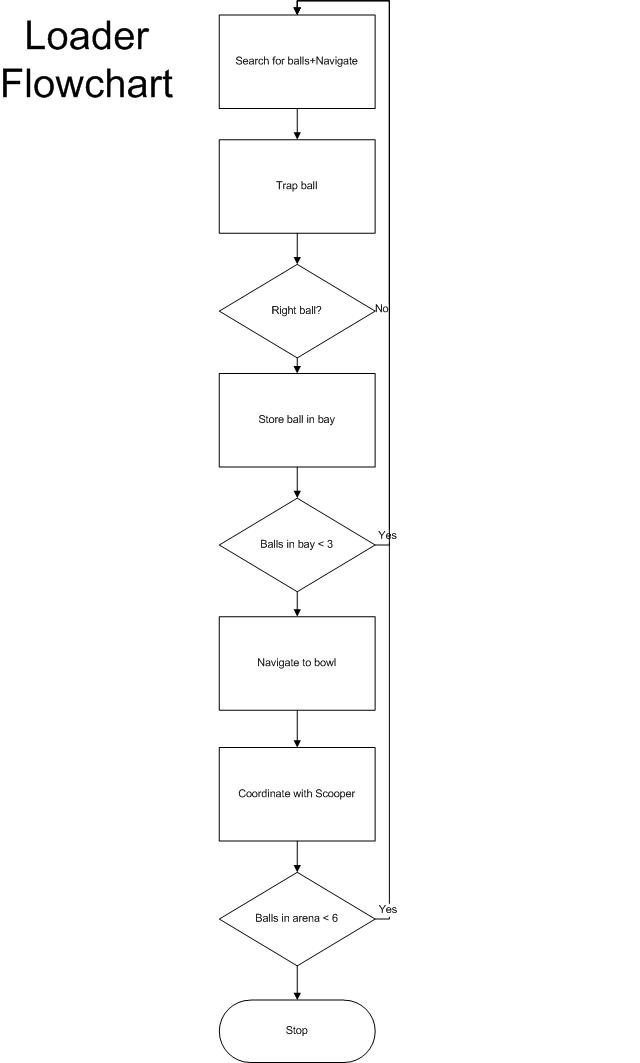
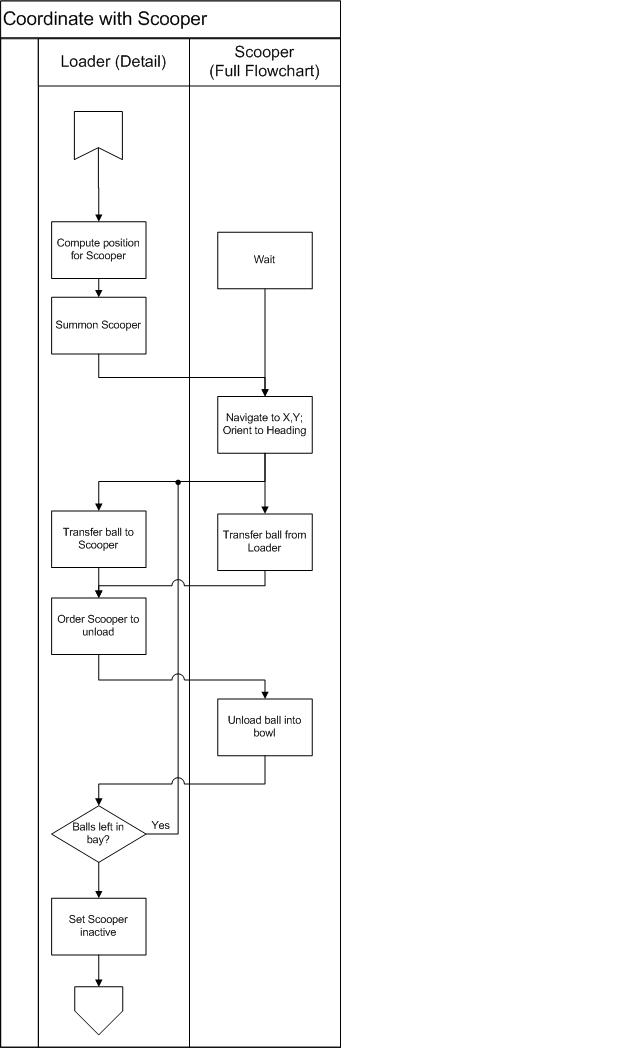
1. Design Specifications  
2. Tasks  
3. Task assignment  
4. Schedule  
5. Robot implementation details (subject to change)  
  
1. Design specifications  
  
There will be two robots used: Loader and Scooper. Loader will find, load, meet with the Scooper, and then unload the balls in the Scooper's scoop. Scooper will arrive at the bowl location after the Loader, it will get the balls in its scoop and then unload them in the bowl.  
The Loader is also the master. It has various high-level commands it can send to the Scooper. By itself, the Scooper will not act unless ordered by the Loader.  
  
Loader & Scooper  
The loader is the more complex of the two robots. It starts by searching for balls while at the same time navigating around obstacles. When it finds a ball it loads it in the front bay and checks the color. If it is the right color, it is added to the bay. Otherwise, it is thrown away, on the side. When three balls have been collected or 5 minutes have passed, the Loader proceeds to the bowl. Once there, it will summon the Scooper. Once the Scopper has arrived, the Loader will transfer one ball to the Scooper and help it keep the ball while straightening the scoop. The Scooper will then unload the ball in the bowl and then repeat until all balls have been deposited in the bowl. The Loader will then proceed to find other balls, and the process repeats. All these actions are being orchestrated by the Loader, using a communication protocol.



Figure : Software Architecture

2. Tasks  
2.1 Common tasks for both robots  
     Navigation - this implies that the robots have to move around the "arena", making sure to avoid cynder blocks, walls and balls. For the Scooper the restriction of avoiding balls is relaxed, as it might have to knock the around in order to get to the destination.  
    Navigation is composed of a couple subtasks:  
    -Odometry  - keepting track of position  
    -Control - given a movement command, making sure that it is carried correctly; i.e. minimize the difference between the desired and actual position  
   -Obstacle avoidance - given a starting position A and an end position B, move from A to B while avoiding obstacles. Note for communication protocol design: sometimes the final heading - at the destionation - will be specified too.  
  
  Error correction: using the compass and the grid lines on the floor, adjust odometry information.  
  
  Exception detection and handling - detect unforseen situations (running into a wall, slipping on the floor, being stuck etc), recover from them (get unstuck) and recover the correct odometry information (real heading, x and y position).  
  
2.2 Tasks for Loader  
     Find balls - using the navigation module, move around the arena searching for balls.  
     Load balls - if the ball is the correct one, load it into the bay (recognizing the ball is part of this task)  
     Navigate to bowl - find a path from the current position to the bowl  
     Unload ball - by communicationg and coordinating with Scooper, unload ball  
     Repeat

2.3 Tasks for Scooper  
    Wait - until Loader in Summoning state  
    Navigate to position - navigate to the position indicated by Loader. Orient towards the specified heading  
      Transfer ball - capture ball in scoop, from Loader  
      Unload ball - unload from scoop into bowl - the only step that the Scooper will perform independently  
      Repeat until no balls left in Loader  
    Repeat until no balls left / time over  
  
  
  
    
3. Task assignment  
3.1 Backlog (List of all tasks to be completed)  
  
Updating documents - recurring task - 2 hours every week  
Design and write high-level code - 8 hours  
Find and adjust initial position - 4 hours  
Assemble Scooper and Loader - 9 hours  
Write Loading/Transfer code for Loader - 10 hours  
Write Transfer/Unloading code for Scooper - 10 hours  
Write Navigation code - shared - 20 hours  
Customize Navigation code - Scooper + Loader - 5 hours  
Write ball-finding and recognition code - 7 hours  
Design communication protocol and infrastructure (physical connections) - 20 hours  
Write cooperation code - 20 hours  
Design + implement exception detection, recovery and readjustment code - 10 hours  
Design + implement rush strategy and integrate with the rest - 10 hours  
Testing, integrating - recurring task - 5 hours every week  
  
Meetings - recurring task - 2 1/2 hours every week (1 hour with Client, 1 1/2 with group)  
  
3.2 Task assignment  
  
Tony  
Developing the prototype for the robust Navigation module - 7 hours  
Updating documents - recurring task - 2 hours every week  
Testing, integrating - recurring task - 3 hours every week  
Help where needed - 3 hours every week  
  
Patrick  
Assemble Scooper and Loader - 9 hours  
Customize Navigation code - Scooper + Loader - 5 hours  
Design and write high-level code - 4 hours  
  
George  
Coordinate team - recurring task - 2 hours every week  
Assemble Scooper and Loader - 5 hours  
Finish Navigation module - 13 hours  
Write ball-finding and recognition code - 7 hours  
Design and write high-level code - 4 hours  
  
Sucong  
Write Loading/Transfer code for Loader - 10 hours  
Write Transfer/Unloading code for Scooper - 10 hours  
Design + implement exception detection, recovery and readjustment code - 5 hours  
Design + implement rush strategy and integrate with the rest - 5 hours  
  
    
Rana  
Design communication protocol and infrastructure (physical connections) - 10 hours  
Write cooperation code - 10 hours  
Design + implement exception detection, recovery and readjustment code - 5 hours  
Test + Integration - 5 hours  
  
Kamran  
Design communication protocol and infrastructure (physical connections) - 10 hours  
Write cooperation code - 10 hours  
Find and adjust initial position - 4 hours  
Design + implement rush strategy and integrate with the rest - 5 hours  
  
4. Schedule  
  
Please see the "Hours" spreadsheet on Google docs.  
  
  
5. Robot implementation details  
Loader  
  2 Bricks - 1/2 installed  
  1 Light sensor (LA) for checking balls - installed  
  1 Light sensor (LB) for detecting floor lines  
  2 Ultrasonic sensors for balls and obstacles - 1/2 installed  
  1 Compass  
  
  2 Motors used for moving the robot  
  1 Motors used for Pushing balls  
  1 Motors used for Trapping balls  
Brick 1 - Connected to LB, Ultrasonic sensors, Compass. Responsible for navigation. Controls 2 Motors used for moving the robot  
Brick 2 - Connected to LA. Responsible for Loading/Transferring. Controls 2 Motors used for Trapping/Pushing balls  
  
  
Scooper  
1 Brick  
1 Compass  
1 US Sensor  
1 Light sensor  
3 Motors