Product Backlog

1. Game mechanics (20 working unit)

* board and game state representation
  + 2D array board[8][8] (Huy)
  + Board setup (Huy)
  + Display (Huy + Josh)
  + Move (Huy)
* Check if a move is valid (JuAune)
* move: once a move is made,
  + update the board (flip tiles color) (Huy)
  + show next possible moves (JuAune)
* termination condition detection (can any more pieces be placed?)
  + No possible move -> skip turn (Josh)
  + Board is full -> end game (Josh)
* undo/redo
  + undo stack (Huy)
  + redo stack (optional)
* game result report (win/lose/draw).
  + count tiles + winner (Juaune + Huy)

2. Game Server (10 working unit) (Josh)

* allows clients to connect (use telnet)
  + Create socket, bind, listen, accept, send/recv messages.
  + Recv client move
  + Make an AI-move
  + update server board
  + send AI-move to client
  + Shutdown, close.
  + Run full game using AI engine and game mechanics

3. Game AI (30 working unit)

* random player
  + pick a random move from available moves (Josh)
* min-max with limited depth
  + Min-max Algorithm (Huy + Josh)
  + evaluate moves. ( Huy)
  + Integrate AI to server code, user set difficulty and ai color. (Josh & Juaune)
* alpha-beta-pruning (optional)
* [optional] more advanced game AI, customize evaluation function.

4. GUI client (30 working unit) - use Qt with C++

Allows users to connect to the server using a GUI interface

* GUI client
  + connect to the server + send/recv msg.
  + Board display
  + Move
  + Undo
  + Flip tiles color after a move
  + Show Possible moves
  + End Game

5. Agile computing methodology (10 working unit) - teamwork