

Final Projects – AP Computer Science AB

- 1) Memory – Add pictures to the memory game instead of just colors. Use the ImageIcon class.

```
ImageIcon picture=new ImageIcon("logo.jpg");
```

- 2) Battleship

- a) Figure out the directions
- b) Add error checking for placing the ships
- c) End the game when a winner is found
- d) Have the computer be smarter because it is pretty dumb right now.

- 3) Write the game of Othello – Use GridWorld project instead of the JMBS environment –

In these exercises you will work with an 8x8 bounded grid that contains single character strings. The grid itself can be declared with the statement:

```
Grid<String> board = new BoundedGrid<String>(8, 8);
```

1. Write an interface BoardDisplay that specifies a single void method display. (We use the interface so that the console display described below could easily be replaced by a graphic display.)
2. Write a class ConsoleBoardDisplay that implements BoardDisplay and has a reference to a Grid<String> that is bounded. The grid is set by the ConsoleBoardDisplay constructor. You may assume that the strings in the grid are single characters. The display method should print the board to the console, using a "." to represent empty cells and the appropriate letter for occupied cells.
3. Test your ConsoleBoardDisplay by writing a driver class that declares a board, as indicated above, and a ConsoleBoardDisplay. The driver should then place several single letter strings on the board and display the board.

Programming project I - OTHELLO

- a. Look up the rules for the game Othello.
- b. Write a class OthelloGame that has the following instance variables:

an 8x8 Grid<String>, board, a BoardDisplay, bdDisplay, two instances of OthelloPlayer, player1, player2 (OthelloPlayer is an abstract class, see below)

The OthelloGame should have a playGame method that has the players alternate moves until the game is over, then declares the winner. After a player is asked for its move, the game should first check the legality of the move. If it is not legal, the same player is again asked to move. Otherwise, the board is reconfigured according to the rules and the next player is asked to move.

- c. Use an abstract class OthelloPlayer that has instance variables to store its name and a reference to the board (set in its constructor and accessed with accessor methods). OthelloPlayer should have an abstract method move() that returns a Location.
- d. Write a class HumanOthelloPlayer that implements OthelloPlayer and has a move() method that prompts a human user to specify a move.
- e. Write a class StupidComputerPlayer that implements OthelloPlayer. The move() method returns a randomly selected legal move.
- f. Write a driver class to set up and run our Othello game.

