Name

Advanced Programming in C++

Lab Exercise 3/12/2020 Stardate: 73193.99

Frog Hop

Today we will use a 2D array to simulate a frog in a marsh. Your frog will start at some initial position in the middle of the marsh and hop from pad to pad. In this exercise you will use “old school” C programming. Once your program is working, rewrite the program in C++ using the vector class or a 2D array. Attach your source code (C++ version) to this document and turn in. Also turn in your Frog.cpp and Frog.h class definition and class implementation files.

Program specifications:

1. Create a 40 x 40 array that holds characters which is initialized to the character ‘\_’
2. Put the frog in the location row 20 column 20 and mark that space with a ‘F’
3. Generate a random number from 1 to 5 that creates the following result.

1 = move one pad to the north

2 = move one pad to the south

3 = move one pad to the east

4 = move one pad to the west

5 = don’t move at all

1. Ensure the frog does not leave the marsh.
2. Each step in the simulation you should do the following:
3. generate random number
4. determine move
5. move frog to new location (be sure to erase frog from old location)
6. display marsh
7. You program should allow the user to specify the number of steps in the simulation.
8. Run the simulation for a variety of steps and fill out the following table

|  |  |
| --- | --- |
| **# of steps** | **Distance from origin** |
| 10 |  |
| 100 |  |
| 1000 |  |
| 10000 |  |
| 100000 |  |
| 1000000 |  |

1. What can you surmise from the following data?
2. What specifically prevents the frog from leaving the swamp?
3. Now create a Frog Class that implements all of the functions in this program. Print your class definition and class implementation files and attach to this sheet.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* FROGHOP.C By: Mr. Messa 12/4/2001 \*\*

\*\* This program simulate the random movement of a frog in a pond \*\*

\*\* this technique may be used to simulate an ecosystem \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <conio.h>

#include <time.h>

#define ROWS 40

#define COLS 40

// Function prototypes

void initialize(char [][40]);

void jump(char [][40], int &, int &);

void display(char [][40]);

float distance(int, int);

int main()

{

char marsh[ROWS][COLS];

int steps, frow = 20, fcol = 20, index;

float dist, maxDistance=0.0;

printf("How many steps in your simulation?\n");

scanf("%d", &steps);

srand(time(NULL));

initialize(marsh);

marsh[frow][fcol]='F';

for (index = 1; index <= steps; index++)

{

//system("CLS");

jump(marsh, frow, fcol);

//display(marsh);

dist = distance(frow, fcol);

if (dist > maxDistance)

maxDistance = dist;

}

marsh[20][20] = 'S';

system("CLS");

display(marsh);

printf("You wandered %f away from home\n\n" , maxDistance);

return 0;

}

// This function initializes the array

void initialize(char m[][COLS])

{

int i, j;

for(i = 0; i < ROWS; i++)

for (j = 0; j < COLS; j++)

m[i][j] = '\_';

}

// This function moves the frog 1 unit in a random direction

// The function also marks the location where the frog has been

void jump(char m[][COLS], int &frow, int &fcol)

{

int rnumber;

rnumber = rand() % 5 + 1;

switch (rnumber)

{

case 1: break;

case 2: if (frow == ROWS-1)

break;

m[frow][fcol] = '\*';

frow++;

m[frow][fcol] = 'F';

break;

case 3: if (frow == 0)

break;

m[frow][fcol] = '\*';

frow--;

m[frow][fcol] = 'F';

break;

case 4: if (fcol == COLS-1)

break;

m[frow][fcol] = '\*';

fcol++;

m[frow][fcol] = 'F';

break;

case 5: if (fcol == 0)

break;

m[frow][fcol] = '\*';

fcol--;

m[frow][fcol] = 'F';

break;

}

}

// This function displays the pond

void display(char m[][40])

{

int i, j;

for(i=0; i < ROWS; i++)

{

for (j = 0; j < COLS; j++)

printf("%c", m[i][j]);

printf("\n");

}

}

// This function calculates the distance the frog is from the starting point

float distance(int frow, int fcol)

{

return sqrt(pow(abs(frow-20), 2.0)+ pow(abs(fcol-20), 2.0));

}