**Project 1**

**Christopher Avalos**

**11/08/15**

**CSC-11 48982**

**“Craps”**

**Introduction**

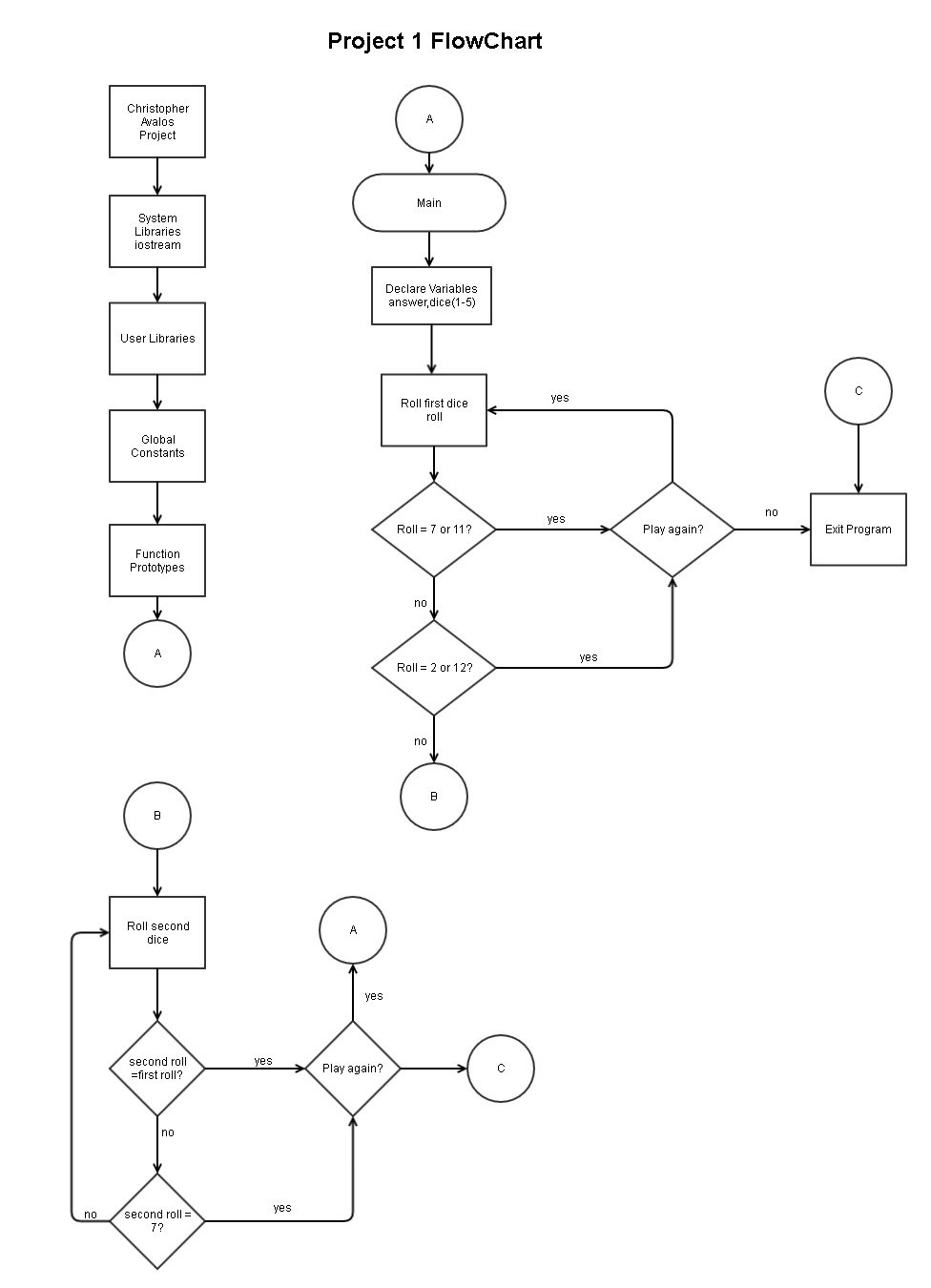
If you have ever been to Vegas or have been around a board game of any type, at some point you may already know what a dice is. Dice are six sided cubes that, when thrown, revels a number face up that ranges from 1-6. Now, this dice can be used in different amount and in different combination in order to create complex and ingenious games. However, for the sake of this project, only the use of two dice will be needed. The game the two dice will be used in is called “craps”. In the game, players take turns rolling two dice, with the attempt being to roll a 7 or an 11 the first time the dice are thrown. If a 7 or and 11 is rolled, then the player wins and is able to choose to continue rolling to try again. If the player on the first try rolls a 2 or a 12, then the player loses, and again, may roll again or hand it off to another player. If however, neither one of the above 4 numbers are rolled on the first try, then the player begins to roll again, with the intent of rolling the same number they did the first time. They continue to roll until they either roll the number they rolled the first time, or they roll a 7. If they roll a 7, it is considered a loss and the player must start all over or hand over the dice. Now, this game may seem a bit complicated being read out-loud however once in motion, the game is very simple. The odds are more than likely in favor of a person rolling a 7 than any other number. Therefore, unless a 7 is rolled the very first time, chance are the player will lose on a second roll by rolling the undesired 7. The program will recreate the rolls of dice as they are thrown, with the player losing and winning according to how the dice are rolled.



**Summary**

The program will be composed of a series of loops that will repeat themselves when needed and send out the appropriate responses according to what is going on in the game. In other words if the player wins or lose, the program will tell them, and will also give them the opportunity to roll the dice or quit whenever they want. The program will be written in C++ first and then transferred over to assembly in order to make the implementation much easier on me.

**Flow Chart**

****

**What Have I learned**

I’d like to thank all the previous programmers before me that paved the way in order to make programing so much easier. I can’t stress how many problems I’m having trying to translate c code into assembly. Something as simple as cout and cin have become multiple lines of code in order to be able to simply put out or input any type of data. Even then, it’s not going to work the first time or even show me where the problem lies! This program is giving me major headaches in an attempt to replicate that only took me a few hours to code up in C. The project is still considered very simple by many standards, and is expandable by adding the ability to bet money and mabe even involve multiple players in it. Before any of that can happen however I will need to find a way to easily put a random number generator into assembly language.