File - /Users/mason/dev/CLionProjects/hangman/main.cpp

```
 1 #include <iostream>
 2 #include "random"
 3
 4 const char matrices[7][6][8] = {
 5         {
 6                 {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
 7                 {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
 8                 {' ', ' ', ' ',  ' ', ' ', ' ',  ' ', ' '},
 9                 {' ', ' ', ' ', ' ',  ' ', ' ', ' ', ' '},
10                 {' ', ' ', ' ', ' ', ' ', ' ',  ' ', ' '},
11                 {'/', '\\', ' ', ' ', ' ', ' ',  ' ', ' '},
12         },
13         {
14                 {'|', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
15                 {'|', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
16                 {'|', ' ', ' ',  ' ', ' ', ' ',  ' ', ' '},
17                 {'|', ' ', ' ', ' ',  ' ', ' ', ' ', ' '},
18                 {'|', ' ', ' ', ' ', ' ', ' ',  ' ', ' '},
19                 {'/', '\\', ' ', ' ', ' ', ' ',  ' ', ' '},
20         },
21         {
22                 {'|', '-', '-', '-', '-', ' ', ' ', ' '},
23                 {'|', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
24                 {'|', ' ', ' ',  ' ', ' ', ' ',  ' ', ' '},
25                 {'|', ' ', ' ', ' ',  ' ', ' ', ' ', ' '},
26                 {'|', ' ', ' ', ' ', ' ', ' ',  ' ', ' '},
27                 {'/', '\\', ' ', ' ', ' ', ' ',  ' ', ' '},
28         },
29         {
30                 {'|', '-', '-', '-', '-', ' ', ' ', ' '},
31                 {'|', ' ', ' ', ' ', '0', ' ', ' ', ' '},
32                 {'|', ' ', ' ',  '/', '|', '\\', ' ', ' '},
33                 {'|', ' ', ' ', ' ',  ' ', ' ', ' ', ' '},
34                 {'|', ' ', ' ', ' ', ' ', ' ',  ' ', ' '},
35                 {'/', '\\', ' ', ' ', ' ', ' ',  ' ', ' '},
36         },
37         {
38                 {'|', '-', '-', '-', '-', ' ', ' ', ' '},
39                 {'|', ' ', ' ', ' ', '0', ' ', ' ', ' '},
40                 {'|', ' ', ' ',  '/', '|', '\\', ' ', ' '},
41                 {'|', ' ', ' ', ' ',  '|', ' ', ' ', ' '},
42                 {'|', ' ', ' ', '/', ' ', '\\', ' ', ' '},
43                 {'/', '\\', ' ', ' ', ' ', ' ',  ' ', ' '},
44         },
45         {
46                 {'|', '-', '-', '-', '-', ' ', ' ', ' '},
47                 {'|', ' ', ' ', ' ', 'X', ' ', ' ', ' '},
48                 {'|', ' ', ' ',  '/', '|', '\\', ' ', ' '},
49                 {'|', ' ', ' ', ' ',  '|', ' ', ' ', ' '},
50                 {'|', ' ', ' ', '/', ' ', '\\', ' ', ' '},
51                 {'/', '\\', ' ', ' ', ' ', ' ',  ' ', ' '},
52         },
53         {
54                 {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
55                 {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
56                 {' ', ' ', '\\', ' ', '0', ' ',  '/', ' '},
57                 {' ', ' ', ' ', '\\', '|', '/', ' ', ' '},
58                 {' ', ' ', ' ', ' ', '|', ' ',  ' ', ' '},
59                 {' ', ' ',  ' ', '/', ' ', '\\', ' ', ' '},
60         }
61 };
62
63 std::string getWord() {
64     const char *words[5] = {"Hello", "Coffee", "Newspaper", "Animal", "Cheese"};
65
66     std::mt19937 mt(time(nullptr)); // NOLINT(cert-msc51-cpp)
67     std::uniform_int_distribution<int> dist(0, 4);
```

```
 68
 69     int idx = dist(mt);
 70     std::string word = reinterpret_cast<const char *>(words[idx]);
 71
 72     return word;
 73 }
 74
 75 std::string stickManProgressDisplay(const int *nFailedGuesses) {
 76     int n = *nFailedGuesses;
 77     std::string drawing;
 78     for (int i = 0; i < 6; ++i) {
 79         std::string row;
 80         for (int j = 0; j < 8; ++j) {
 81             row += matrices[n][i][j];
 82         }
 83         drawing += row + "\n";
 84     }
 85     return drawing;
 86 }
 87
 88 std::string wordProgressDisplay(char *currentLetters, int nLetters) {
 89     std::string display;
 90     for (int i = 0; i < nLetters; ++i) {
 91         if (currentLetters[i] == '/') { display += "_"; }
 92         else { display += currentLetters[i]; }
 93     }
 94     return display;
 95 }
 96
 97 int main() {
 98     int maxGuesses = 5;
 99     int nFailedGuesses = 0;
100     int nCorrectGuesses = 0;
101
102     std::string word = getWord();
103     int len = (int) word.length();
104     char guessedLetters[len];
105     for (int i = 0; i < len; ++i) {
106         guessedLetters[i] = '/';
107     }
108
109     while (true) {
110         std::cout << stickManProgressDisplay(&nFailedGuesses) << "\n";
111
112         if (maxGuesses - nFailedGuesses == 1) { std::cout << "You have 1 guess
    remaining. \n"; }
113         else { std::cout << "You have " << maxGuesses - nFailedGuesses << " guesses
    remaining. \n"; }
114
115         std::cout << "Current progress: " << wordProgressDisplay(guessedLetters, len
    ) << "\n";
116         std::cout << "Enter your guess: ";
117
118         std::string guess;
119         std::cin >> guess;
120
121         size_t pos = word.find(guess);
122         if (pos != std::string::npos) {
123             for (int i = 0; i < guess.length(); ++i) {
124                 guessedLetters[pos + i] = guess[i];
125             }
126             nCorrectGuesses += (int) guess.length();
127         } else {
128             std::cout << "Oops, wrong!\n";
129             nFailedGuesses++;
130         }
131
```

```
132         if (nCorrectGuesses == len) {
133             nFailedGuesses = 6;
134             std::cout << stickManProgressDisplay(&nFailedGuesses) << "\n";
135             std::cout << "Well done, you guessed the correct word: " << word << "\n"
    ;
136             break;
137         }
138         if (nFailedGuesses == maxGuesses) {
139             std::cout << stickManProgressDisplay(&nFailedGuesses) << "\n";
140             std::cout << "You failed. Better luck next time :)";
141             break;
142         }
143     }
144 
145     return 0;
146 }
147 
```