**COMP 3981 Project**

**Part 1**

**Team Members**

Mangat Toor | A00991501

Nicolas Rodriguez | A01335359

Tomasz Stojek | A00989515

Vitor Guara | A01339914

**Contents**

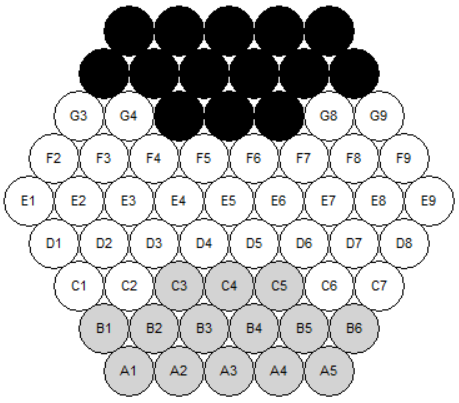
**I     Game Board Representation**

**II    Moving Notation**

**III   Problem Formulation**

**IV Team Member Contribution**

**I     Game Board Representation**

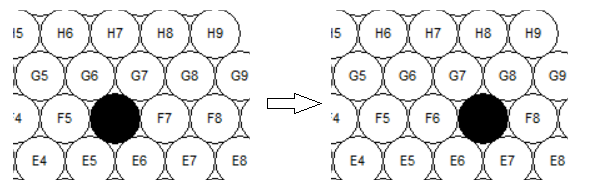
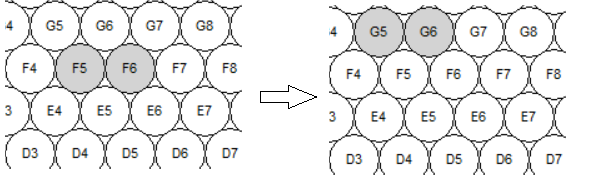
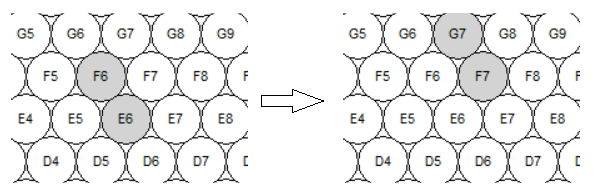
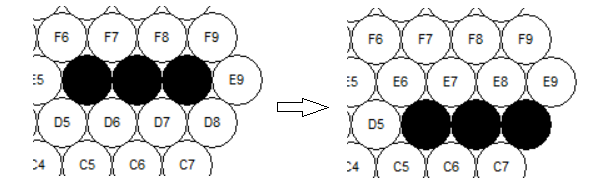
The game board is represented by 61 circles; Each circle has a tag made of a letter and a number. Each circle can contain a white or a black marble.  


**II    Moving Notation**

Moves are represented as [[X], Z], where:

* X: Contains coordinates of each marble that will be moved in an array
* Z: Direction of movement (R, L, UL, DL, UR, DR).
  + R: Right
  + L: Left
  + UL: Up Left
  + DL: Down Left
  + UR: Up Right
  + DR: Down Right

**Example Notation with Pictures**

* Single Black Marble Move Right: [[F6],R] 
* Double White Marble (Straight Line) Move Up Left: [[F5,F6],UL] 
* Double White Marble (Diagonal) Move Up Right: [[F6,E6],UL] 
* Triple Black Marble Move Down Right: [[E6,E7,E8],DR] 

**III    Problem Formulation**

**a. State Representation**

The state is represented by a dictionary({circle\_name: circle\_object}) of circles that can contain a black or white marble. With each move, the marble(s) are moved to the intended circle. It also has a captured\_mables attribute that stores how many white or black marbles are captured.

state = {

captured\_mables: [w, b],

circles: {

"I5": {marble: marble\_object(black | white)},

"H5": {marble: none},

"G5": {marble: none},

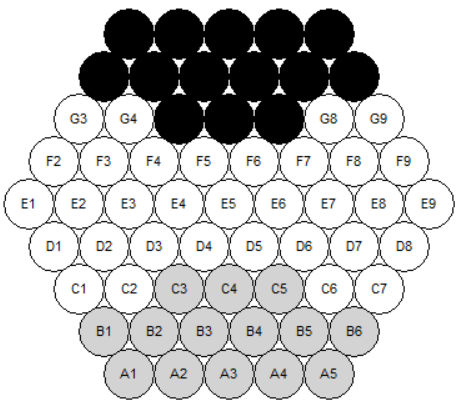
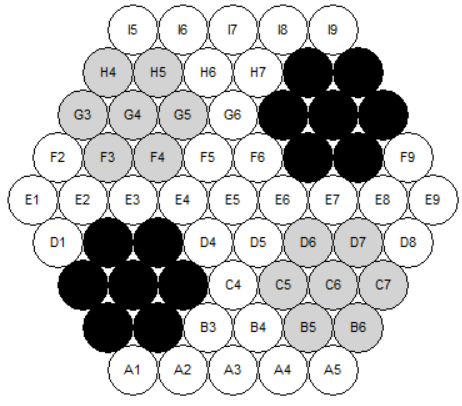
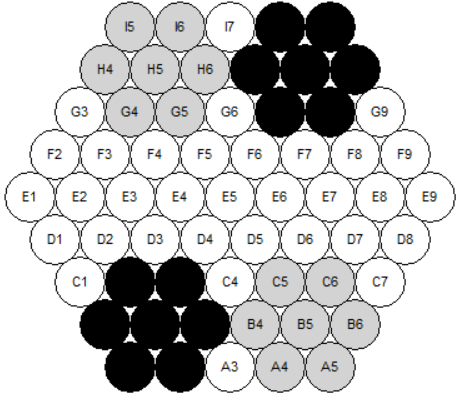
...,

}

}

**b. Initial State**

The initial state can be one of three states:

1. Standard  
   
2. German Daisy  
   
3. Belgian Daisy  
   

**c. Actions**

The actions are defined using the move notation and involve moving marbles (1 to 3) to one of six directions (as defined in part II).

**d. Transition Model**

| **Actions** | **Resulting State** |
| --- | --- |
| [[marble(s)],R] | Move each marble to: Circle(old\_l, old\_num + 1) |
| [[marble(s)],L] | Move each marble to: Circle(old\_l, old\_num - 1) |
| [[marble(s)],UL] | Move each marble to: Circle(old\_l + 1, old\_num) |
| [[marble(s)],UR] | Move each marble to: Circle(old\_l + 1, old\_num + 1) |
| [[marble(s)],DL] | Move each marble to: Circle(old\_l - 1, old\_num - 1) |
| [[marble(s)],DR] | Move each marble to: Circle(old\_l - 1, old\_num) |

**e. Goal Test**

The goal test consists of checking if any player has gotten six of the opposite marbles out of the board.

**IV  Team Member Contribution**

1. **Mangat Toor**
   * Worked on timer logic for the game.
   * Worked on move notation and problem formulation.
   * Worked on move logic.
   * Worked on pause functionality.
   * Worked on documentation.
2. **Nicolas Rodriguez**
   * Worked on moving single marbles.
   * Worked on axis checking logic when moving marbles.
   * Worked on creating the initial GUI.
   * Worked on move history.
   * Worked on selection logic.
3. **Tomasz Stojek**
   * Worked on logic for moving multiple marbles.
   * Worked on logic for selecting multiple marbles.
   * Worked on move history.
4. **Vitor Guara**
   * Worked on the skeleton of the classes mapped to the GUI.
   * Worked on initial board positions.
   * Worked on undo and start buttons.
   * Worked on saving the states for the undo button.
   * Worked on displaying the score.