Bagels

Number guessing games take many forms. There's the famous high/low variation. There's the warmer/colder, and then there's games like Bagels. Bagels is almost a mini version of Mastermind™. When you make a guess you get one of three clues; "Fermi" you've got one correct digit in the correct place, "Pico" means you've got one correct digit in the wrong place, and "Bagles" means you've got nothing, a big zero. The goal, of course, is fermi-fermi-fermi, meaning you've guessed the 3 digit number.

Bagels is written by Joseph Larson based on a BASIC game by D. Resek and P. Rowe as found in 'BASIC

BAGELS.C	You will need: a C/C++ complier .

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void) {
  char goal[3],guess[3],input[6];
  int c, pico, fermi, guesses, wins, rounds;
  float rand_const10;
  char yesno[25];
  char goodguess;

  printf ("\nBagels\n------\n"
  "In the game of Bagels the computer chooses 3 digits and you have 20\n"
  "chances to  guess it.\n"
  "After every guess the computer will give you hints to tell you how\n"
  "you're doing:\n"
  "Pico\t- You have a correct digit in the wrong place.\n"
  "Fermi\t- You have a correct digit in the right place.\n"
  "Bagels\t- You have no correct digits.\n");
  yesno[0] = 'y';
  rounds = 0;
  wins = 0;
  rand_const10 = (RAND_MAX+1) / 10;
  srand (time (NULL));
  do {
    rounds++;
    guesses = 1;
    goal[0] = (int) (rand () / rand_const10);
    do goal[1] = (int) (rand () / rand_const10);
    while (goal[1] == goal [0]);
    do goal[2] = (int) (rand () / rand_const10);
    while ((goal[2] == goal[0]) || (goal[2] == goal[1]));
    do {
      do {
        goodguess = 1;
        printf ("\nGuess %d : ", guesses);
        scanf ("%s", input);
        if (strlen(input) != 3) {
          printf ("\nYou need to input 3 digits only.\n");
          goodguess = 0;
        }
        for (c = 0; c < 3; c++) {
          guess[c] = input[c] - '0';
          if ((guess[c] < 0) || (guess[c] > 9)) {
            printf ("\nWhat?\n");
            goodguess = 0;
          }
        }
        if ((guess[0] == guess[1]) || (guess[1] == guess[2])
```

Listing continued on page 2...

BAGELS.C	Listing Continued from page 1….

```
          || (guess[0] == guess[2])) {
          printf ("\nBy the way, all digits of the digits you are trying to\n"
          "guess are unique so no duplicates are allowed in your guess\n"
          "either. Try again.\n");
          goodguess = 0;
        }
      } while (!goodguess);
      printf ("\n", goal[0], goal[1], goal[2]);
      pico = 0;
      fermi = 0;
      for (c = 0; c < 2; c++) {
        if (guess[c] == goal[c+1]) pico++;
        if (guess[c+1] == goal[c]) pico++;
        if (guess[c] == goal[c]) fermi++;
      }
      if (guess[0] == goal[2]) pico++;
      if (guess[2] == goal[0]) pico++;
      if (guess[2] == goal[2]) fermi++;
      if (fermi != 0)
        for (c = 1; c <= fermi; c++)
          printf("fermi\n");
      if (pico != 0)
        for (c = 1; c <= pico; c++)
          printf ("pico\n");
     if ((fermi == 0) && (pico == 0))
       printf ("bagels\n");
     guesses++;
    } while ((guesses <= 20) && (fermi < 3));
    if (fermi == 3) {
      printf("You got it!\n");
      wins++;
    } else {
      printf ("You ran out of guesses.\n");
      printf ("The number you were trying to guess was %d%d%d\n"
      , goal[0], goal[1], goal[2]);
    }
    printf ("\nDo you want to play again? (y/n) ");
    scanf ("%s", yesno);
  } while ((yesno[0] == 'y') || (yesno[0] == 'Y'));
  printf ("\nYou won %d out of %d games.\n", wins, rounds);
  exit (0);
}
```

Author's Notes

This game reproduces the experience of playing the BASIC version of the game. Consequently the game is pretty basic. As a beginner program it's a good one to fiddle around with, there are a lot of modifications that could be made:

- Fewer than 20 guesses definitely ups the challenge, tho in a "worst case" scenario, I've come pretty close to needing every guess.
- There's an unwritten rule that all the numbers are unique. This doesn't have to be so.
- More than 3 digits is definitely possible, but it will require a little more coding than changing the number of guesses.
- Adding more digits and removing the limitation that all numbers be unique together and you've pretty much got the game Mastermind™.