Sprint 3 Retrospect:

Toward the final sprint, we encountered the most challenges compared to the other two sprints. The first challenge we faced was with the UI. This posed problems because we were forced to learn an unfamiliar frame. Java is event driven which made things harder than it should have been. Trying to learn this, as well as, executing the specific requirements needed to complete the project made this very time consuming. The second challenge we encountered was the minimax tree. It was a challenge at the beginning because we did not know how to associate the data structure. We understood the concept; however, implementing it and figuring out how it fit our program was definitely the hardest part. The final main challenge we encountered was the client-server portion. The client would not receive a message until it had written something. Figuring out how to better implement the communication was the difficult part.

Product Backlog:

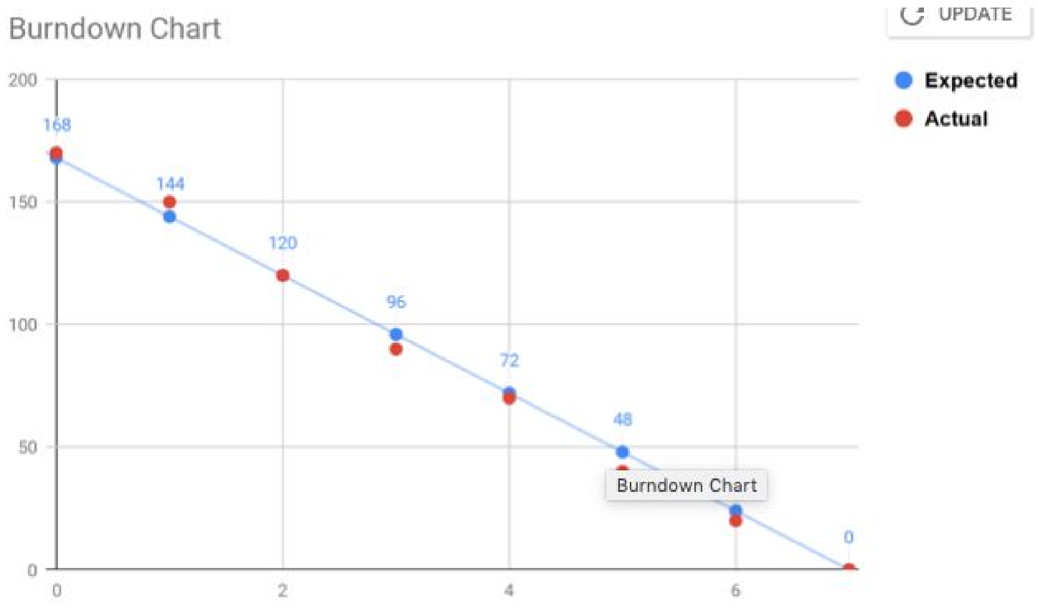
1. Research game theory for chess
    a. Chess rules
    b. Min and max tree
    c. Alpha beta pruning
2. Building GUI interface
    a. Greeting message
    b. Correct board
    c. Placement of colored pieces
    d. Scoreboard
    e. Create game creation menu
    f. Ending game popup
3. Move generation
    a. Move by defined click-and-move process
    b. Display of the current board state
    c. Stop movement generation when an opponent piece is run into
    d. Stop movement generation when a similar piece is run into
    e. Give every possible movement allowed on the board for selected piece
4. Logic of move generation
    a. Checked to determine if either player has won/lost (server-side)
    b. Determine if maximum number of moves has been exceeded
    c. Check for invalid moves
    d. Check for color of the pieces
    e. Check to see if movement is within a certain bound (game board)
    f. Computer randomly selects a valid move
    g. Computer looks ahead one move in tree

   h. Board evaluation (winner/loser)
   i. Identify correct color
   j. Correct attack movements
   k. Implement selectable color

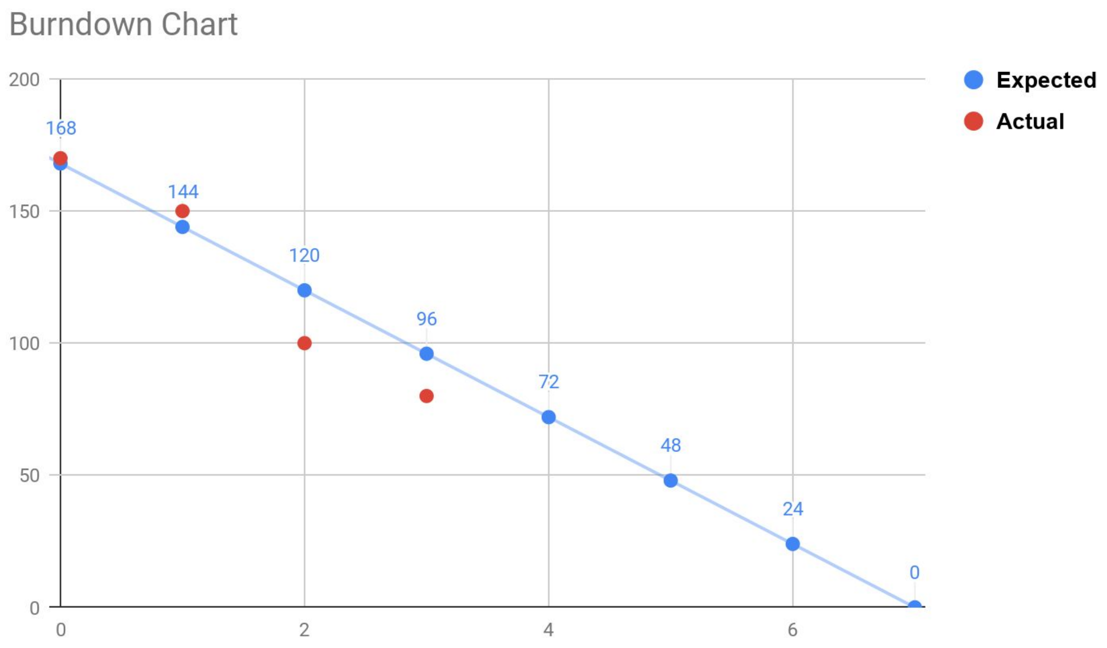
5. Server/Client Communication
   a. Create server connection menu
   b. Threaded Server communication with server logic (server sends board state after every move)
   c. Threaded GUI to client calls
   d. Build establish player interface for local and remote players/AI’s
   e. Establish better separation of concerns for GUI and clients
6. Opponent Behavior
   a. Create an interface for receiving opponent actions
   b. Create a side selector
   c. Implement improved AI logic
   d. Implement local opponent controls
   e. Implement remote opponent communication system
   f. Add UI widgets to support features
   g. Add multiplayer capabilities (no computer)

Burndown Chart:

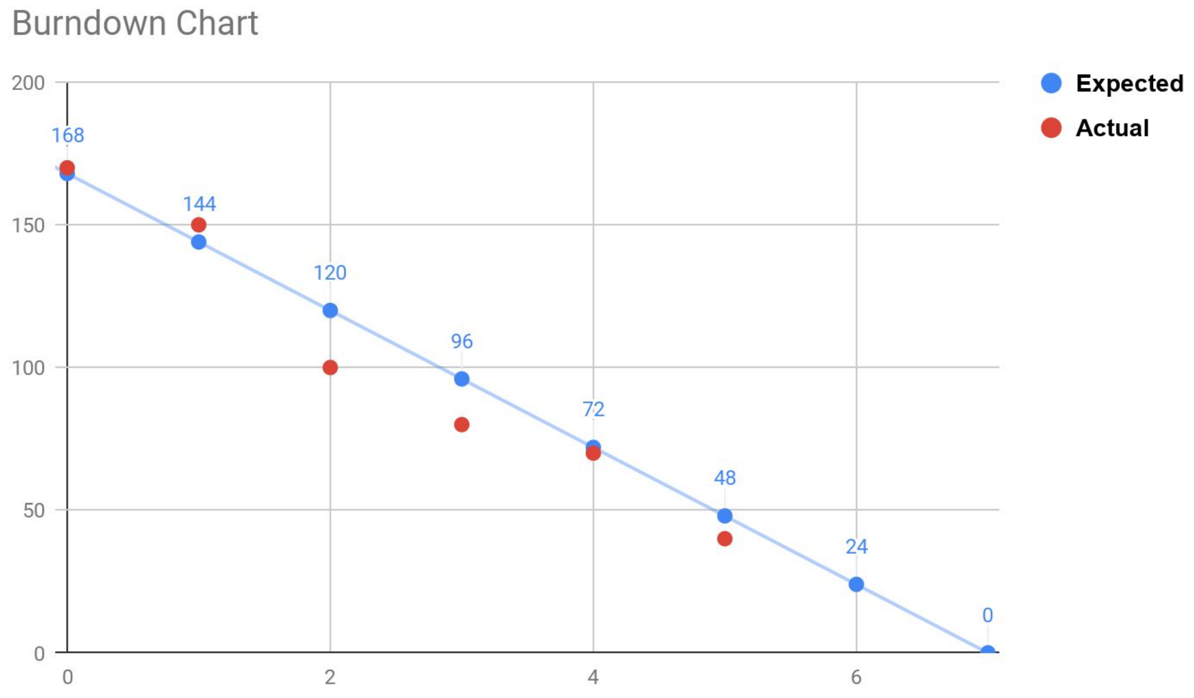
First Chart:



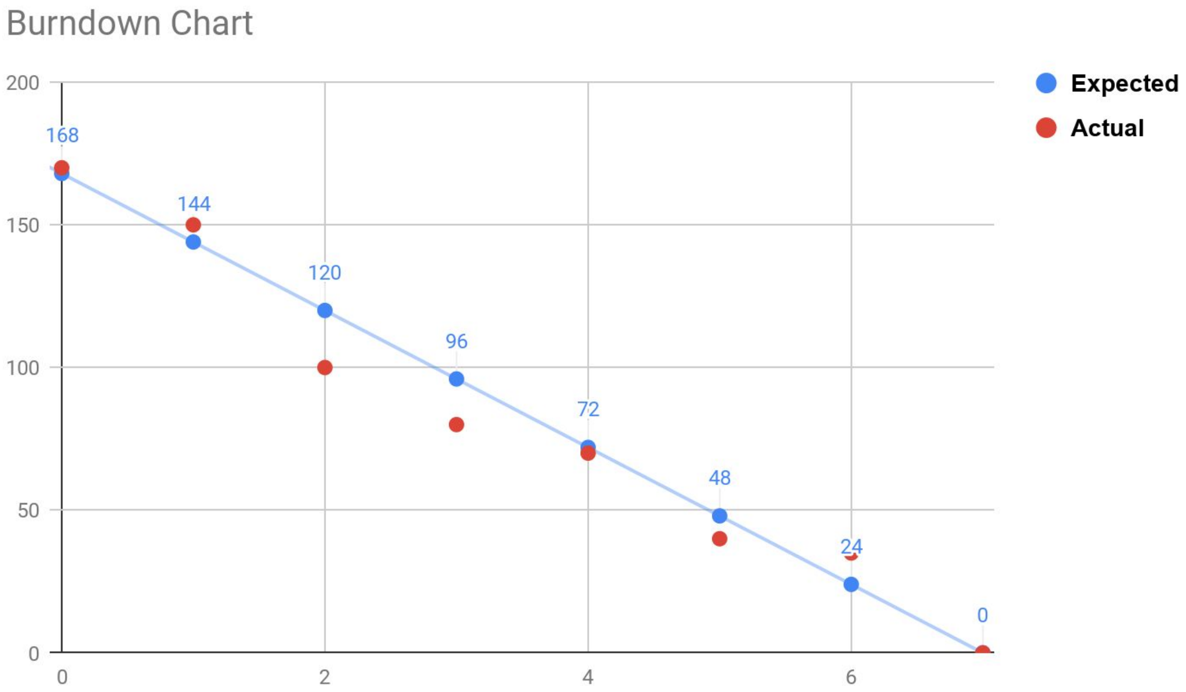
Sprint 1 Chart:



Sprint 2 Chart:



Sprint 3 Chart:


Burndown Chart
● Expected
● Actual