Hangman

Hangman is a game that is so old and well known that it needs no introduction. Played by the French aristocracy it originally involved the drawing of a guillotine. The modern traditional gallows stemmed from when French settlers brought the game with them during the American western movement.

Actually I made that up. No one really seems to know where the game came from. But the rules are simple. Guess one letter at a time until you have the word, but take too many guesses and you lose.

The listing comes with a small dictionary of tough words but if you want to use your own vocabulary lists all you need to do is make a text file with a different word on each line and pass the name of the file to the program executable when you run it. (In windows you can simply drag and drop the text file into the program for the same result.)

This game draws a minimalistic picture that you can only see if you turn your head to the side. If you think you can do better rewrite drawgallows() to draw a different picture.

Hangman was written by Joseph Larson.

/* hangman.c linsting begins: */

```
/* Hangman 2006-Mar-06
 * by Joseph Larson (c) 2008
 */

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <limits.h>
#include <ctype.h>

char missed[11] = "==========";
char dict[255][25] = {"artfull", "baton", "brazier", "create", "disgard",
  "flight", "great", "hydrant", "justify", "lacerate", "master", "niagra" ,
  "oxygen", "pieces", "question", "rough", "sachet", "trumpet", "vestibule"};
int dictnum = 19;

void drawgallows (int d) {
  printf ("\n                   %c ||\n", d < 9 ? ' ' : 'O');
  printf ("||    %c  %s%s%s%s ||\n", d < 1 ? ' ' : '@', d < 3 ? "   " : "/==",
    d < 7 ? "  " : "O ", d < 5 ? "    " : "/===", d < 9 ? "  " : "//");
  printf ("||%s%s      ||\n", d < 1 ? "--ss(O)" : "-O):o{)",
    d < 2 ? "      " : "X###]<");
  printf ("||\\   %c  %s%s%s%s ||\n", d < 1 ? ' ' : '@', d < 4 ? "   " : "\\==",
    d < 8 ? "  " : "O ", d < 6 ? "    " : "\\===", d < 10 ? "  " : "\\\\");
  printf ("||\\\\               %c ||\n", d < 10 ? ' ' : 'O');
  printf ("|==========%s||\n\n", missed);
}

void play (void) {
unsigned long guessed = 0;
char *word, input;
int c, found, nummissed = 0;

  for (c = 0; c < 10; c++) missed[c] = '=';
  c = rand () % dictnum;
  word = dict[c];
  guessed = ULONG_MAX << strlen (word);
  for (c = 0; c < strlen (word); c++)
    if (toupper(word[c]) < 'A' || toupper(word[c]) > 'Z')
      guessed |= (1 << c);
  do {
    drawgallows (nummissed);
    for (c = 0; c < strlen (word); c++)
      putchar (guessed & (1 << c) ? word[c] : '-');
    printf ("\nWhat is your guess? "); input = getche ();
    found = 0;
    for (c = 0; c < strlen (word); c++)
      if (toupper(word[c]) == toupper(input)) {
        found = 1;
        guessed |= (1 << c);
      }
    if (toupper(input) < 'A' || toupper(input) > 'Z') {
      found = 1;
      puts ("\nPlease guess a letter");
    }
    if (!found) {
      printf ("\nNope, no %c.\n", input);
      missed[nummissed++] = toupper(input);
    }
```

Listing continued on next page...

Listing continued from previous page

```
  } while (nummissed < 10 && guessed < ULONG_MAX);
  drawgallows (nummissed);
  if (nummissed == 10) printf ("\nYour man is hanged. My word was '%s'.\n",
word);
  else printf ("%s\n\nGood work! You guessed my word!", word);
}

int play_again (void) {
  char yesno;

  printf ("\nWould you like to try another one? ");
  while (!isalpha (yesno = getche ()));
  if (tolower(yesno) != 'n') return 1;
  return 0;
}

int main (int argc, char *arg[]) {
FILE *fp;
char buffer[25];

  srand (time (NULL));
  if (--argc > 0) {
    dictnum = 0;
    fp = fopen (arg[1], "r");
    while (fgets (buffer, 25, fp)) {
      buffer[strlen(buffer) - 1] = 0;
      strcpy (dict[dictnum++], buffer);
    }
  }
  puts ("Hangman\n-------\n"
  "Try to guess the word and save yourself from the gallows.\n"
  "(To see the graphic tilt your head to the left.)\n");
  do {play ();} while (play_again ());
  puts ("Good-bye.\n");
  exit (0);
}
```