

*Programming Assignment Sheet*

|  |  |
| --- | --- |
| To: | Troy Tuckett |
| From: | Elbio Iseas |
| Class: | PRG/410 |
| Date: | 11/23/2015 |
| Re: | Individual Assignment for Week 4 |

|  |
| --- |
| Design: |
| The GuessingGameThree.cpp program is the third and last part of this guessing the number application. It is a procedural program written in C++ programming language where the computer generates a random number in the range specified by the programmer. Basically, the first screen prompts the user to enter a guess number with the purpose of matching the computer’s hidden random number. The idea is that after the first time the user guesses the computer’s number, the program starts listing past guessed numbers. There is a constant integer MAXGUESSES with a value of five in this case that means the amounts of attempts the user can guess for the hidden number, which is a number in the range 1-10. After the user enters the number, there is a module that validates the user’s typed number. If the user types an out of range number, an error message will appear asking for a good guess number in the range mentioned before. One module will generate the random number in the range 1 – 10, and another module will accept the user’s guess number. The third module will review the guess number and it will return one of three possible values, 0 if it is a match, -1 if the guessed number is smaller than the computer’s generated number, and 1 if the guessed number is bigger that the random number. There are some helping functions for positioning the cursor on screen for displaying messages and for accepting user’s input, and it will control the input and output of the program. The main function uses a loop conditioned by a boolean variable named “stay” for keeping the user in the game until he/she decides to stop playing the guessing game which it will change the value of the boolean variable “stay” to false to terminate the program. |
|
| Source Program(s) : |
| // ============================================================================  // GuessingGameThree.cpp : Defines the entry point for the console application.  // Author : Elbio Iseas 2015  // ============================================================================  #include "stdafx.h"  #include <iostream>  #include <iomanip>  #include <conio.h>  #include <cstdlib>  #include <ctime>  #include <string>  #include <windows.h>  using namespace std;  int reviewGuess (int, int);  const int MAXGUESSES = 5; // constant with the maximun number of possible guesses  int userGuesses[] = { 0, 0, 0, 0, 0 }; // int array to save the user's guesses  int attempts = 1; // if defines an int var for counting user's attempts to guess the computer random number  int currentIndex = 0; // it defines an int var for storing user's guesses in the array userGuesses  bool firstGuess = false; // it declares a boolean variable flag to switch to on when user's first guess is met  bool endLoop = false; // it declares a boolean variable flag to know when to exit the loop  // ============================================================================================================  // https://msdn.microsoft.com/en-us/library/windows/desktop/ms686025(v=vs.85).aspx  // result of the online research for the old C language function for positioning the cursor  // The purspose of this function is to position the cursor on screen  // so there is no need to use endl or leading blanks  // ============================================================================================================  void gotoxy(int x, int y) // function with the two int parameters x, and y for column, and row respectively  {  HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);  COORD coord; // it creates an object COORD named coord for the screen coordinates  coord.X = x; // it assigns the value of the parameter x (column) to the object coord.X  coord.Y = y; // it assigns the value of the parameter y (row) to the object coord.Y  SetConsoleCursorPosition(hStdOut, coord); // it sets the cursor position with the values of the object coord  }  // ====================================================================================  // this function displays a message or character in the x, y coordinates of the console  // x is the column argument to display the message  // y is the line argument to display the message  // message is the argument you want to display at the coordinates x, y  // ====================================================================================  void say (int x, int y, string message)  {  gotoxy(x,y);  cout<< message;  }  // =====================================================  // this function when called it clear the console screen  // it uses the system OS command CLS  // =====================================================  void clearscreen()  {  system("CLS");  }  // ===============================================================  // this function makes the border of a box with five parameters :  // top left screen column coordinate  // top left screen line coordinate  // bottom right screen column coordinate  // bottom right screen line coordinate  // the parameter symbol can have one of three possible values  // 0 for blank  // 1 for single line  // 2 for double line  // ===============================================================  void drawboxborder(int x1, int y1, int x2, int y2, int symbol)  {  string topleftcorner,  toprightcorner,  bottomleftcorner,  bottomrightcorner,  topsymbol,  bottomsymbol,  leftsymbol,  rightsymbol;  if (symbol == 0)  {  topleftcorner = " ";  toprightcorner = " ";  bottomleftcorner = " ";  bottomrightcorner = " ";  topsymbol = " ";  bottomsymbol = " ";  leftsymbol = " ";  rightsymbol = " ";  }  else if (symbol == 1 || symbol == 2)  {  topleftcorner = ( symbol == 1 ) ? "Ú": "É";  toprightcorner = ( symbol == 1 ) ? "¿": "»";  bottomleftcorner = ( symbol == 1 ) ? "À": "È";  bottomrightcorner = ( symbol == 1 ) ? "Ù": "¼";  topsymbol = ( symbol == 1 ) ? "Ä" : "Í";  bottomsymbol = ( symbol == 1 ) ? "Ä" : "Í";  leftsymbol = ( symbol == 1 ) ? "³" : "º";  rightsymbol = ( symbol == 1 ) ? "³" : "º";  }  if (symbol >= 0 && symbol <= 2)  {  int col, line;  say(x1,y1,topleftcorner);  say(x2,y1,toprightcorner);  for (col = x1+1; col <= x2-1; col++)  say(col,y1,topsymbol);  for (line = y1+1; line <= y2-1; line++)  {  say(x1,line,leftsymbol);  say(x2,line,rightsymbol);  }  say(x1,y2,bottomleftcorner);  say(x2,y2,bottomrightcorner);  for (col = x1+1; col <= x2-1; col++)  say(col,y2,bottomsymbol);  }  }  // =======================================================================  // this function paints a screen box with the symbol passed as an argument  // x1 is the left column value of the top left corner  // y1 is the top line value for the top left corner  // x2 is the right column value of the bottom right corner  // y2 is the bottom line value of the bottom right corner  // symbol is the character you pass as argument to fill the box  // =======================================================================  void paintbox(int x1, int y1, int x2, int y2, string symbol)  {  int col, line;  for (line = y1; line <= y2; line++)  {  for (col = x1; col <= x2; col++)  say(col,line,symbol);  }  }  // ====================================================  // This function displays the paints the user's guesses  // ====================================================  void displayUserGuesses()  {  if (firstGuess) // if first guess was done  {  paintbox(63, 5,78,17," "); // clear screen box  int line = 8; // initialize variable line of screen in 8  drawboxborder(63, 5,78,17,1); // draw box border with single line  drawboxborder(63, 5,78, 7,1); // draw box border with single line  say(65, 6,"Your Guesses"); // display title for user's guesses  for (int i = 0; i < MAXGUESSES; i++) // starts loop for displaying user's guesses  {  if ( userGuesses[i] != 0 ) // only display value if value different from zero  {  gotoxy(71,line); // position the curson according to coordinates  cout << setw(2) << userGuesses[i]; // display array element value in a two digit format  }  line++; // increment line variable for displaying array elements  }  }  }  // ============================================================================================================  // This function paints the main screen for this program  // ============================================================================================================  void displayScreenOne()  {  clearscreen(); // it clears the screen  drawboxborder( 0,1,79,3,1); // it displays a box with the coordinates provided with single line border  say( 9, 2, "T H E G U E S S I N G G A M E - P A R T T H R E E"); // it displays the title of the program  drawboxborder(16,5,62,19,1); // it displays a box with single line for the coordinates provided  say(20, 9,"Your Computer's Secret Number is : "); // it display the message at the specified position  say(20,11,"Your Best Guess Number is : "); // it display the message at the specified position  say(20,15,"Type your Guess Number (1 - 10) : "); // it display the message at the specified position  drawboxborder(0,21,79,23,1); // it displays a box with single line for the coordinates provided  }  // =============================================================================================================  // This function generates a random number in the range minValue to maxValue arguments  // and returns an integer value in that range  // =============================================================================================================  int randomize (int minValue, int maxValue)  {  int randnum; // random number  unsigned int seed; // used in association with srand() and rand() to randomize numbers  seed = time(0); // needed to generate random numbers  srand(seed); // needed to generate random numbers  randnum = rand() % maxValue + minValue; // generates a random number in the range minValue to maxValue  return randnum; // it returns the random number  }  // =============================================================================================================  // This function accepts input from the user in the range specified by the constants minnum and maxnum  // and it returns a valid integer value in the right range.  // =============================================================================================================  int userinput (int minnum, int maxnum)  {  int guess = 0; // it declares and initializes an int var with 0;  bool ready = false; // it creates and assigns the value of false to the var    say(56, 9," "); // it erases the old value from screen  say(56,11," "); // it erases the old value from screen  say(56,15," "); // it erases the old value from screen  if (firstGuess && (attempts > 1 && attempts <= MAXGUESSES)) // if attempts in range  {  displayUserGuesses(); // display list of user's guesses  }  else if (attempts >= MAXGUESSES) // if attempts attemps reached maximun allowed guesses  {  endLoop = true; // turn flag on to end loop  }  while ( !ready ) // it stays doing the loop while it is not ready  {  say(56,15," "); // it clears the screen at the position established  gotoxy(56,15); // it places the cursor in the location established  cin >> guess; // it accepts an integer value and it assigns it to the variable guess  if (guess >= minnum && guess <= maxnum) // it asks if the number entered is in the range  {  ready = true; // assigns the value of true to the boolean variable ready  attempts++; // increment attempts by one  userGuesses[currentIndex] = guess; // load userGuesses array with user's guess number  currentIndex++; // after assigning value to array, increment the currentIndex variable by one  }  else  {  say(8,22,"Type your best guess ( 1 - 10 ). Press any key to continue ... "); // it displays an error message  getch(); // it pauses the program and it waits for the user to press any key  say(8,22," "); // it erases the message  }  }  return guess; // it returns the number entered by the user  }  // =============================================================================================================  // This function returns an int value, and it accepts two arguments:  // the random number the computer created, and the guessed number the user entered.  // =============================================================================================================  int reviewGuess (int randnum, int usernum)  {  int retvalue = 3; // it defines an int variable to return a value    if (usernum == randnum) // it checks if the number guessed is equal to the computer's generated number  {  retvalue = 0;  }  else if (usernum < randnum) // it checks if the number guessed is smaller than the computer's random number  {  retvalue = -1;  }  else if (usernum > randnum) // it checks if the number guessed is greater than the computer's random number  {  retvalue = 1;  }  return retvalue;  }  int \_tmain(int argc, \_TCHAR\* argv[])  {  HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);  SMALL\_RECT windowSize = { 0,0,80,24 }; // it creates an object with the screen size  SetConsoleWindowInfo(hConsole, TRUE, &windowSize); // it sets the console with the windowSize values  displayScreenOne(); // it displays the first screen  const int maxnum = 10; // it defines a constant with the value of 99 (highest number to guess)  const int minnum = 1; // it defines a constant with the value of 1 (lowest number to guess)  char answer = ' '; // it defines a char variable with the value of ' ' for user's response  int randnumber = 0; // it defines an int variable to store the random number obtained by the computer  int userguess = 0; // if defines an int variable to store the guessed number typed by the user  int result = -1; // it defines an int variable to store the evaluation of the guessed number  currentIndex = 0; // initialize index variable for userGuesses array  bool stay = true; // it defines a boolean variable stay in the loop  bool another = true; // if defines a boolean variable another for getting a random number from the computer  firstGuess = false; // initialize variable user's first guess in false  endLoop = false; // initialize variable for exiting the loop in false  attempts = 1; // initialize variable for storing user's attempt for guessing the computer's number  while (stay) // it will stay in the while loop while stay is true  {  answer = ' '; // initialaze variable for accepting confirmation of another game  userguess = 0; // initialize variable for accepting user's guess number  result = -1; // initialize variable for storing result from function evaluation  if (another) // if there is need to generate another computer's random number  {  randnumber = randomize(minnum,maxnum); // it calls the randomize functions with the arguments  // minnum and maxnum to create a random number and it returns  // the random number generated by the computer and it stores it  // in the int variable randnumber  another = false; // set another variable in false  }    userguess = userinput(minnum, maxnum); // it calls the userinput function with the arguments randnumber  // and maxnum to accept the user's input and it stores its value  // in the int variable userguess  result = reviewGuess(randnumber, userguess); // it calls the function reviewGuess for comparing the random number  // with the user's typed number and it returns a 0 if it is a match,  // and -1 if the guessed number was lower from the random number,  // or 1 if the guessed number was higher than the random number, then  // it assigns the returned number to the variable result  if (!endLoop) // if it does not have to end loop  {  switch (result) // evaluate result variable  {  case 0: // in case there is a match with the computer's number  gotoxy(56,9); // it positions the cursor at the coordinates location  cout << randnumber; // it displays the computer's random number  gotoxy(56,11); // it positions the cursor at the coordinates location  cout << userguess; // display userguess number  answer = ' '; // initialize variable for accepting confirmation.  // it displays this message  say( 6,22,"Good job!!! Your guess was correct. Do you want to play again? (Y/N) ");  while (answer!='Y'&& answer!='y'&& answer!='N'&& answer!='n') // it will keep in the do..while loop until answer is Y/y/N/n  {  say(75,22," "); // it displays a blank at 75,22  gotoxy(75,22); // it moves the cursor to the coordinates specified  answer = getch(); // it gets a character from them keyboard  cout << answer; // accepts user's answer for another game  }  if (answer == 'N' || answer == 'n') // if the answer is N then  {  stay = false; // it assigns false to the boolean variable stay  }  else  {  another = true; // it changes the value of another to true to  // get another random number from the computer  randnumber = 1;  paintbox(63, 5,78,17," "); // clears the place specified by coordinates  for (int i = 0; i < MAXGUESSES; i++) // initialize with zero userGuesses array  userGuesses[i] = 0;  firstGuess = true; // turn flag to first guess on  currentIndex = 0; // initialize in zero currentIndex variable  attempts = 1; // initialize attempts in zero  endLoop = false; // initialize variable for exiting the loop in false  say(56,9," "); // erase screen place  }  say(2,22," ");  break;  case -1:  gotoxy(56,11); // it positions the cursor at the coordinates location  cout << userguess; // display user's guess number  // it displays this message  say(5,22,"Sorry, your guessed was too low. Press any key to keep guessing ... ");  getch(); // waits for the user to press any key  say(56,11," "); // erase screen place  say(2,22," ");  break;  case 1:  gotoxy(56,11); // it positions the cursor at the coordinates location  cout << userguess;  // it displays this message  say(5,22,"Sorry, your guessed was too high. Press any key to keep guessing ... ");  getch(); // waits for the user to press any key  say(56,11," "); // erase screen place  say(2,22," ");  break;  default:  {  // the program found an execption value  say(15,22,"Exception found. Press any key to continue ... ");  getch(); // waits for the user to press any key  }  }  }  else  {  answer = ' ';  say( 2,22,"You reached the maximun guesses allowed. Do you want to play again? (Y/N) ");  while (answer!='Y'&& answer!='y'&& answer!='N'&& answer!='n') // it will keep in the do..while loop until answer is Y/y/N/n  {  say(76,22," "); // it displays a blank at 75,22  gotoxy(76,22); // it moves the cursor to the coordinates specified  answer = getch(); // it gets a character from them keyboard  cout << answer; // display that character on screen  }  if (answer == 'N' || answer == 'n') // if the answer is N then  {  stay = false; // it assigns false to the boolean variable stay  }  else  {  say( 2,22," ");  another = true; // it changes the value of another to true to  // get another random number from the computer  randnumber = 1; // initialize randnumber in 1  paintbox(63, 5,78,17," "); // clear box according to coordinates  for (int i = 0; i < MAXGUESSES; i++) // initialize with zero userGuesses array  userGuesses[i] = 0;  firstGuess = true; // turn flag to first guess on  currentIndex = 0; // set current index to zero  attempts = 1; // starts attempts with 1  endLoop = false; // initialize endLoop variable in false  say(56,9," "); // erase screen place  }  }  }  say( 2,22," ");  say( 9,22,"I hope we can play again soon. Goodbye. Press any key to exit."); // it displays the Goodbye message  getch(); // it pauses the execution of the program until a key is pressed  return 0;  } |
| Output Results: |
|  |
| Testing: |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Test # | Random num | User's num | in range (1-10) | Too low | Too high | Match | List guesses | | 1 | ? | 5 | PASS | PASS | FAIL | FAIL | FAIL | | 2 | ? | 6 | PASS | PASS | FAIL | FAIL | FAIL | | 3 | ? | 15 | FAIL | FAIL | FAIL | FAIL | FAIL | | 4 | ? | 7 | PASS | PASS | PASS | FAIL | FAIL | | 5 | ? | 10 | PASS | FAIL | FAIL | FAIL | FAIL | | 6 | 9 | 4 | PASS | PASS | FAIL | FAIL | PASS | | 7 | 9 | 5 | PASS | PASS | FAIL | FAIL | PASS | | 8 | 9 | 6 | PASS | PASS | FAIL | FAIL | PASS | | 9 | 9 | 7 | PASS | PASS | FAIL | FAIL | PASS | | 10 | 9 | 9 | PASS | FAIL | FAIL | PASS | PASS | |
|  |