Revision 1:

Author: Max Moore, Ernest Selman, Derek Savery, Andrew Danha, Eric Le Fort, Guy Meyer

Description: Creation of document and solidification of initial ideas

Date: Oct. 22/ 2016

Guy Meyer - 1320231 - meyerg

Ernest Selman - 1201291 - selmae

Eric Le Fort - 1308609 - leforte

Derek Savery - 1219142 - saverydj

Andrew Danha - 1223811 - danhaas

Maxwell Moore - 1320009 - moorem8

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Author(s) | Description | Date |
| 1 | Max Moore  Ernest Selman  Derek Savery  Andrew Danha  Eric Le Fort  Guy Meyer | Creation of document and solidification  of initial ideas | Oct. 22/2016 |

·

Overall Process workflow

o What steps need to be done and in what order?

Our intention is for the hardware and software development to work on certain parts of the project at the same time.

|  |  |
| --- | --- |
| Software | Hardware |
| 1) Simulation of Physical grounds   * Take tests for physical constraints * Design GUI for visualization * Implement basic physics engine * Test and set boundaries in safety   2) Visual Recognition (VR)   * Design physical system to take photos automatically * Design system to recognize position of balls relative to table edges * Be able to map the balls to the simulation to scale.   3) artificial Intelligence(AI)   * Design an overall decision making strategy * Have the decision be mapped into the real world, with the correct parameters to be accepted by the corresponding hardware system ~ direction, power, position, etc.   4) advanced strategy   * Add to the Artificial intelligence system the ability to predict where the ball will land, and be able to place the which ball afterwards in a strategic position, such that either out opponent's next shot will be limited, or our own would be benefited. | 1. Create requirements for the mechanical portion of the system    * Collect measurements    * Address all safety requirements    * Address all functional requirements 2. Research prior creations of the project to develop ideas 3. Create CAD model of the workspace 4. Develop design alternatives in CAD using the workspace model for a given part of the robot 5. Select alternative based on feasibility, efficiency and availability of parts (actuators, motors, etc.) 6. Purchase necessary parts or create parts in custom design 7. Build physical implementation of the robot part 8. Begin interfacing the system component with the software system   (Note: Steps 4 to 8 are repeated for the next part of robot) |

o What inputs are needed for each step and what are the outputs for that step.

|  |  |
| --- | --- |
| Software | Hardware |
| 1. inputs : physical pool table, cue, and balls for testing  Outputs: Software system to accurately simulate given shots based upon parameters such as power, direction, position, and our tables specific physical attributes. 2. Inputs: some sort of photo-device, and a pool table setup such that we can take reproducible photos from a somewhat static location. Outputs: a software system that can map the physical setup that we have to the simulation that we have created dynamically and automatically. 3. Inputs: a Functional, shot taking robot, that can accept Software inputs from out system.  Outputs: a system that can autonomously make decisions that are at least reasonable as to where and why it wants to shoot, communicate that with the physical system, and take the shot somewhat accurately. 4. Inputs: the completion and testing of an acceptable software system and Hardware interface, in short a complete project that we can show. Output: A smarter decision making Robot, that will be more difficult to play against on average. | 1. Inputs: Physical pool table Outputs: Measurements and functional requirements 2. Inputs: Measurements and functional requirements Outputs: Possible design ideas 3. Inputs: Possible design ideas Outputs: CAD model of workspace 4. Inputs: CAD model of workspace Outputs: Multiple CAD design alternatives for given robot part 5. Inputs: Multiple CAD design alternatives for given robot part Outputs: One CAD design for robot part 6. Inputs: One CAD design for robot part Outputs: Physical components needed to build part 7. Inputs: Physical components needed to build part Outputs: To-scale implementation of part 8. Inputs: To-scale implementation of part Outputs: Functional part that is fully interfaced with the software system |

o What are the acceptance criteria for the each of the outputs of a step?

|  |  |
| --- | --- |
| Software | Hardware |
| 1)   * GUI with reasonable beginning and ending locations before and after shots tested for being reasonable with real world expectations.   2)   * The balls are correctly mapped in software automatically within 1mm accuracy of what is seen by our measurements of a pre-set table. * Note this should be tested in a physical system (however we intend to set up the camera in hardware should be complete)   3)   * have a basic strategy implemented that, given the completion of the associated hardware will be able to make a decision that will hit a desired ball 90% of the time, given a straight shot. 50% of the time these these straight shots will be able to be sunk, in good conditions. In completely random conditions the system will be able to decide on and sink a desirable ball, 1 in every shots on average.   4)   * have a strategy implemented that, given the completion of the associated hardware, will be able to make users say that their shot was more difficult given the position of the cue ball 70% of the time after the robot is finished, and see at least some increase in the number of balls the robots sinks in a row, on average. | 1. Accurate measurements of the physical table, realistic functional requirements based on the physical attributes of the table 2. Realistic design ideas. Make sure the entire group is agreed on possible design ideas. 3. Representationally accurate model of the workspace 4. Realistic design alternatives. Multiple design alternatives. Make sure everyone in the group is agreed on design alternatives. 5. Have chosen one alternative which is decided to be the most feasible. 6. Have every component needed to build the part 7. Have finished part fully built 8. Have finished part fully interfaced with the software system |

· Provide some details on how each step should be done?

o What tools are you going to use? What versions?

o Are there any special instructions on settings or how to use the tools? What information to put in version control?

o Any standards you should follow? (e.g. coding standards)

|  |  |
| --- | --- |
| Software | Hardware |
| 1. The simulation of the real world will be done almost entirely within java, likely with some libraries for easier visuals and the like, also physics libraries for realistic results. 2. Visual recognition will be a crossover between a android app, and a Java app, most of the computation will likely be done in the java app, but the Android will take care of automatically taking photos at required times given certain inputs, and returning photos to us. 3. Any optimisation done in the artificial intelligence will likely be done by matlab, but the bulk of it will be contained within java, and the java application ~which will have limited communication with the java app. 4. This will require advanced optimisation algorithms, which will almost certainly be easier to implement as a complete subsystem that works in matlab. | 1. Tape measurer will be used for taking physical measurements. Metric system will be adhered to. 2. Google patents, project websites, youtube videos, past capstone projects 3. Autodesk Inventor Professional 2011 will be used 4. All CAD designs will be in scale with one another 5. 5 degrees of freedom are required for the arm. The cue ball will be shot by a shortened cue attached to a pneumatic actuator. |

**Version Control for Software Applications:**

The software Version control will be handled with Git Hub, where all of the code will be organised by the programming language that has been used for the portion, and subcategories based on modules that we think are applicable at the time.

Github will manage who added what to the code, and when, anything that is added by one piece of the software team, should be tested independently by the other piece of the Software team before continuation, comments should be added to code if there is something that isn’t ready for testing, and also if there is something that needs particular attention when testing. When testing is completed a note should be made in comments that a portion has been tested and as of when and what version, this should also be noted on github. If something is tested and seen to be unsuccessful against original expectations githubs “blame” function can be used, and cleared once the problem has been resolved.   
Comments should be kept on pieces of code where different testing iterations have taken place in form “Date, test description,test ID(i.e. AI-4.1 for module 4 test one), success/failure, suggestions”

If large changes are made based upon this then that should be noted below the testing comments in form

“Date, change descriptions, because of which test”

If there has been changes of course they need to be tested, as such it will be an iterative process that ends in a set of successful tests.

**###note for eric, here it says we should decide on certain coding conventions I have my own that I work by, but i think we should both decide on a set of standards to follow for capitalization and the like so that we can both read it as easily as possible. Which is apparently supposed to be part of this document.###**

o Who should perform each step?

· How are you using version control?

· How are you going to deal with changes to development artefacts?

o What bug tracking/change request tool?

o How do you document change requests/bugs?

o How do you classify changes?

o How do you disposition them (decide what to do, verify that they have been completed, etc)