8 Ball Pool Log

Team: AL Mafia
Michael Ruddy – Team Lead
Michael Sherrick
Esai Ulloa
CIS 123
May 26th, 2020

Project Description

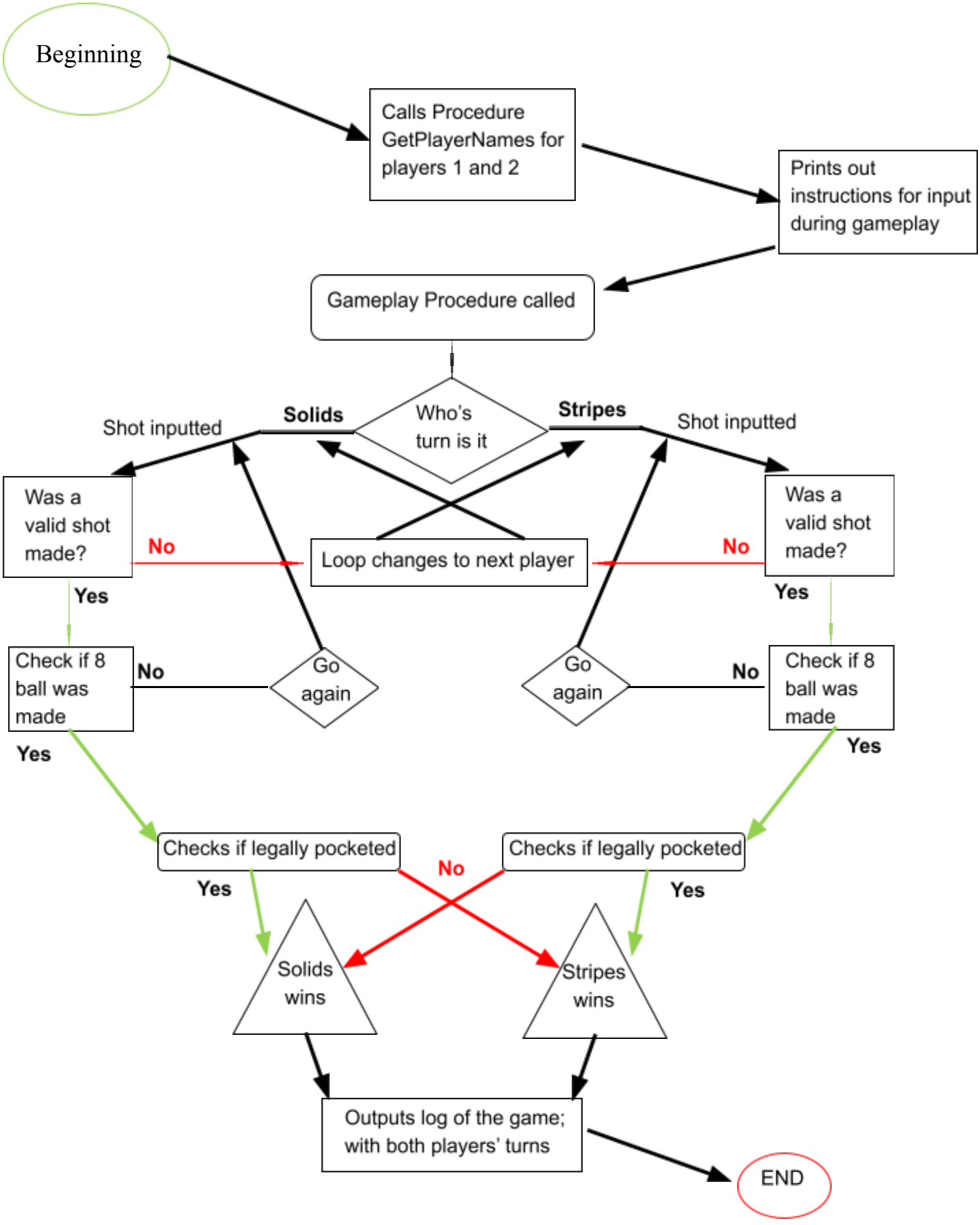
Abstract:

Our final project tracks a game of 8 ball pool and produces a log of what occurs during each game, making use of different modular files to achieve our goal. It takes both players names and as the game is played, keeps track of any scratches, and what ball, if any, was made on each turn. The program also ensures the 8 ball is not prematurely made thus giving the opposing player the win. The program ensures the correct logical path is taken as constrained by the standard rules of 8 ball. Once the match finishes our code outputs a visually aesthetic log of the game turn by turn for players to use to analyze their play and keep a memory of their favorite games.

Motivation:

When deciding on what idea to tackle for the final project, a few ideas were put on the table including an attempt at machine learning, but we chose against it in favor of something more appropriate for assembly programming. That's when we thought of this project, it's not a solution to world hunger, but it definitely solves the problem of not remembering the details of a match. Originally we wanted to work off of an actual computer pool game, but steered away due to the line limit and time constraint, but it is certainly something we would add in the future. Most often pool is played in groups and our program allows a spectator to easily record the input and provide the output to those involved in the game. Additionally, for those without a pool table at their disposal, our program could be used for "air-pool" using imaginary pool cues and imaginary balls being made or missed.

High Level View



Project Description

Project Explanation:

As discussed earlier, the purpose of our application is to track what takes place during a match by generating a document reviewing the details of each shot in the match once a winner is decided. There being different ways to play pool, we stuck with 8-ball since it's the one we all knew how to play and has rules we were already familiar with. Our group decided to plan ahead to make sure our coding procedure wouldn't be sporadic. The advantage of working on this idea in a language like assembly is that there would be no libraries or data too complex or large to work with. Each of us used our knowledge from numerous lectures to build the program in modular parts, like Lego pieces that would all fit together at the end. We'd write one procedure at a time and unit test before adding it to the main project, being careful to avoid any mistakes upon completion of the entirety of the program. The three of us used Visual Studio Live Share where we could all work simultaneously, debugging each time we added in a few lines or a procedure, using breakpoints to see where any errors were made and to make sure the register being used was storing the correct value. Live Share was quite wonky and we would not recommend this for future projects unless Microsoft improves it. Our group started with the input of the two players' names since not only would it be the first user interaction with the application, but because it was something that could be done quickly and serve as a model of how we'd write the rest of our distinct procedures. Communication through text and in real time on Zoom while on Visual Studio eased the process since we could all discuss what was being added as it happened. After completion of this project we gained valuable x86 experience, enhanced our collaborative capabilities, and created a practical and useful application which we are all very proud of.

Project References

Our main source that we kept turning to was the Kip Irvine book Assembly Language for x86 processors, which was the book we had been using throughout

the past few months for anything assembly related. We referred back to chapters we had already worked through and also some we hadn't covered completely to make sure we were writing our program as efficiently as we could. Like everybody else taking any computer science class, we also used Stack Overflow to offer some useful information when we got stuck. While this tool was helpful we made sure to remember that our minds are our most useful attribute, and wanted to be sure not to rely too heavily on Stack Overflow by stitching together code we didn't truly understand.

Responsibilities, Contributions, and Resolutions

All group members contributed to the brainstorming process and the code. Working together, we learned from one another and collaborated very effectively.

Michael Ruddy:	33 .3%
Michael Sherrick:	33.3%
Esai Ulloa:	33.3%
Total:	100%