Design

Preliminary Design

Need:

- Data Flows
- Large modules/packages
- Architectural Design
  - The modules, and
  - Their interactions

Detailed Design

Need:

- Data Formats/Table Layouts
- Code interfaces (i.e. public class interfaces)
  - Method names
  - Post-, pre-, and error conditions
- Optional pseudocode for complex operations

Code Interfaces

Robot Object
int teamNumber
int robotNumber
int pointsLeft
int maxMoves
int movesLeft
int power
int health
int range
bool hasFired
void: move(coords)
void: fire(coords)
Robot: scan(coords)
Stats: getStats()

Match Controller Object
Robot [ ] robots
Cell [ ] cells
int turnNumber
int maxTurns
int currentTeam
int executionSpeed
void: stepForward()
void: stepBack()
Robot: getContentsOfCell(coords)
void: damageRobot(robot)

Views

Main Menu View
void loadWatchMatch()
void loadInstantResults()
void loadTestBench()

Team Select View
Team [ ] teams
void loadTeam(int slot)
void confirm()

Watch Match View
MatchController controller
void step()
void play()
void stop()
void setRate(int rate)
void updateLog(string log)
void (string log)

Test Bench View (inherits from Watch Match View)
void runCommand()

Match Results View
score [ ] scores *void* returnToMenu