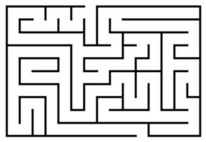
Project 2
mrBlob

Exercise 21 pg. 581

jasonMaynard

summer_13



Program design

Class: Grid

public methods:

+ Grid()
+ Grid(ifsteam & f)
+ unsigned long getNumRows()
+ unsigned long getNumCols()
+ void displayGrid()

private data members:

- vector<vector<char>> _aGrid
- int _c

Main()

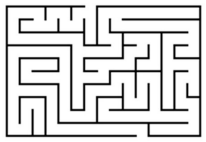
Functions:

```
void setFilled(Grid& g, vector<vector<bool> >& filled)
void clearFilled(Grid& g, vector<vector<bool> >& filled)
void clearVisited(Grid& g, vector<vector<bool> >& visited)
int sizeofBlob(int r, int c, const Grid& g,
        vector<vector<bool> >& filled,
        vector<vector<bool> >& visited)
```

Key data structures:

```
vector<vector<bool> > filled(rows,
                    vector<bool>(columns, false));
vector<vector<bool> > visited(rows,
                    vector<bool>(columns, false));
int blobCount
```

Input file is read and sent to the second grid constructor to create the grid



Program flow

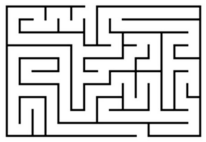
- Main drives program
  - Sets up grid objects by reading the text file
  - Initializes bool “arrays” for “full” and “visited” cells
    - Program actually uses C++ vectors to dynamically allocate size
  - Performs recursive function to count the size of a blob
  - If the size of a blob is > 0, the blob counter is incremented
  - Prints grid and number of blobs as result

- All done in three simple steps:

  STEP 1 - READ GRID

  STEP 2 - FIND and COUNT BLOBS

  STEP 3 - DISPLAY RESULTS

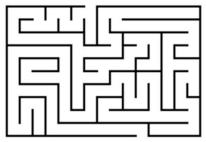


Reading the input file

```
// Creates grid object by passing a reference to input file stream
Grid::Grid(ifstream& f)
{
  // Read the character from the file stream
  _c = f.get();

  // If the char is not the end of file then...
  while (_c != EOF)
  {
    vector<char> row; // Create an temp row to fill with chars

    // If the char is not a new line then..
    while (_c != '\n')
    {
      //
      // fill the first row
      //
      if (_c == '1')
      {
        //change the '1's from the input file to '*' for printing
        _c = '*';
        //push the value to the end of the row
        row.push_back(_c);
        _c = f.get();
      }
      else if (_c == '0')
      {
        //change the '0's from the input file to ' ' for printing
        _c = ' ';
        //push the value to the end of the row
        row.push_back(_c);
        _c = f.get();
      }
```

```
else if (_c == ' ')
      {
        // do nothing
        _c = f.get();
      }
      else if (_c != '0' && _c != '1' && _c != ' ' && _c != -1)
      {
        // If chars aren't '1', '0', ' ', or '\n' break out
        cout << "\n*** ERROR BUILDING GRID - BAD CHAR ***" << endl;
        break;
      }
      else
      {
        break;
      }

    }//end while (c!='\n')

    //
    // We now have a row filled with '*'s and ' 's. Push the row on the
    // grid and get the next char to start building the next row.
    //
    _aGrid.push_back(row);
    _c = f.get();

  }//end while (!eof)

  // Potential enhancement - Perform check to make esure all columns are
  // same size, if not return error. For now we assume a well formed grid
  // as an input per Catrain's email 7/6/13
```

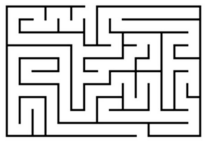
The input file is validated in Main() and a reference to that stream is sent to this constructor. Further error checking is done to look for bad characters. This assumes a rectangular grid where column size = the size of the first row. A "jaggy" array will be edited based on the size of that first row.



Find and count blobs

"sizeofBlob()" Recursive function

```
int sizeofBlob(int r, int c, const Grid& g,
               vector<vector<bool> >& filled,
               vector<vector<bool> >& visited)
{
  //
  // Error checking. Don't walk of grid when looking for blobs
  //
  if (r < 0 || r >= g.getNumRows() || c < 0 || c >= g.getNumCols())
  {
    // This position is not in the grid, so there is
    // no blob at this position.  Return a blob size of zero.
    //
    return 0;
  }//end if "don't walk off grid"

  //
  // Look for cells that might have a blob.
  //
  if (filled[r][c] == false || visited[r][c] == true) {
    // This square is not part of a blob, or else it has
    // already been counted, so return zero.
    return 0;
  }//end if "not a new blob"

  //
  // Mark the square as visited so that we won't count it again during the
  // following recursive calls.
  //
  visited[r][c] = true;

  // Count the square at this position, then count the
  // the cells that are connected to this square
  // horizontally or vertically.
  //
  int size = 1; //local size var

  size += sizeofBlob(r-1, c, g, filled, visited);
  size += sizeofBlob(r+1, c, g, filled, visited);
  size += sizeofBlob(r, c-1, g, filled, visited);
  size += sizeofBlob(r, c+1, g, filled, visited);

  return size;

}
```



Screen snap showing output in x code IDE

Tools Bookmarks Window Help — Sat Jul 6 3:20 PM

blobCount.xcodeproj — Grid.cpp

Finished running blobCount : blobCount

No Issues

Breakpoints Editor View Organizer

blobCount › Grid.cpp › Grid::Grid(ifstream& f)

```
// Creates grid object by passing a reference to input file stream
Grid::Grid(ifstream& f)
{
   // Read the character from the file stream
   _c = f.get();

   // If the char is not the end of file then...
   while (_c != EOF)
   {
      vector<char> row; // Create an temp row to fill with chars

      // If the char is not a new line then..
      while (_c != '\n')
      {
         //
         // fill the first row
         //
         if (_c == '1')
         {
            //change the '1's from the input file to '*' for printing
            _c = '*';
            //push the value to the end of the row
            row.push_back(_c);
            _c = f.get();
         }
         else if (_c == '0')
         {
            //change the '0's from the input file to ' ' for printing
            _c = ' ';
            //push the value to the end of the row
            row.push_back(_c);
            _c = f.get();
         }
         else if (_c == ' ')
         {
```

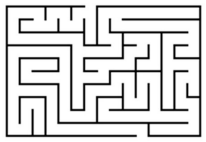
grid.txt

```
0 0 0 1
1 0 0 1
0 1 0 0
```

No Selection

Auto — Clear

```
The grid looks like this:
   *
*  *
 *

The number of blobs in Grid 1 is: 3
```

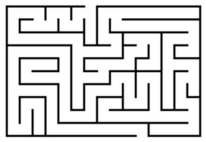


Running in UNIX on USF Circe

blobCount — jlmaynard@login2:~/ds_project2 — ssh

```
[jlmaynard@login2 ds_project2]$ whoami
jlmaynard
[jlmaynard@login2 ds_project2]$ ls -l
total 24
-rw-r--r-- 1 jlmaynard jlmaynard 3180 Jul  6 15:47 Grid.cpp
-rw-r--r-- 1 jlmaynard jlmaynard 1707 Jul  6 15:47 Grid.h
-rw-r--r-- 1 jlmaynard jlmaynard   23 Jul  6 15:49 grid.txt
-rw-r--r-- 1 jlmaynard jlmaynard 5252 Jul  6 15:48 main.cpp
-rw-r--r-- 1 jlmaynard jlmaynard   48 Jul  6 15:48 makefile
[jlmaynard@login2 ds_project2]$ make
g++ main.cpp Grid.h Grid.cpp -o blobCount
[jlmaynard@login2 ds_project2]$ ls -l
total 84
-rwxrwx--- 1 jlmaynard jlmaynard 60189 Jul  6 15:49 blobCount
-rw-r--r-- 1 jlmaynard jlmaynard  3180 Jul  6 15:47 Grid.cpp
-rw-r--r-- 1 jlmaynard jlmaynard  1707 Jul  6 15:47 Grid.h
-rw-r--r-- 1 jlmaynard jlmaynard    23 Jul  6 15:49 grid.txt
-rw-r--r-- 1 jlmaynard jlmaynard  5252 Jul  6 15:48 main.cpp
-rw-r--r-- 1 jlmaynard jlmaynard    48 Jul  6 15:48 makefile
[jlmaynard@login2 ds_project2]$ cat grid.txt
0 0 0 1
1 0 0 1
0 1 0 0[jlmaynard@login2 ds_project2]$ ./blobCount

The grid looks like this:
   *
*  *
 *

The number of blobs in Grid 1 is: 3
[jlmaynard@login2 ds_project2]$
```

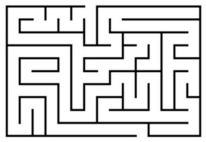
Here we see that we are logged on to USF running in UNIX. Makefile produces "blobCount" as required and reads properly formatted input file. Output produces the correct number of blobs



Testing different grids…

blobCount — jlmaynard@login2:~/d

```
[jlmaynard@login2 ds_project2]$ ls
blobCount  Grid.cpp  Grid.h  grid.txt  main.cpp  makefile
[jlmaynard@login2 ds_project2]$ nano grid.txt
[jlmaynard@login2 ds_project2]$ cat grid.txt
0 0 0 0 0 0
1 1 1 1 1 1
0 0 0 0 0 0
[jlmaynard@login2 ds_project2]$ ./blobCount

The grid looks like this:

******


The number of blobs in Grid 1 is: 1
[jlmaynard@login2 ds_project2]$ vim grid.txt
[jlmaynard@login2 ds_project2]$ cat grid.txt
0 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 0
[jlmaynard@login2 ds_project2]$ ./blobCount

The grid looks like this:

 *
  *
   *


The number of blobs in Grid 1 is: 3
[jlmaynard@login2 ds_project2]$ ls
blobCount  Grid.cpp  Grid.h  grid.txt  main.cpp  makefile
[jlmaynard@login2 ds_project2]$ whoami
jlmaynard
[jlmaynard@login2 ds_project2]$ pwd
/home/j/jlmaynard/ds_project2
[jlmaynard@login2 ds_project2]$
```



Error checking examples

Bad characters in grid

```
blobCount — jlmaynard@login2:~/
[jlmaynard@login2 ds_project2]$ nano grid.txt
[jlmaynard@login2 ds_project2]$ cat grid.txt
Abc123
[jlmaynard@login2 ds_project2]$ ./blobCount

*** ERROR BUILDING GRID - BAD CHAR ***

*** ERROR BUILDING GRID - BAD CHAR ***

*** ERROR BUILDING GRID - BAD CHAR ***

*** ERROR BUILDING GRID - BAD CHAR ***

*** ERROR BUILDING GRID - BAD CHAR ***

The grid looks like this:

The number of blobs in Grid 1 is: 0
[jlmaynard@login2 ds_project2]$
```

Bad input file

```
blobCount — jlmaynard@login2:~/ds_project2 — ssh — 113
[jlmaynard@login2 ds_project2]$ ls -l
total 84
-rwxrwx--- 1 jlmaynard jlmaynard 60189 Jul  6 15:49 blobCount
-rw-r--r-- 1 jlmaynard jlmaynard  3180 Jul  6 15:47 Grid.cpp
-rw-r--r-- 1 jlmaynard jlmaynard  1707 Jul  6 15:47 Grid.h
-rw-r--r-- 1 jlmaynard jlmaynard     7 Jul  6 16:21 grid.txt
-rw-r--r-- 1 jlmaynard jlmaynard  5252 Jul  6 15:48 main.cpp
-rw-r--r-- 1 jlmaynard jlmaynard    48 Jul  6 15:48 makefile
[jlmaynard@login2 ds_project2]$ mv grid.txt bogus.txt
[jlmaynard@login2 ds_project2]$ ls -l
total 84
-rwxrwx--- 1 jlmaynard jlmaynard 60189 Jul  6 15:49 blobCount
-rw-r--r-- 1 jlmaynard jlmaynard     7 Jul  6 16:21 bogus.txt
-rw-r--r-- 1 jlmaynard jlmaynard  3180 Jul  6 15:47 Grid.cpp
-rw-r--r-- 1 jlmaynard jlmaynard  1707 Jul  6 15:47 Grid.h
-rw-r--r-- 1 jlmaynard jlmaynard  5252 Jul  6 15:48 main.cpp
-rw-r--r-- 1 jlmaynard jlmaynard    48 Jul  6 15:48 makefile
[jlmaynard@login2 ds_project2]$ ./blobCount
blobCount: main.cpp:43: int main(): Assertion `inputFile.good()' failed.
Aborted
[jlmaynard@login2 ds_project2]$
```