Delta Document

Project Name:	RobotGo
Team:	CMPT370-B3
Team Member:	
	Cheng, Gong
	Hounjet, Carlin
	Lan, Shaoxiong
	Xie, Joey
	Yue, YiRui

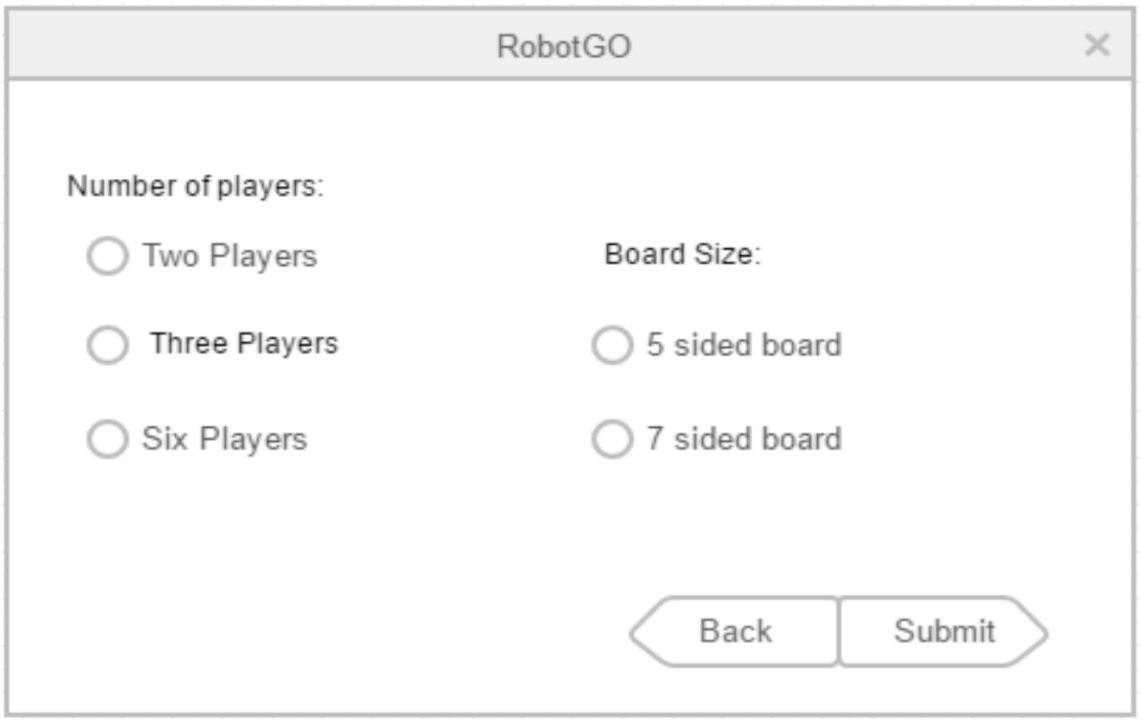
Date: Dec 08, 2016

1 Requirements that were not met.....3
1.1 Differences in interfaces.....3
1.2 Initialization Interface One from Requirement document.....3
1.3 Initialization Interface Two from Requirement document 4
1.4 Main Game Interface from Requirement document.....4
1.5 Initialization Interface in the final product.....5
1.6 Main Game Interface in final product.....5
1.7 Explanation of diagrams.....6
1.8 Changes to Robot Librarian interface.....6
1.9 Minor changes and wrap up.....6
2 Designs that didn't hold up to construction.....6
2.1 Introduction.....6
2.2 Robot mail boxes.....7
2.3 Game board size settings.....7
2.4 Architecture changes.....7
3 The Bugs That Remain.....8

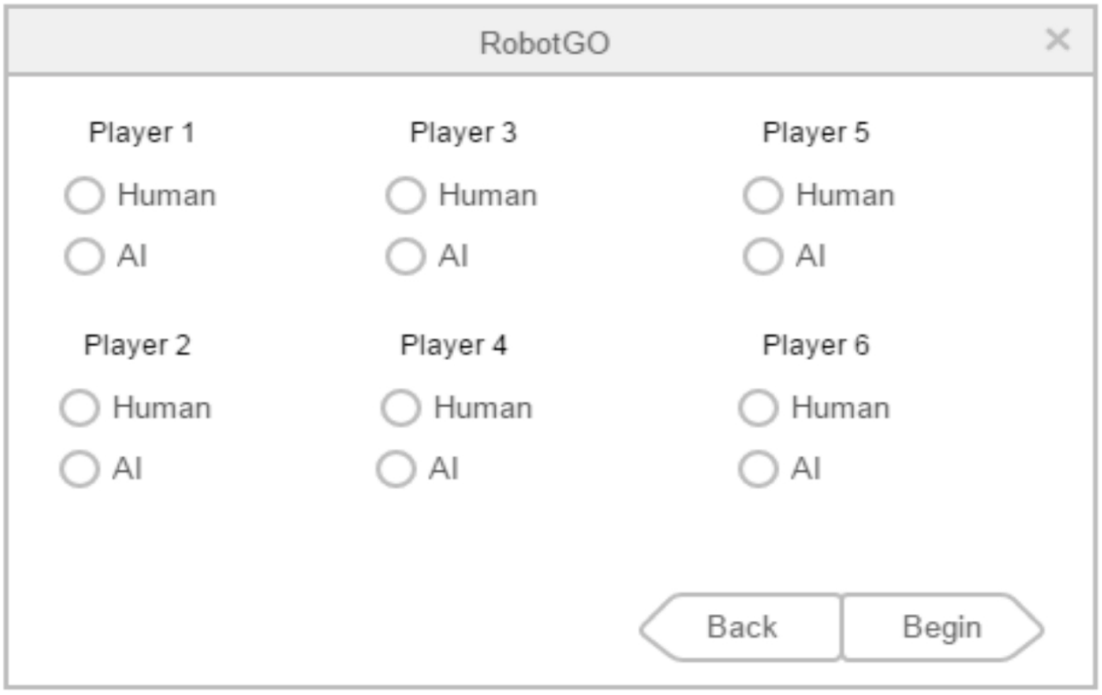
1.The requirements that were not met

1.1 Differences in interfaces

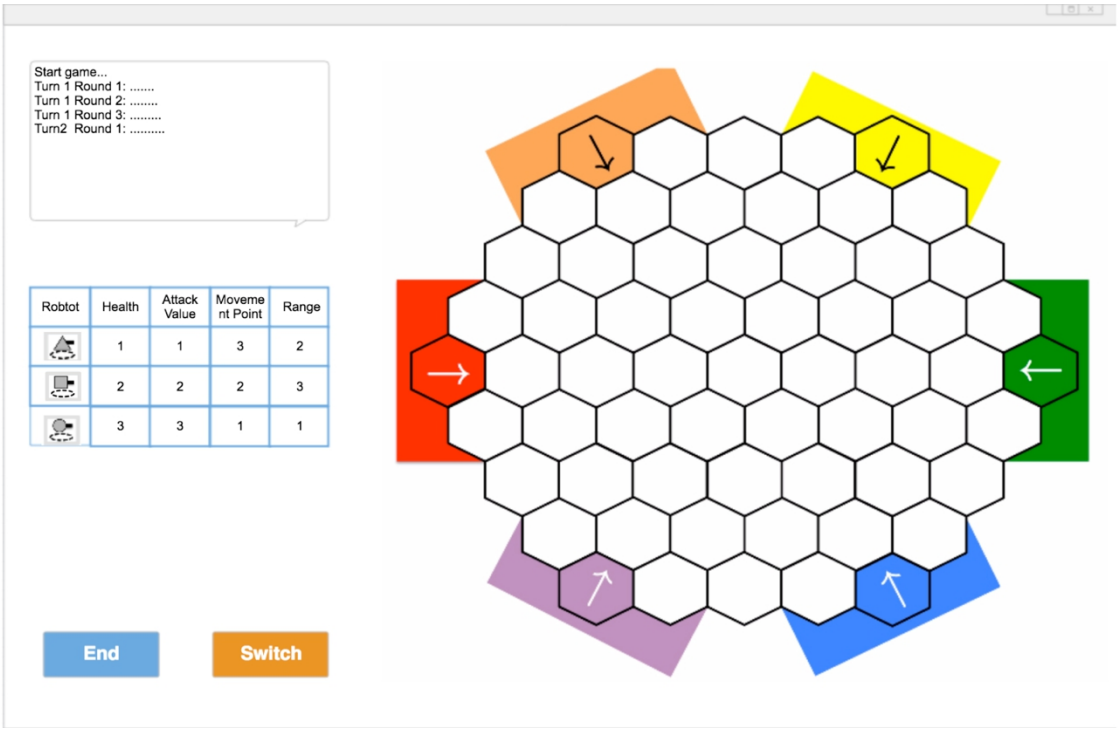
To highlight the changes between the GUI design and implementation, interface prototypes from the requirement documents are shown in Figure 1,2,3.



1.2 Figure 1: Initialization Interface One from Requirement document

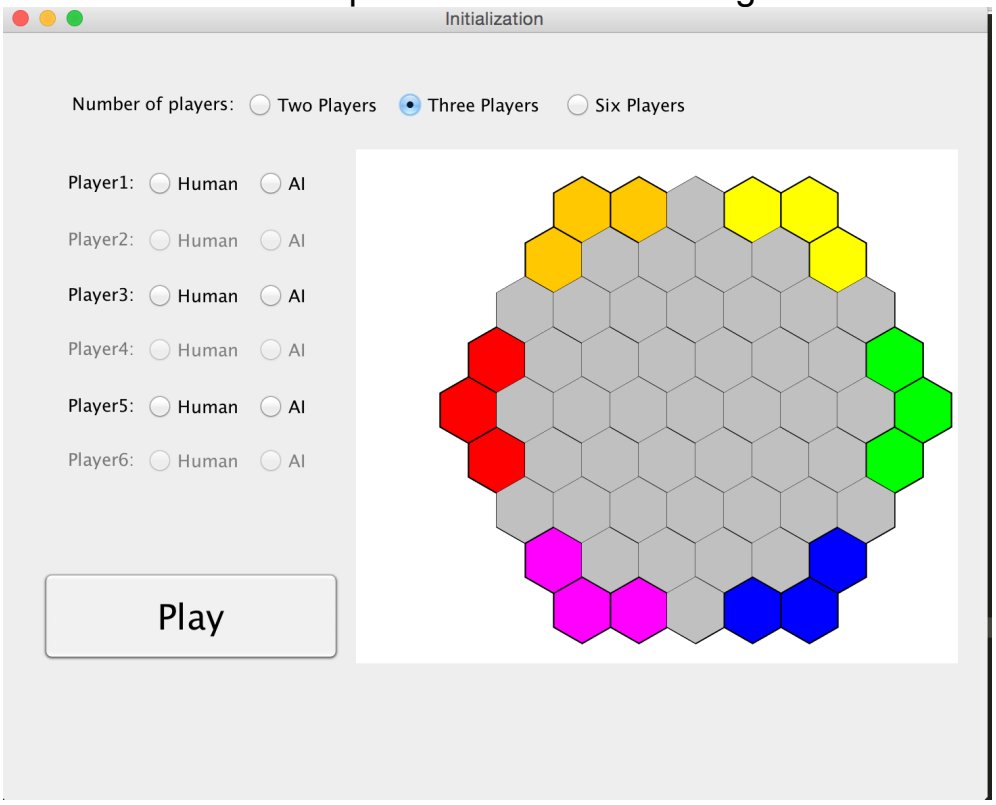


1.3 Figure 2: Initialization Interface Two from Requirement document

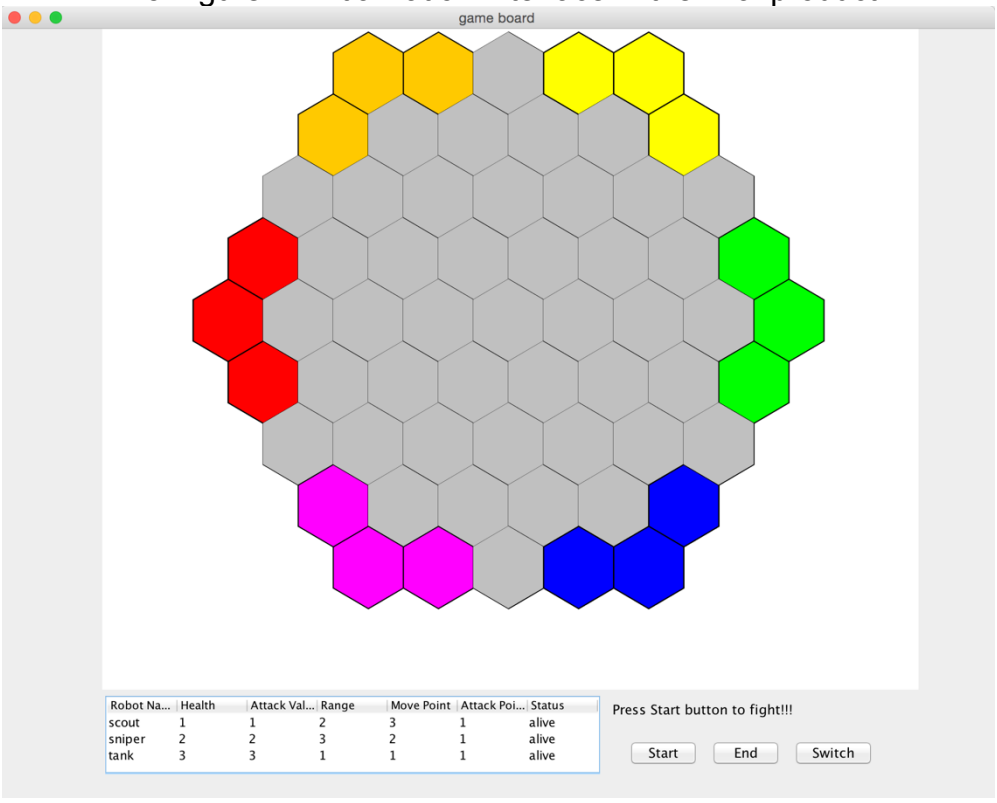


1.4 Figure 3: Main Game Interface from Requirement document.

The interfaces from our final product are shown in Figure 4 and 5 below.



1.5 Figure 4: Initialization Interface in the final product



1.6 Figure 5: Main Game Interface in final product.

1.7 Explanation of diagrams

The changes that we made to the GUI are obvious from these pictures and were done to streamline the flow of the game as well as to simplify construction. The end result maintains all features that we wanted, enhanced our construction experience, and added as some added information about each robot.

1.8 Changes to Robot Librarian interface

Another significant change between our Requirements Document and the rest of our process was the removal of the ambitious interactions we had planned with the robot librarian. Our requirements document outlined interfaces that would have enabled us to revise, register and retire robots during initialization of the game. This idea was cut after feedback was received questioning the necessity of these complex interactions. From seeing what other projects had achieved in this line of thinking during the Showcase on December 6th, we came to the conclusion that not implementing these features was the right choice as many groups had the setup implemented but their main game obviously suffered from having to design and construct these features.

1.9 Minor changes and wrap up

Other than the minor changes of how the screen doesn't darken between team's turns and the lack of a win screen (due to time constraints during construction), we are happy with our fulfillment of the requirements that we were aiming to achieve with this project.

2) Designs that didn't hold up to construction.

2.1 Introduction

The main reason our designs didn't hold up to construction was due to time restrictions and not due to design flaws or overly ambitious design goals. We feel that we were not far off in our cost estimations following the requirements document but still learned a lot about estimating the cost of time when working through these challenges. The only thing that had to be changed due to improper design during construction was a slight change in our architecture design due to redundancies that became clear when we started building the game. These challenging design factors are explored below.

2.2 Robot mail boxes

Robot's mail boxes are not fully implemented. We built the mailbox model class and the design would allow for the addition of this behavior through a few modifications in our class constructors and adding a few Forth words but unfortunately we did not reach this point. This became a low priority after running into troubles implementing certain more important aspects of the Interpreter. For example, implementing loops took a lot longer than we had expected and that is far more important for running AI behavior than the mailboxes.

2.3 Game board size settings

The option for having a game board size of 7 is not implemented. This was a choice we made after getting the 5 sided board working well and deciding to move on to implementation of the game behavior. We would have added the 7 sided option later if we had time. Unfortunately due to heavy course load for our team members, this was not possible.

2.4 Architecture changes:

1. We combined the initialization controller and main game controller that we had outlined in our design document into one class class robot controller. This became necessary when we realized that separating the functionality of the initialization controller and the main game controller was redundant, since they both accessed the same resources in the model and the view. This change exposed a flaw in our design but was caught very early on in construction and implementing it gave us nothing but benefits for the construction process.

2. We also made the decision early on in construction to not use the Tiles and Gameboard model classes we had designed. We had build them but then when we moved on to construct the GUI and controller classes, their functionality was implemented by a combination of the controller and the “Hexmesh Pointy”(the hexagon building) class. This made using the classes redundant and were therefor removed.

3. The addition of two classes to assist with GUI implementation. The Hexmesh Pointy class is based off of a heavily modified tutorial on

how to make a grid of hexagons and proved essential in setting up and updating our gameboard. The addition of an actionlistener class was not part of our design as we were unfamiliar with interfaces in Java and did not foresee the addition of this class.

3 The Bugs That Remain.

Our final product didn't end up with a whole lot of bugs due to our rigorous test driven design and constant communication of changes and challenges. However, of course, some remained. The ones that we know about are explained below.

AI is not totally working. There is a problem in the interpreter that doesn't allow conditionals inside of a loop. This makes it currently impossible to identify! all of the robots that we scan! for. We can only identify the robot at index 0 so right now if the AI scans a team member before an enemy, no shot is taken. This can be fixed b

If there are two robots remaining and they are both destroyed by shooting the same tile that they are both on, the game freezes and no one wins.

Sometimes the colored tiles that indicate which corner belongs to a team are painted white.

If an AI shoots, the shoot sound is not played.