

| |
|--|
| Game |
| <pre>private Player guesser private Player mastermind private int[][] _board private int[][] _pegs private int _turn private int[] _correctAns public Game(int i) public void setFinalTwoPlayer() public int[][] getBoard() public int[][] getPegs() public boolean turn() public int getTurn() public int[] getFinal()</pre> |
| Mastermind |
| <pre>public static int numTwos() public static int numOnes() public static void printBoard(boolean fancy, gameGame) public static void play(int totalTurns, boolean isFancy)</pre> |
| Guesser |
| <pre>public static void printBoard(boolean fancy, gameGame) public static void play(int totalTurns, boolean isFancy)</pre> |
| TwoPlayer |
| <pre>public static void printBoard(boolean fancy, gameGame) public static void play(int totalTurns, boolean isFancy)</pre> |

Player (abstract)

** this has no abstract methods but it makes sense for it to be abstract because there is no generic player, there has to be a User or Computer

```
protected static int[] _currGuess
protected static int[] _finalAns
private int removeFromTempAns
protected int _currentTurn
protected boolean _masterOrGuesser
private int[][] _gameBoard
private int[][] _pegsBoard

public int[] setGuess(int[] guess)
public boolean check(int val, int[] collection)
public int[] givePegs() //will be used to respond with number of pegs
public int[] setFinal()
public void setGameBoard(int[][] a)
public void setPegsBoard(int[][] a)
public void setTurn()
public int[] getFinal()
```

User extends Player

*** this class seems useless but it is nice to have conceptually. The distinction between a User and Player isn't functional but rather helps us organize and distinguish between a user and computer.

```
public User() //constructor
```

Computer extends Player

```
public int[] makeAnswer()
public Computer(boolean mOrG) //constructor
```

generateAll

**this class only exists to generate all possible guesses for the algorithm

```
public static void main(String[] args)
```

| |
|-----------|
| makeGuess |
|-----------|

| |
|--|
| <pre>private static int[] all private ArrayList<ArrayList<Integer>> narrow public makeGuess() //constructor public boolean noMatch(ArrayList<integer> checking, int[] lastG, int times) public boolean noperfmatch(ArrayList<Integer> checking, int[] lastG, int perf) public int match(int index, ArrayList<Integer> checking, int[] lastG, int red, int white) public int check(int index, ArrayList<Integer> checking, int[] lastG, int[] pegs) public ArrayList<Integer> guess(int[] lastG, int[] pegs)</pre> |
|--|

| |
|--------|
| Prompt |
|--------|

| |
|---|
| <pre>private static String _Correct private static final String NUMS public static final String[] ITEMS_WINS public static final String[] COLORS public static final String[] ITEMS public static void reference() public static String promptWord(String question, boolean hidden) public static String promptWord(String question) public static String arrToStr(int[] arr, String seperator) public static String arrToStr(int[] arr) public static boolean isAllDigs(String s) public static int[] getGuess(String question, boolean hidden) public static int[] getGuess(String question) public static boolean isAllDigsPegs(String s) public static int[] getPegs(String question) public static int getChoice(String question, int n) public static int getChoice(string question)</pre> |
|---|

| |
|--|
| cs1/Keyboard (Slight modifications from Lewis and Loftus version in library) |
|--|

| |
|---|
| <pre>private static boolean printErrors private static int errorCount private static String current_token</pre> |
|---|

```
private static StringTokenizer
private static BufferedReader

public static int getErrorCount()
public static void resetErrorCount (int count)
public static boolean getPrintErrors()
public static void setPrintErrors (boolean flag)
private static void error (String str)
private static String getNextToken()
private static String getNextToken (boolean skip)
private static String getNextToken (boolean skip, boolean hide)
private static String getNextInputToken(boolean skip)
private static String getNextInputToken (boolean skip, boolean hide)
public static boolean endOfLine()
public static String readString()
public static String readWord()
public static String readWord(boolean hide)
public static boolean readBoolean()
public static char readChar()
public static int readInt()
public static long readLong()
public static float readFloat()
public static double readDouble()
```

cs1/Masker Implements Runnable (built in java class): Actively replaces system in/output with spaces when enabled by running in a while loop in it's own thread

```
private boolean isHide
```

```
public void run()
public void stop()
```