COMP6710 ASSI2 PRESENTATION

1. NAME OF STUDENTS WHO CONTRIBUTED:

Name: Shiyu Chen

UID: u5959123

Name: Ujjwal Bansal

UID: u6808986

Name: Shaocong Lang

UID: u6499863

2. SUMMARY OF WHAT WE DID:

- Basic implementations:
  - ✓ Determine whether a piece placement is well formed (Task2)
  - ✓ Determine whether a placement string is well formed (Task3)
  - ✓ Determine whether a placement string is valid (Task5)
  - ✓ Determine what piece placements are viable from a given starting point with a given challenge (Task6)
- A simple placement viewer (Task 4):
  - ✓ The viewer can display images of pieces in the window
  - ✓ Translate piece positions to x and y positions in the window.
  - ✓ Display images of pieces so that their origin is in the correct place.
  - ✓ Display images so that their origin is in the correct place and their orientation is correct.
  - ✓ Break placement strings into piece placements.
  - ✓ Fix the makePlacement()method of the Viewerclass so that it works correctly.
- A playable game:
  - ✓ At the beginning of the game, images of pieces and board is displayed in the window
  - ✓ The challenge is displayed on board semi-transparently
  - ✓ Ensure that the player can place pieces on the screen provided by your Board:

a. *Piece is draggable*
b. *When a piece is on board, piece will be snapped into the nearest location*
c. *When a piece is not on board, piece will be snapped back to the initial location*
d. *When the piece placement not match the 'challenge' the piece will be snapped back to initial location*
e. *Pieces can be rotated*
f. *The level of difficulty can be modified*

3. A SCREEN SHOT OF YOUR GAME:

- rules on hint:
  - ✓ press slash a transparent piece will occur to give hint on the next piece placement
  - ✓ if the current piece placements exist pieces that violate the only solution, the hint will be given to one of those invalid pieces. That means, show the correct transparent piece placement of one of those invalid pieces.
- Rule of challenge:
  - ✓ Designed by the choose by the amount of time it takes to find the solution, that is the total trying amount needed in each challenge
  - ✓ The more trying amount needed, the more difficult