Sprint 1

Goals:

- We should have game mechanics code fully implemented with a playable board
- An AI engine must exist using randomly-picked moves
- Server code should be implemented allowing a user to connect to the open port and play a game against the AI. Difficulty level and AI-AI should be implemented int he protocol, however it is not necessary to have the related functionality implemented yet

Backlog

Task	Time Finished
Create Game class	1
Create State class	1
Create Board class	1
Parser: Take input	2
Parser: Split command on delimiters	2
Parser: Check command for validity	3
Parser: Direct command to function	5
Print out board	3
Undo function	7
Create left operator	4
Create right operator	4
Create forward operator	4
Check operator validity	5
Perform move	6
Termination condition check	9
Report game results	5
Open port on server	9

Task	Time Finished
Read input	5
Check for password authentication	8
Send output	3
Find possible moves for a piece	4
Random Algo: Check obvious moves	8
Random Algo: Pick random move	8
Set base structure for difficulty	7

Nathan is working on Game Mechanics

Game Mechanics

- Create `Game` class to handle a game
- Create `State` class
- Create `Board` class

Jonathan is working on Parser

- Parser
    - Take input command
    - Split command on delimiters
    - Check command for validity
    - Direct command to appropriate function if valid
- Print out board
- Undo function(revert to previous state stores in vector of States
- Operators
    - Create left operator

    - Create right operator
    - Create forward operator
    - Check operator validity
- Perform move

Victor is working on Termination Condition Check

- Termination condition check (is the game over?)
- Report game results

Ryan is working on Game Server

Game Server

- Open port on server
- Read input
- Check for password authentication
- Send output

Ryan is working on AI Engine framework(Random algorithm implementation now)

AI Engine

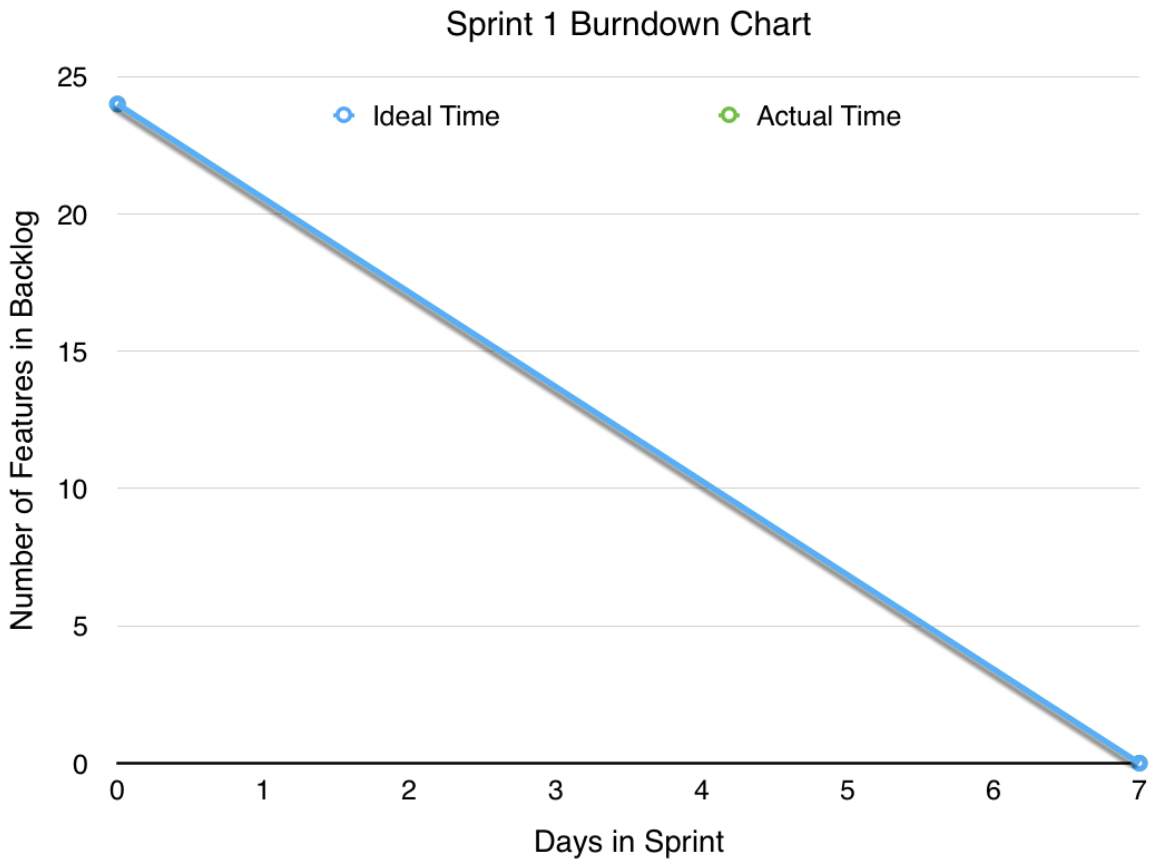
- Find possible moves for a piece
- Random Algorithm
    - Check for obvious moves to make(if a piece can be overtaken)
    - If no obvious moves to make, pick one at random

- Setup base structure for difficulty(assign all difficulties to random algorithm for the time being)

Burndown Chart

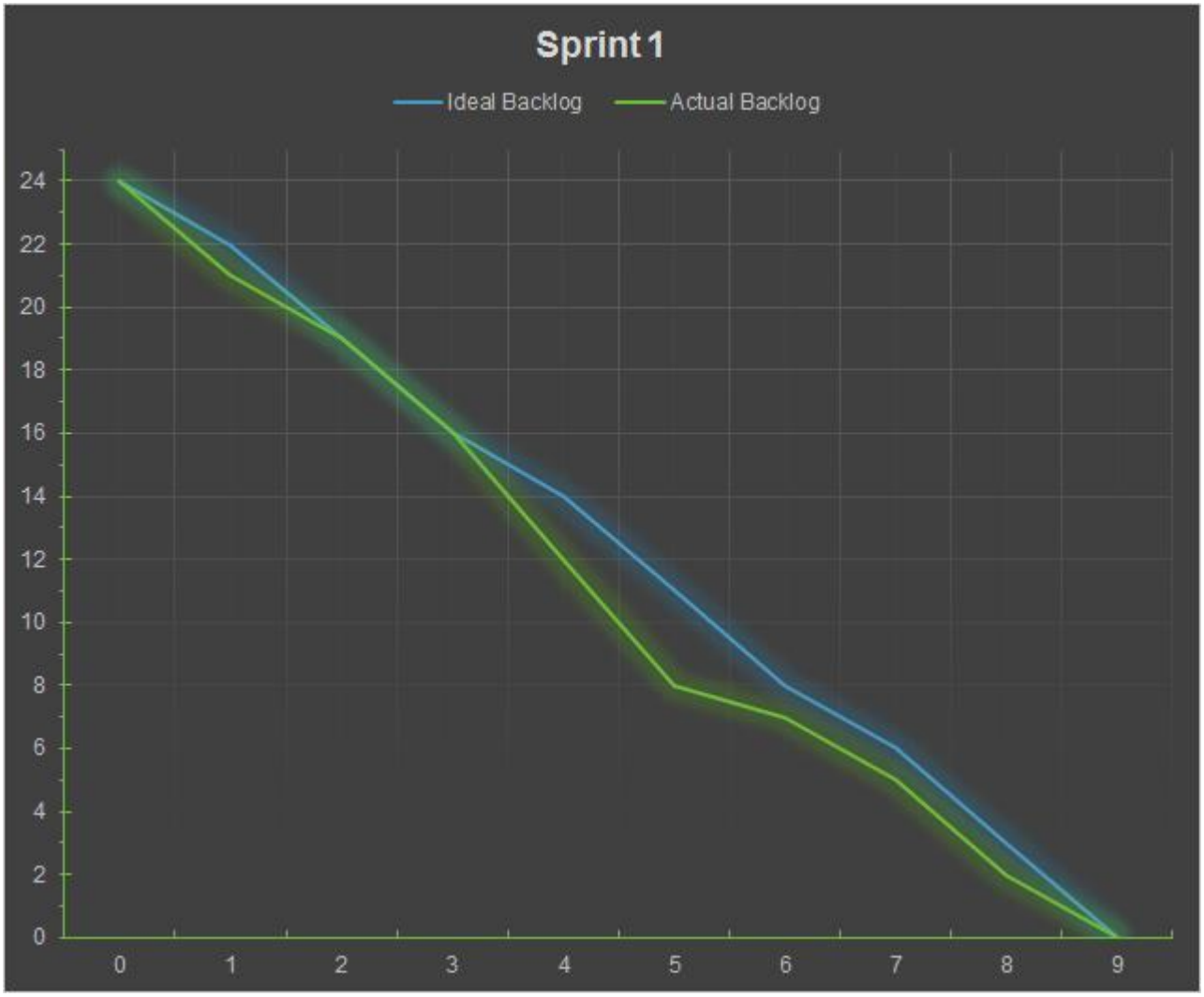
First

Deliverable:



Second

Deliverable:



Time Spent	Ideal Backlog	Actual Backlog
0	24	24
1	22	21
2	19	19
3	16	16
4	14	12
5	11	8
6	8	7

Time Spent	Ideal Backlog	Actual Backlog
7	6	5
8	3	2
9	0	0

Sprint 2

Goals:

- AI engine should implement MIN-MAX algorithm
- AI engine should implement Alpha-Beta Pruning
- Design and implement a cut-off state(depth limit)

Backlog

- MIN-MAX Algorithm
- Update Medium difficulty to use MIN-MAX algorithm
- Alpha-Beta Pruning
- Update Hard difficulty to use Alpha-Beta Pruning

Sprint 3

Goals:

- Updated version of design document
- Implemented GUI client
- Post production notes

    - Changes made to the design and why, difficulties, solutions, lessons learned
    - Individual workload percentages with brief description of each user's contribution
    - Development log

Backlog

GUI client

- Connection to server
- Send move to server
- Interpret server responses
- Display board