Software Task I – Sam McKid

Problem Definition

The current market for free, light, highly customizable blackjack software is in low supply, especially one that can be downloaded without risk of malicious software and attacks. These needs caused the design philosophy of:

1. Safe
2. Low system resource requirements
3. Customizable
4. Free

As I would be the sole developer of the software, it would obviously be free for myself to use. This also allowed me to customize the aesthetics and performance of the software to my preference. This is why the modular design was selected, as it would improve testing, but more importantly would reduce the system resources required by the software. Inorder to make the software as safe as possible, it does not read in any extra files other than the image and text files specified, which are contained within the directory the program is located in. Another safety step is the lack of contact with outside sources, meaning that the software only operates within the computer.

Needs and Objectives

Needs

There are several functions the software must perform in order to satisfy the requirements set out in the problem definition. These are; Must be free, Dealer must function in a manner similar or exactly like a casino dealer, The player must be able to interact with the dealer with hit and stand commands, The cards must be dealt properly, and the graphics must be aesthetically pleasing and modern.

Objectives

These are additional features that would greatly improve the experience of the software. These are; animated interfaces, expanded playing options (split, insurance, surrender), betting, stat counts, computer players.

Boundaries

Boundaries are factors affecting the software that are outside of the control of the software. These are; user choices, hardware of the computer running the software.

Feasibility Report

Operational

The solution is highly likely to fully satisfy the requirements set out as the software will be designed and built without interference by relying on existing software and graphical design. The only constraint that may limit the software's functionality development is time, but with proper management this is unlikely to be a major issue.

Scheduling

The software solution is certain to have basic functionality by the end of the 7 week program, and very likely to fully satisfy the needs and meet all objectives. The only possible issue into whether the program

can reach its full potential is if serious issues arise requiring a complete re-development of the software, but this is likely to be mitigated by careful considerations, design and management.

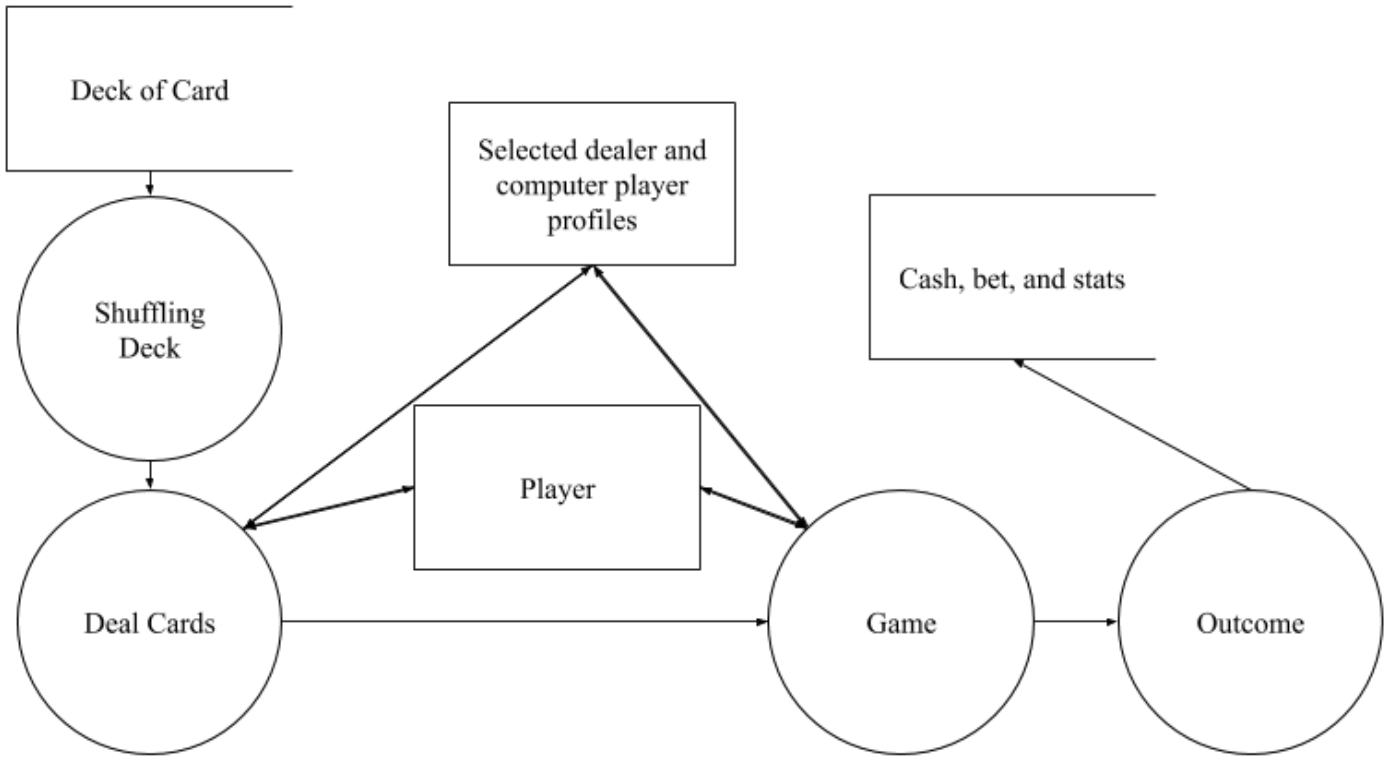
Financial

Due to the lack of any financial pressure related to the project, it is almost certain that there will be no financial restrictions to the development of this software.

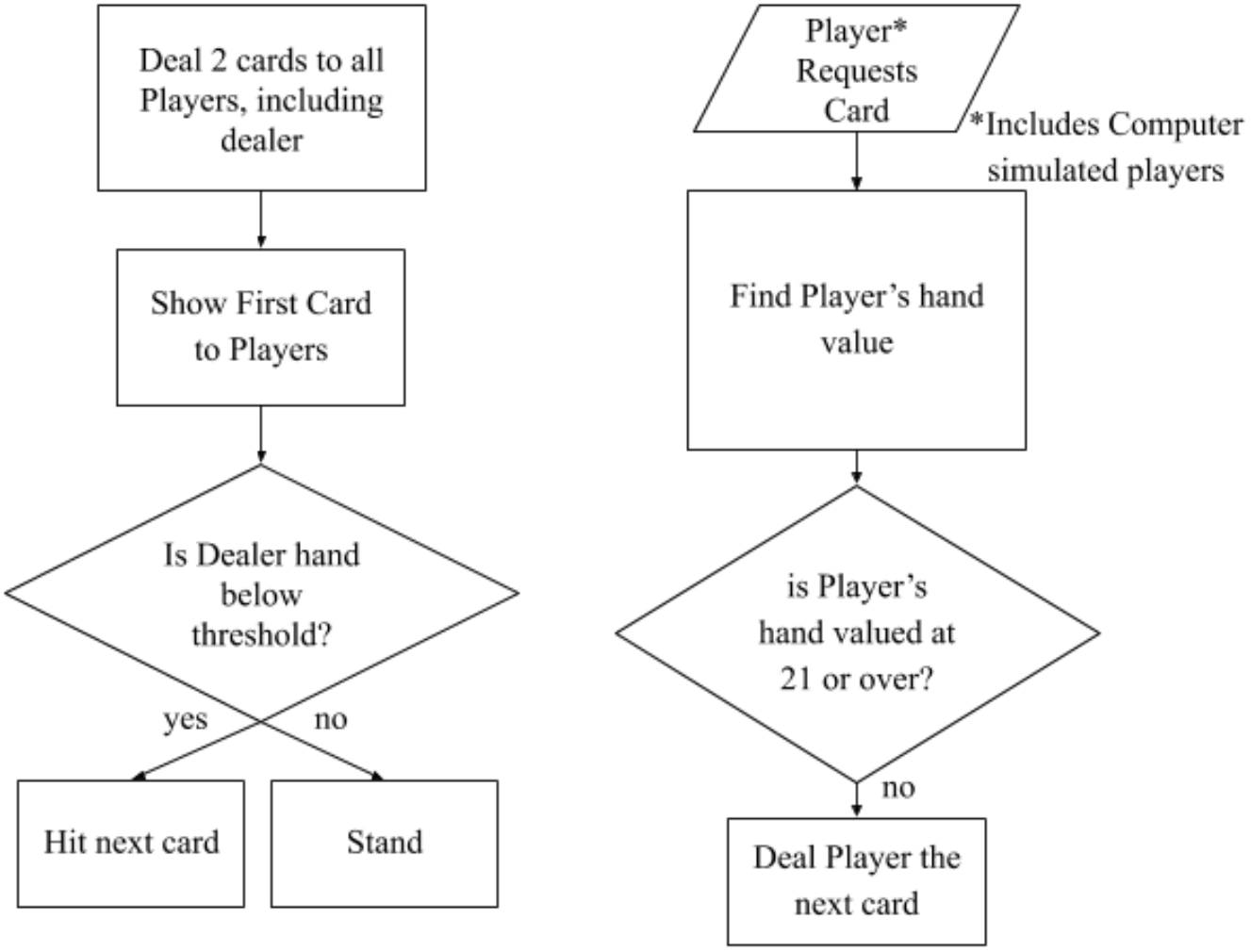
Technical

As the software is designed and developed as a highly customized software for myself, there are very little technical restrictions as the software does not need to be exported to a publicly accessible area, except as an additional storage, which will happen throughout the development as the most recent version is published to GitHub via GitKraken.

Data Flow Diagram



System flow chart of Rules for Dealer



Evaluation of installation methods

Installation methods are the way a developer would replace existing software with the newly developed software. There are four installation methods; Direct Cut-over, Parallel, Phased and Pilot.

Direct Cut-over.

This is when the new software system completely replaces the old existing software, and thus must be completely functional. While efficient at moving the software forward, the developer may be restricted in what new features they can add to retain portions of the existing software.

Parallel

Parallel is where both systems are operated simultaneously. This allows for users to adjust to the new system, while the old system remains in backup. The developer also has time to perfect the code using user feedback, without severely affecting the user.

Phased.

Phased installation is where the new software is slowly introduced, and the old software is slowly phased out by changing a part of the existing system to the new software at a time. This allows the changes to be managed as only a small part of the total software is new, and this limits the possible damages.

Pilot.

Pilot installation is where a small number of users are exposed to the new system, and provide feedback and learn the way the system works and can be used most efficiently. Once the new system is functioning to a satisfactory level, the old system is removed, and the new system is introduced to all users.

For this project, the Direct Cut-over installation method is the best, as there is a lack of existing software for this system, as many existing solutions present a risk of malicious software. All other installation methods revolve around slowly introducing the solution, whereas Direct Cut-over will mean as soon as the software is ready (passes test), it can be fully installed.

Links

Code (https://github.com/MCSQU1D/21_Blackjack)
Scots ICT Forum (https://scotsict.com/t/mckid-blackjack-2020-yr12/986/5)