Project I

<**Code Adventure**>

Name: Idalia Ramirez

Class: CSC 5 - 46023

RCC - Summer 2014

15 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……………………………………………….... 7

**Overview**

*Code Adventure* 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 eight coins: four gold and four silver. It also offers the player a choice from five 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 2 gold coins and 2 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 2 gold coins and 2 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 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: 163 lines of code.

Number of variables: 9 (4 characters/5 integers).

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

The main purpose for the creation of this game was the convenience in build-up possibility. It is a game that can be easily expanded and easily modified to become more complex. It is different and basically sparked from my own imagination, making it unique in comparison with other games. The game is not difficult to code; the only difficult part of programming this game is seeding the randomization and setting the shuffle function for different outputs. The shuffle function originates from the <algorithm> library, which is a concept that we have not yet learned or discussed in class. It was a function I had to research on my own to make my game easier to code.

**Pseudocode**

*Execute Here.*

*Define variables*

*Name of player*

*Favorite place of player*

*Player's worst fear*

*Player's answer to repeat the game*

*Total gold coins*

*Total silver coins*

*Total coins - gold + silver*

*Player's choice of door number*

*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*

*Total gold coins*

*Total silver coins*

*Total coins in general*

*Case 2*

*Calculations*

*Total gold coins*

*Total silver coins*

*Total coins in general*

*Case 3*

*Displays code figure*

*Case 4*

*Figuratively setting tCoins equal to zero so it ends loop*

*Case 5*

*Calculations*

*Total gold coins*

*Total silver coin*

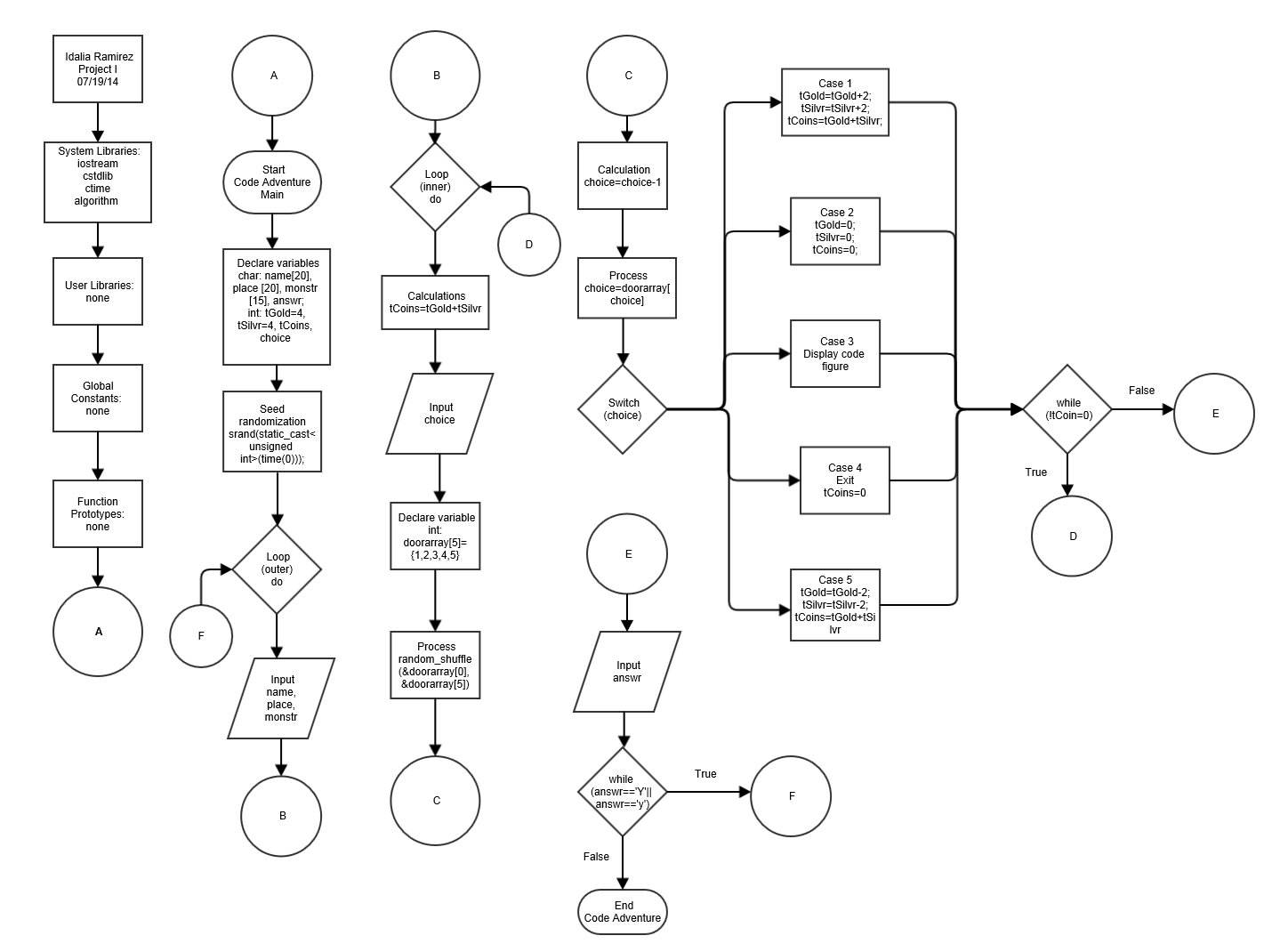
*Total coins in general*

*As long as the player has coins, game continues*

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

*End Here.*

**Code Adventure Flowchart**



**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. |
| 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. |
| 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. |