```
Script started on 2024-07-15 21:06:19-05:00 [TERM="xterm-256color" TTY="/dev/pts/0"
dg46622@ares:~/sheep$ pwd
/home/students/dg46622/sheep
dg46622@ares:~/sheep$ cat sheep.info

Gerald Dichoso
CSC122-001

Project: Where, oh where, has my little SHEEP gone?! - Level:   6

                                              Total Levels:   6

Description:

The following program creates a word search from a file that contains
a list of words. The program contains a class that makes a grid of
letters and manages them as well. When the user enters an invalid
file, it will prompt the user to enter another file name. Once the
user enters a valid file name they now can play the game. The user is
asked to enter a word they're searching for, a starting position (x &
y-coordinates), and a cardinal direction (N, SE, W, etc). If a word
from the list is found, the program will capitalize it in the word
search. Once there's no more words on the list, then the game ends.

dg46622@ares:~/sheep$ cat words.txt
blueberry
banana
apple
cabbage
carrot
strawberry
pear
onion
watermelon
bread
toast
butter
spinach
celery
grape
mango
pineapple
carbon
oxygen
radon
lithium
neon
boron
helium
bromine
argon
tin
iodine
sodium
calcium
potassium
sulfur
nitrogen
chlorine
dg46622@ares:~/sheep$ show-code search.h search.cpp sheep.cpp


search.h:
```

```
 1  #pragma once
 2  #include <memory>
 3  #include <vector>
 4  #include <fstream>
 5  #include <iomanip>
 6  #include <iostream>
 7  #include <random>
 8  #include <algorithm>
 9
10  class lettergrid
11  {
12      size_t row{0};
13      size_t column{0};
14      std::vector<bool> word_found;
15      std::unique_ptr<char[]> letter;
16      std::unique_ptr<char*[]> grid;
17
18  public:
19
20      const std::vector<bool>& getWordFound() const
21      {
22          return word_found;
23      }
24
25      lettergrid(size_t max_length);
26
27
28      lettergrid(const lettergrid& a);
29
30
31      lettergrid& operator=(const lettergrid& a) = delete;
32
33
34      void chooseorien(std::vector<std::string>& wordlist,
35      std::mt19937 randomnum);
36
37
38      bool placeword(short pitx, short pity, std::size_t positx,
39      std::size_t posity, std::string word);
40
41
42      void disp();
43
```

```
    void insertspace(std::mt19937& randomnum);


    bool checkword(std::string guess, std::size_t positx,
    std::size_t posity, std::string direct,
    std::vector<std::string>& wordlist);


    void choosedir(short dir, short& dx, short& dy);


    ~lettergrid();
};

std::string randomword(std::istream& file, std::size_t maxline,
std::mt19937& randomnum);


void deletevec(std::vector<std::string>& vec, std::size_t i);
```

search.cpp:

```
#include "search.h"
#include <algorithm>
#include <map>

lettergrid::lettergrid(size_t max_length)
    : row(max_length + 2), column(max_length + 2), word_found(),
    letter(new char[row * column]), grid(new char*[row])
{
    for (size_t i = 0; i < row; i++)
    {
        grid[i] = letter.get() + i * column;
    }

    for (size_t a = 0; a < row; a++)
    {
        for (size_t b = 0; b < column; b++)
        {
            grid[a][b] = '*';
        }
    }
}


lettergrid::lettergrid(const lettergrid& a)
    : row(a.row), column(a.column), word_found(a.word_found),
    letter(new char[a.row * a.column]), grid(new char*[a.row])
{
    for (size_t i = 0; i < row; i++)
    {
        grid[i] = letter.get() + i * column;
        std::copy(a.grid[i], a.grid[i] + column, grid[i]);
    }
}


void lettergrid::chooseorien(std::vector<std::string>& wordlist,
std::mt19937 randomnum)
{
    const short triesmax = 100;
    short tries;
    short orien;
    short dx = 0;
    short dy = 0;
    std::size_t x = 0;
    std::size_t y = 0;
    std::size_t listend = wordlist.size();
    std::uniform_int_distribution<short> rangeorien(1, 8);
    std::uniform_int_distribution<std::size_t> rangeplace(0, row);

    word_found.assign(listend, false);

    for (std::size_t a = 0; a < listend; a++)
    {
        tries = 0;

        do
        {
            orien = rangeorien(randomnum);
            choosedir(orien, dx, dy);
            x = rangeplace(randomnum);
            y = rangeplace(randomnum);
            tries++;
        }

        while (!placeword(dx, dy, x, y, wordlist[a]) && tries != triesmax);

        if (tries == triesmax)
        {
            deletevec(wordlist, a);
            listend--;
        }
    }
}


bool lettergrid::placeword(short pitx, short pity, std::size_t positx,
std::size_t posity, std::string word)
{
    std::size_t wordlength = word.length();
    short wordendx = static_cast<short>(positx + pitx * wordlength);
    short wordendy = static_cast<short>(posity + pity * wordlength);

    if (wordendy > static_cast<short>(row - 1) ||
```

```
        wordendx > static_cast<short>(column - 1) ||
        wordendy < 0 ||
        wordendx < 0 || posity > row - 1 ||
        positx > row - 1)
    {
        return false;
    }

    else
    {
        for (std::size_t i = 0; i < wordlength; i++)
        {
            if (grid[pity * i + posity][pitx * i + positx] !=
                word[i] &&
                grid[pity * i + posity][pitx * i + positx] != '*')
            {
                return false;
            }
        }

        for (std::size_t i = 0; i < wordlength; i++)
        {
            grid[pity * i + posity][pitx * i + positx] =
                word[i];
        }
        return true;
    }
}


void lettergrid::insertspace(std::mt19937& randomnum)
{
    std::uniform_int_distribution<int> distribution('a', 'z');

    for (std::size_t a = 0; a < row; a++)
    {
        for (std::size_t b = 0; b < row; b++)
        {
            if (grid[a][b] == '*')
            {
                grid[a][b] =
                    static_cast<char>(distribution(randomnum));
            }
        }
    }
}


std::string randomword(std::istream& file, std::size_t maxline,
std::mt19937& randomnum)
{
    std::string word;
    std::uniform_int_distribution<long> distribution(0,
        maxline);
    std::size_t random = distribution(randomnum);
    file.seekg(0);

    for (std::size_t i = 0; i < random; i++)
    {
        file.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
    }

    std::getline(file, word);
    std::size_t wordlength = word.length();

    for (std::size_t a = 0; a < wordlength; a++)
    {
        word[a] = static_cast<char>(std::tolower(word[a]));
    }
    return word;
}


void lettergrid::choosedir(short dir, short& dx, short& dy)
{
    static const std::map<short, std::pair<short, short>> dirMap =
    {
        {1, {1, 0}}, {2, {-1, 0}}, {3, {0, -1}}, {4, {0, 1}},
        {5, {-1, -1}}, {6, {1, -1}}, {7, {-1, 1}}, {8, {1, 1}}
    };

    auto it = dirMap.find(dir);

    if (it != dirMap.end())
    {
        dx = it->second.first;
        dy = it->second.second;
    }
}


bool lettergrid::checkword(std::string guess, std::size_t positx,
std::size_t posity, std::string direct, std::vector<std::string>& wordlist
{
    short dx = 0, dy = 0;
    short dir = 0;
    std::size_t listend = wordlist.size();
    std::size_t guesslength = guess.length();
    bool inlist = false;
    std::size_t b;

    for (std::size_t a = 0; a < guesslength; a++)
    {
        guess[a] = static_cast<char>(std::tolower(guess[a]));
    }

    for (b = 0; b < listend; b++)
    {
```

```
        if (guess == wordlist[b])
        {
            inlist = true;
            break;
        }
    }

    if (inlist)
    {
        if (direct[0] == 'N')
        {
            if (direct == "NW")
            {
                dir = 5;
            }

            else if (direct == "NE")
            {
                dir = 6;
            }

            else
            {
                dir = 3;
            }

        }

        else if (direct[0] == 'S')
        {
            if (direct == "SW")
            {
                dir = 7;
            }

            else if (direct == "SE")
            {
                dir = 8;
            }

            else
            {
                dir = 4;
            }
        }

        else if (direct == "E")
        {
            dir = 1;
        }

        else if (direct == "W")
        {
            dir = 2;
        }

        choosedir(dir, dx, dy);

        if ((posity + dy * guesslength) > row - 1 ||
            (positx + dx * guesslength) > column - 1 ||
            static_cast<short>(positx + dx * guesslength) < 0 ||
            static_cast<short>(posity + dy * guesslength) < 0)
        {
            return false;
        }

        else
        {
            for (std::size_t a = 0; a < guesslength; a++)
            {
                if (std::toupper(grid[positx + dx * a][posity + dy * a])
                    != std::toupper(guess[a]))
                {
                    return false;
                }
            }

            for (std::size_t a = 0; a < guesslength; a++)
            {
                grid[positx + dx * a][posity + dy * a]
                    = static_cast<char>(std::toupper(guess[a]));
            }

            word_found[b] = true;
            return true;
        }
    }

    else
    {
        return false;
    }
}


void lettergrid::disp()
{
    std::cout << "   X ";

    for (std::size_t a = 0; a < row; a++)
    {
        std::cout << std::setw(4) << a + 1;
    }

    std::cout << "\n Y\n";

    for (std::size_t a = 0; a < column; a++)
    {
```

```
        std::cout << std::setw(2) << a + 1 << std::setw(4);

        for (std::size_t b = 0; b < row; b++)
        {
            std::cout << grid[b][a] << "   ";
        }

        std::cout << "\n\n";
    }
}

lettergrid::~lettergrid() {}

void deletevec(std::vector<std::string>& vec, std::size_t i)
{
    vec[i] = vec.back();
    vec.pop_back();
}
```

sheep.cpp:

```
#include "search.h"
#include <iostream>
#include <fstream>
#include <string>
#include <random>
#include <limits>

using namespace std;

void openfile(ifstream& a, string name);
size_t maxline(ifstream& a);
size_t longword(vector<string> a);


int main()
{
    size_t wordnum;
    size_t wordmaxlen;
    size_t maxlines;
    vector<string> words;
    string filename;
    ifstream file;
    mt19937 randomnum;
    randomnum.seed(random_device()());

    cout << "\nWelcome To Word Search!!";
    cout <<"\n\nHow many words would you like?\n";
    cin >> wordnum;

    openfile(file, filename);
    maxlines = maxline(file);
    file.clear();

    for (size_t i = 0; i < wordnum; i++)
    {
        words.push_back(randomword(file, maxlines, randomnum));
    }

    wordmaxlen = longword(words);
    lettergrid wordgrid(wordmaxlen);
    wordgrid.chooseorien(words, randomnum);

    for (size_t a = 0; a < words.size(); a++)
    {
        cout << words[a] << endl;
    }

    wordgrid.insertspace(randomnum);
    wordgrid.disp();

    string guess;
    string direct;
    size_t xposition;
    size_t yposition;

    while (any_of(wordgrid.getWordFound().begin(),
    wordgrid.getWordFound().end(), [](bool found) { return !found; }))
    {
        cout << "\n\nEnter a Word, X-Coordinate, Y-Coordinate, ";
        cout << "and Direction.\nDirection In form N, NW, SW, etc.";
        cout << "\nEXAMPLE: apple 1 7 N\n";

        cin >> guess >> xposition >> yposition >> direct;
        cout << "\n\n";
    wordgrid.checkword(guess, xposition - 1, yposition - 1, direct, words);

        for (size_t a = 0; a < words.size(); a++) {
            cout << words[a] << endl;
        }
        wordgrid.disp();
    }
    cout << "\n\nGAME OVER!!! You Found All the Words!!\n\n";
    file.close();
    file.clear();
}


void openfile(ifstream& a, string name)
{
    bool okay = false;

    while (!okay)
    {
        cout << "\nEnter File Name To Get Words From:\n";
        cin >> name;
```

```
        cout << "\n\n";
        a.open(name);

        if (!a)
        {
            cout << "ERROR OPENING FILE!! ENTER AGAIN!\n\n";
        }
        else
        {
            okay = true;
        }
    }
}


size_t maxline(ifstream& a)
{
    size_t max_line = 0;
    a.seekg(0);
    a.peek();

    while (!a.eof())
    {
        a.ignore(numeric_limits<streamsize>::max(), '\n');
        max_line++;
        a.peek();
    }
    return max_line;
}


size_t longword(vector<string> a)
{
    size_t max_length = 0;

    for (size_t i = 0; i < a.size(); i++)
    {
        if (a[i].length() > max_length)
        {
            max_length = a[i].length();
        }
    }
    return max_length;
}


string randomword(ifstream& file, size_t maxline, mt19937& randomnum)
{
    size_t line;
    string word;
    line = uniform_int_distribution<size_t>(0, maxline - 1)(randomnum);
    file.seekg(0);

    for (size_t i = 0; i < line; i++)
    {
        getline(file, word);
    }
    return word;
}
dg46622@ares:~/sheep$ CPP search sheep
search.cpp...
sheep.cpp***


dg46622@ares:~/sheep$ ./sheep.out

Welcome To Word Search!!

How many words would you like?
3

Enter File Name To Get Words From:
nowords.txt


ERROR OPENING FILE!! ENTER AGAIN!


Enter File Name To Get Words From:
haha.txt


ERROR OPENING FILE!! ENTER AGAIN!


Enter File Name To Get Words From:
nope.txt


ERROR OPENING FILE!! ENTER AGAIN!


Enter File Name To Get Words From:
words.txt


iodine
strawberry
toast
   X    1   2   3   4   5   6   7   8   9  10  11  12
 Y
 1   s   p   h   o   a   v   g   s   j   e   c   o

 2   n   q   z   t   s   a   o   t   m   y   s   h

 3   z   g   j   x   g   j   l   j   r   g   w   t

 4   k   e   w   v   z   l   u   r   w   w   e   t
```

```
 5   x   f   h   b   y   m   e   j   i   r   i   s

 6   k   h   c   s   a   b   n   o   v   b   m   o

 7   g   o   y   u   w   j   x   i   f   p   x   l

 8   m   a   f   a   t   t   y   g   t   b   i   z

 9   x   x   r   z   m   g   q   i   d   f   z   a

10   a   t   s   g   f   z   z   c   o   v   d   e

11   s   f   x   p   g   t   h   o   n   f   i   b

12   x   h   v   n   y   i   o   d   i   n   e   i


Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
iodine 6 12 E


iodine
strawberry
toast
   X    1   2   3   4   5   6   7   8   9  10  11  12
 Y
 1   s   p   h   o   a   v   g   s   j   e   c   o

 2   n   q   z   t   s   a   o   t   m   y   s   h

 3   z   g   j   x   g   j   l   j   r   g   w   t

 4   k   e   w   v   z   l   u   r   w   w   e   t

 5   x   f   h   b   y   m   e   j   i   r   i   s

 6   k   h   c   s   a   b   n   o   v   b   m   o

 7   g   o   y   u   w   j   x   i   f   p   x   l

 8   m   a   f   a   t   t   y   g   t   b   i   z

 9   x   x   r   z   m   g   q   i   d   f   z   a

10   a   t   s   g   f   z   z   c   o   v   d   e

11   s   f   x   p   g   t   h   o   n   f   i   b

12   x   h   v   n   y   I   O   D   I   N   E   i
```

```
Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
toast 8 2 W


iodine
strawberry
toast
   X    1   2   3   4   5   6   7   8   9  10  11  12
 Y
 1   s   p   h   o   a   v   g   s   j   e   c   o

 2   n   q   z   T   S   A   O   T   m   y   s   h

 3   z   g   j   x   g   j   l   j   r   g   w   t

 4   k   e   w   v   z   l   u   r   w   w   e   t

 5   x   f   h   b   y   m   e   j   i   r   i   s

 6   k   h   c   s   a   b   n   o   v   b   m   o

 7   g   o   y   u   w   j   x   i   f   p   x   l

 8   m   a   f   a   t   t   y   g   t   b   i   z

 9   x   x   r   z   m   g   q   i   d   f   z   a

10   a   t   s   g   f   z   z   c   o   v   d   e

11   s   f   x   p   g   t   h   o   n   f   i   b

12   x   h   v   n   y   I   O   D   I   N   E   i


Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
strawberry 1 11 NE


iodine
strawberry
toast
   X    1   2   3   4   5   6   7   8   9  10  11  12
 Y
 1   s   p   h   o   a   v   g   s   j   e   c   o

 2   n   q   z   T   S   A   O   T   m   Y   s   h
```

```
 3   z   g   j   x   g   j   l   j   R   g   w   t

 4   k   e   w   v   z   l   u   R   w   w   e   t

 5   x   f   h   b   y   m   E   j   i   r   i   s

 6   k   h   c   s   a   B   n   o   v   b   m   o

 7   g   o   y   u   W   j   x   i   f   p   x   l

 8   m   a   f   A   t   t   y   g   t   b   i   z

 9   x   x   R   z   m   g   q   i   d   f   z   a

10   a   T   s   g   f   z   z   c   o   v   d   e

11   S   f   x   p   g   t   h   o   n   f   i   b

12   x   h   v   n   y   I   O   D   I   N   E   i


GAME OVER!!! You Found All the Words!!

dg46622@ares:~/sheep$ ./sheep.out

Welcome To Word Search!!

How many words would you like?
1

Enter File Name To Get Words From:
fruit.txt


ERROR OPENING FILE!! ENTER AGAIN!


Enter File Name To Get Words From:
words.txt


spinach
   X    1   2   3   4   5   6   7   8   9
 Y
 1   o   x   m   a   i   p   f   s   l

 2   q   j   n   v   r   j   g   f   f

 3   s   p   i   n   a   c   h   i   h

 4   m   c   d   l   u   k   i   o   w

 5   x   x   q   a   g   c   g   k   b

 6   e   k   y   h   x   o   a   f   l

 7   h   a   t   x   s   i   r   y   k

 8   i   s   h   x   p   h   l   b   y

 9   o   c   i   s   n   m   z   n   x


Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
spinach 1 3 E


spinach
   X    1   2   3   4   5   6   7   8   9
 Y
 1   o   x   m   a   i   p   f   s   l

 2   q   j   n   v   r   j   g   f   f

 3   S   P   I   N   A   C   H   i   h

 4   m   c   d   l   u   k   i   o   w

 5   x   x   q   a   g   c   g   k   b

 6   e   k   y   h   x   o   a   f   l

 7   h   a   t   x   s   i   r   y   k

 8   i   s   h   x   p   h   l   b   y

 9   o   c   i   s   n   m   z   n   x


GAME OVER!!! You Found All the Words!!

dg46622@ares:~/sheep$ ./sheep.out

Welcome To Word Search!!

How many words would you like?
3

Enter File Name To Get Words From:
words.txt


blueberry
```

```
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
 Y
 1   s   e   r   w   n   k   c   p   b   a   i

 2   r   u   g   s   p   a   h   h   l   w   x

 3   y   l   z   k   x   a   l   e   u   w   q

 4   h   q   x   j   w   n   o   k   e   w   z

 5   w   l   x   x   v   g   r   l   b   i   h

 6   k   a   p   t   d   g   i   q   e   d   o

 7   e   n   o   q   b   a   n   b   r   i   o

 8   e   g   q   n   s   t   e   d   r   d   q

 9   d   n   l   k   g   z   k   k   y   g   r

10   o   n   i   o   n   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k



Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
blueberry 8 1 S


blueberry
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
 Y
 1   s   e   r   w   n   k   c   p   b   a   i

 2   r   u   g   s   p   a   h   h   l   w   x

 3   y   l   z   k   x   a   l   e   u   w   q

 4   h   q   x   j   w   n   o   k   e   w   z

 5   w   l   x   x   v   g   r   l   b   i   h

 6   k   a   p   t   d   g   i   q   e   d   o

 7   e   n   o   q   b   a   n   b   r   i   o

 8   e   g   q   n   s   t   e   d   r   d   q

 9   d   n   l   k   g   z   k   k   y   g   r

10   o   n   i   o   n   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k



Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
blueberry 9 1 S


blueberry
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
 Y
 1   s   e   r   w   n   k   c   p   B   a   i

 2   r   u   g   s   p   a   h   h   L   w   x

 3   y   l   z   k   x   a   l   e   U   w   q

 4   h   q   x   j   w   n   o   k   E   w   z

 5   w   l   x   x   v   g   r   l   B   i   h

 6   k   a   p   t   d   g   i   q   E   d   o

 7   e   n   o   q   b   a   n   b   R   i   o

 8   e   g   q   n   s   t   e   d   R   d   q

 9   d   n   l   k   g   z   k   k   Y   g   r

10   o   n   i   o   n   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k



Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
chlorine 6 1 S


blueberry
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
```

```
 Y
 1   s   e   r   w   n   k   c   p   B   a   i

 2   r   u   g   s   p   a   h   h   L   w   x

 3   y   l   z   k   x   a   l   e   U   w   q

 4   h   q   x   j   w   n   o   k   E   w   z

 5   w   l   x   x   v   g   r   l   B   i   h

 6   k   a   p   t   d   g   i   q   E   d   o

 7   e   n   o   q   b   a   n   b   R   i   o

 8   e   g   q   n   s   t   e   d   R   d   q

 9   d   n   l   k   g   z   k   k   Y   g   r

10   o   n   i   o   n   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k



Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
chlorine 7 1 S


blueberry
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
 Y
 1   s   e   r   w   n   k   C   p   B   a   i

 2   r   u   g   s   p   a   H   h   L   w   x

 3   y   l   z   k   x   a   L   e   U   w   q

 4   h   q   x   j   w   n   O   k   E   w   z

 5   w   l   x   x   v   g   R   l   B   i   h

 6   k   a   p   t   d   g   I   q   E   d   o

 7   e   n   o   q   b   a   N   b   R   i   o

 8   e   g   q   n   s   t   E   d   R   d   q

 9   d   n   l   k   g   z   k   k   Y   g   r

10   o   n   i   o   n   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k



Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
onion1 1 10 E


blueberry
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
 Y
 1   s   e   r   w   n   k   C   p   B   a   i

 2   r   u   g   s   p   a   H   h   L   w   x

 3   y   l   z   k   x   a   L   e   U   w   q

 4   h   q   x   j   w   n   O   k   E   w   z

 5   w   l   x   x   v   g   R   l   B   i   h

 6   k   a   p   t   d   g   I   q   E   d   o

 7   e   n   o   q   b   a   N   b   R   i   o

 8   e   g   q   n   s   t   E   d   R   d   q

 9   d   n   l   k   g   z   k   k   Y   g   r

10   o   n   i   o   n   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k



Enter a Word, X-Coordinate, Y-Coordinate, and Direction.
Direction In form N, NW, SW, etc.
EXAMPLE: apple 1 7 N
onion 1 10 E


blueberry
chlorine
onion
   X    1   2   3   4   5   6   7   8   9  10  11
 Y
 1   s   e   r   w   n   k   C   p   B   a   i
```

```
 2   r   u   g   s   p   a   H   h   L   w   x

 3   y   l   z   k   x   a   L   e   U   w   q

 4   h   q   x   j   w   n   O   k   E   w   z

 5   w   l   x   x   v   g   R   l   B   i   h

 6   k   a   p   t   d   g   I   q   E   d   o

 7   e   n   o   q   b   a   N   b   R   i   o

 8   e   g   q   n   s   t   E   d   R   d   q

 9   d   n   l   k   g   z   k   k   Y   g   r

10   O   N   I   O   N   v   d   t   i   s   d

11   n   y   e   c   a   d   t   b   n   h   k


GAME OVER!!! You Found All the Words!!

dg46622@ares:~/sheep$ exit
exit

Script done on 2024-07-15 21:10:45-05:00 [COMMAND_EXIT_CODE="0"]
```