a keyboard key has been pressed, and if yes – grab the character

```
// check for keyboard hit
if(_kbhit())
    input_char = _getch();
```

_kbhit   keyboard hit
    Returns false if nothing is in the keyboard buffer (no key hit)
    Returns true if there is something in the keyboard buffer (a key has been pressed)
    Remains true until the character is removed from the keyboard buffer
_getch   get character
    Returns the current (first) character from the keyboard buffer

Random Number code

```
#include <time.h>
```

Use the srand() function to set the random number generator seed (starts at a different value every time) – do this one time in the program

```
// initialize random seed so game will be different every time
srand(time(NULL));
```

Use the rand() function to generate a random number

```
// Generates a random number between 0 and the max integer represented by the computer
// To limit the range of the numbers – modulo the value by the desired (range + 1)
// e.g. to get values from 0 to 5
foo = rand() % 6;
```

Running the program

This game must be played in a Windows terminal instead of the Eclipse console window:

After compilation – in Windows go to the project_directory/debug folder and double click on the file: my_project_name.exe

A Windows terminal will open

Place your cursor in the window to allow keyboard input

To stop the cursor from flashing

rt-click in the window and select properties

select the Terminal tab

set the cursor color to match the background color

project_directory and my_project_name are what you used for your project

Running the program

Windows 11 has an additional requirement for the program to run

Once the window is open, select Settings → Startup and set the number of rows to 80