Project II

<**Code Adventure II**>

Name: Idalia Ramirez

Class: CSC 5 - 46023

RCC - Summer 2014

30 July, 2014

**Table of Contents**

1. Overview ………………………………………………….….. . 3
2. Highlights ……………………………………………………... 3
3. Summary of Code ………………………………………….….. 3
4. Pseudocode………………………………………………....….. 4
5. Code Adventure Flowchart………………………………….…. 6
6. C++ Constructs……………………………………………….... 6

**Overview**

*Code Adventure II* is an interactive game that is based on randomized results depending on the player’s input. It starts off by giving the player a total of ten coins: five gold coins and five silver coins. It also offers the player a choice from eight different doors. The doors are randomly assigned to different outcomes that may increment or decrement the total amount of coins the player has. In order to win the game, the player must find the exit with an amount greater than zero in terms of coins. If the player losses all of his/her coins before finding the exit, the player automatically losses.

**Highlights**

The game has five different possible rooms. The following contains brief descriptions of each room.

* ***The Room of Happiness***: This room is basically a sanctuary for the player. The code output displays a happy face encrypted in code and does not increment or decrement the amount of coins the player has. It simply encourages the player to keep going.
* ***The Genie Room***: This room is a jackpot for the player. Genie is a compassionate character that likes to give money away to newcomers. Genie grants the player five gold coins and five silver coins.
* ***The Getaway Room***: Depending on the player’s favorite place, the getaway room takes the player to his/her favorite destination. The con to this mini-vacation is that the trip is not free. The unexpected getaway charges the player five gold coins and five silver coins.
* ***The Monster Room***: All hail mercy! The monster room beholds the worst nightmare by containing the monster of the player’s worst fear. This is basically the end because the monster has no mercy and strips away all of the player’s coins. This is an automatic loss for the player and ends the game.
* ***The Game Room***: The game room is home to our ‘rock, paper, scissors’ fan named Bob. In this room, the player is challenged to a game of ‘rock, paper, scissors’ in exchange of five gold coins. Bob and the player go at it for three rounds, and whoever wins takes the five gold coins.
* ***The Birthplace Room***: No births in here! So do not worry. This room simply takes the player back to his birthplace where he happens to reunite with long lost family members. The family members are in need of monetary help, therefore the player is asked if he/she would like to donate five gold coins and five silver coins. The player has the option to refuse to help.
* ***The Letter Room***: In this room the player is given a letter to read. The letter has been left as a reward from the player’s favorite relative. With it, they have left five silver coins for the player to use.
* ***The Exit***: This is every player’s goal. If the player has an amount greater than zero in terms of coins and reaches this room, the player has won and finished the game!

**Summary of Code**

Project size: 339 lines of code.

Number of variables: 22 (12 characters/9 integers/ 1 constant integer).

List of libraries: <algorithm>, <cstdlib>, <ctime>, <fstream>, <iostream>, & <string>.

This game was built off of the primary game named ‘Code Adventure’. As mentioned in the previous project, the game was easy to modify and expandable; which bring us to the creation of ‘Code Adventure II’. This game has more doors to choose from, with more complex code to execute as well. The newer version has C++ concepts such as file I/O, functions (void, int, etc.), two-dimensional arrays, and string functions. The most difficult part of coding this game was reading the input file and creating a two-dimensional array from the same file input using the string class.

**Pseudocode**

System Libraries

User Defined Libraries

Global Constants

Function Prototypes

Function int RPS

Execute Here.

Declare variables

    Seed randomization

        Inputs for the game

        Basic description of game rules

        Loop the input/output code

             Total coins = gold + silver

             Player's choice - Input

            Shuffles the choice output so door assignment won't always be same

            Choice is minus 1 due to counting 0 as an integer

            Directly correlate door choice to array of randomization

            Output door assignment using switch

                Case 1

                    Calculations

                Case 2

                    Zero coins ends the game

                Case 3

                    Displays code figure

                Case 4

                  Figuratively setting tCoins equal to zero so it ends loop

                Case 5

                    Calculations

                Case 6

                    Local variable

                    Input answer to help call

                    Outputs depending on answer

                Case 7

                    Declare local variable

                    Function call

Decision upon result of game

                Case 8

                    Local variable

                    Open file from local disk

                    Read from file

                    String array displayed as 2D character array

                    Close file

        As long as the player has coins, game continues

        Asks the player if he/she would like to repeat the game

    End Here.

Function definitions

    Define variables

    Loop for 3 rounds

    Input choices

    Output results

    Calculations

 Final decision upon sum of points

**Code Adventure Flowchart**

Refer to ‘Code-Adventure II’ flowchart saved separately within the same Project 2 folder.

**C++ Constructs**

Book: *Problem Solving with C++* 8th Ed, by Walter Savitch.

|  |  |  |
| --- | --- | --- |
| Chapter | Section | Concept |
| 2 : C++ Basics | 2.1: Variables and Assignments | Declared variables in lines 20-29.  Used ‘int’ and ‘char’ as variable declarations. |
| 2.2: Input and Output | Used ‘cout’ and ‘cin’ throughout program. In other words, I/O (ex. lines 39-40 or 63-64). |
| 2.3: Data Types and Expressions | Used type ‘int’ in lines 26-29 & type ‘char’ in lines 21-24. |
| 2.4: Simple Flow of Control | Used comparison operator ‘!=’ in line 145 and ‘||’ in line 153 and “==” as well in line 153.  Also used outer do-while loop on lines 34-153 and inner do-while loop on lines 57-145. |
| 2.5: Program Style | Comments are found throughout code after ‘//’ and indenting found after loops and switch statement. |
| 3: More Flow  Control | 3.1: Using Boolean Expressions | Boolean expression used in lines 145 and 153. |
| 3.2: Multi-way Branches | Used ‘switch’ statement starting at line 77, ‘cases’ listed at lines 78, 93, 103, 121, 128. And ‘break’ statements followed in lines 92, 102, 120, 127, & 142.  Used ‘if-else’ statements in lines: 185-199, 218-229, 308-328. |
| 3.3: More About C++ Loop Statements | Used a combination of do-while and Boolean as indicated in the lines above in section 3.1 and 2.4. |
| 3.4: Designing Loops | Used concept of ‘Nested’ loops by using multiple loops. Code contains ‘blocks’ of code (ex. lines 77-143). |
| 4: Procedural Abstraction and Functions That Return a Value | 4.2: Predefined Functions | Used pre-defined functions in lines 26-27. Also seen in line 32 (srand) and 69 (random\_shuffle). Used different header files (algorithm, ctime, cstdlib, and iostream). Lines 7-10. |
| 4.5: Scope and Local Variables | Used a ‘local’ variable in line 68. |
| 5: Functions for All Subtasks | 5.1: *void* Functions | Used a function of type ‘int’ in line 216. Declared up top in line 21, and defined in lines 283-339. Variables are local only within function. |
| 6: I/O Streams as an Introduction to Objects and Classes | 6.1: Streams and Basic File I/O | Used <fstream> library. Opened file in line 247. Read from file in lines 249-251. Close file in line 265. |
| 6.2: Tools from Stream I/O | Used ‘getline’ to facilitate the file input. |
| 7: Arrays | 7.1: Introduction to Arrays | Initialized an array at line 68. |
| 7.3: Programming with Arrays | Sorted an array using ‘random\_shuffle’ in line 69. |
| 7.4: Multidimensional Arrays | Used a two-dimensional array in lines 253-258 using a ‘for’ loop. |
| 8: Strings and Vectors | 8.2 The Standard *string* Class | Used class <string>. Declared a string variable in line 240 and used I/O using ‘getline’ in line 250. |