Main Class

All methods and most variables are non-static as there is no need for them to be static and everything is public throughout all the classes.

Instance Variables

- `public static final int WIDTH = 610;`
    - Sets the width of the window
- `public static final int HEIGHT = 900;`
    - Sets the height of the window
- `public static final int FPS = 60;`
    - Sets the frames per second at which the computer runs
- `private BufferedImage im;`
    - Variable that refers to the image of the red cup.
- `private BufferedImage imb;`
    - Variable that refers to the image of the blue cup.
- `private BufferedImage ball;`
    - Variable that refers to the image of the ball
- `private BufferedImage bar;`
    - Variable that refers to the image of the
- `private BufferedImage splash;`
    - Variable that refers to the image of the initial splash screen
- `private BufferedImage comwin;`
    - Variable that refers to the image of the splash screen if you lose
- `private BufferedImage youwin;`
    - Variable that refers to the image of the splash screen if you win
- `public static Rack user;`
    - Variable that keeps track of the users rack
- `public static Rack comp;`
    - Variable that keeps track of the computers rack
- `int speed;`
    - Int speed gets the number of cups left. And then based of that value will assign a value to a increases the speed of the bar.
- `int barX;`
    - Int that keeps track of the position of the shot bar
- `int cup;`

    - Int value that is assigned to each cup going from front to back left to right
- long startTime;
    - Keeps track of when the user started playing
- boolean sp;
    - Tells the line whether it needs to start moving for the user to take its turn.
- boolean hit;
    - Boolean that is set to true when the user has correctly timed the range necessary for a made shot.
- boolean miss;
    - Boolean that is set to true when the user/computer has missed the range necessary for a made shot.
- boolean testb;
    - Boolean in charge of handing the closing of the splash screen
- boolean move;
    - Helps the program keep track of if it's the users move.
- boolean right;
    - Boolean in charge of telling the line which direction to go in
- boolean uTurn;
    - Boolean that designates when it is the users turn to shoot
- boolean cupSelected;
    - Boolean in charge of recognizing when a cup is chosen. Set to true after receiving a number when given by the user.
- boolean shot;
    - Boolean that is set to false initially
- public boolean shooting;
- boolean cHit;
    - Boolean that is set to true when the user has correctly hit the range necessary for a made shot.
- boolean cTurn;
    - Boolean that designates when it is the computers turn to shoot
- boolean cWin;
    - Boolean that checks to see if the cups have all been hit for the user. If true an end game screen pops up.
- boolean uWin;

    - Boolean that checks to see if the cups have all been hit for the computer. If true an end game screen pops up.
- int sy;
    - Keeps track of the y position of the ball.
- int sx;a
    - Keeps track of the x position of the ball.
- int a;
    - Int that is assigned to x location of a certain cup in the rack letting sx know when it has reached the cup.
- int b;
    - Int that is assigned to y location of a certain cup in the rack letting sy know when it has reached the cup.
- int compDelay;
    - Creates a delay so that the user can see that the computer is taking its turn
- long endTime;
    - Helps keep track of how long it took the user to win or lose
- boolean alreadyP;
    - Prevents the user from selecting a cup that has already been eliminated
- boolean outBounds;
    - Prevents the program from selecting a cup that does not exist and returning out of bounds exception

Main Methods

public void run()

- Method that runs continuously in charge of repainting the graphics on the screen,running various methods for gameplay that allows gameplay these being shots by computer and user, the method that checks for the game being over, and other methods that require continuous updating at times.

public static void main(String[] args)

- Creates a rack for the user and the computer and initiates the graphics window named “CupPong” which is sent to the run method that allows the window to run.
- Also initializes other variables that require initializion.

public void paintComponent(Graphics g){

- Method that handles all the things that will be printed in the graphics window called by repaint within method run. Inside this method the images required for the game that were uploaded into java are loaded in and printed. The read in Graphics G created in this method includes the images on the splash screen, gameplay screen (background, racks(sets of cups), balls, bar and text instructions included). We also have several if statements that determine when certain images, screens, or cups are drawn based on the rules of the game.

public void keyPressed(KeyEvent e)

- Method that is a KeyListener Method added in the constructor in the main class that is always running that grabs the key pressed from the user and sends it to other methods that utilizes the users key inputs.

public void keyReleased(KeyEvent e)

- Method that is a KeyListener Method added in the constructor in the main class that is always running that grabs the key released from the user tells user in the terminal when they have released a key and which key they have released. Required for keyListener and helpful in debugging.

public void keyTyped(KeyEvent e)

- Method that is a KeyListener Method added in the constructor in the main class that is always running that grabs the key pressed from the user tells user in the terminal when they have typed a key and which key they have typed. Required for keyListener and helpful in debugging.

public void addNotify()

- Method that calls the addNotify() method in the KeyListener class. Necessary for the function of the keyListener in the program.

public static int[] locationUSER(int l)

- Method called by the portion of paintComponent that is responsible for drawing the user’s rack, that returns the x and y positions of where a cup should be within a rack for the user. x and y goes up to the paintComponent and prints the images of the cups based on the coordinates. Returns both the x and y values in an arraylist of integers.

public int[] locationCOMP(int l)

- Method called by the portion of paintComponent that is responsible for drawing the computer’s rack, that returns the x and y positions of where a cup should be within a rack for the user. x and y goes up to the paintComponent and prints the images of the cups based on the coordinates. Returns both the x and y values in an arraylist of integers.
-

public int speedSelect()

- Method called by method spee. Essentially speedSelect will find the number of cups left for the user and then based off this int named speed a value, int a will be assigned a number. The higher the speed the higher the a and a higher difficulty. The int a will then be returned back to spee.

public void spee()

- Method called by run handles all movement of the bar. First he speed of the bar will be calculated by calling speedSelect. Then if the boolean sp is true it will cause the bar to start moving. It also makes sure the bar bounces when it reaches the end instead of going off the screen.

```
public void select(keyEvent e)
```

- This method is called by the KeyPressed method and handles all the keyboard inputs by the user. It takes in the key pressed by the user (keyEvent e) based off of what that is it will carry out different tasks like setting testb to false to remove the splash screen, allow the user to select their cup, and calls the method spaced.

```
public void evolve()
```

- This method, was originally designed to have a ball shoot across the screen and eliminate the cup or miss depending on whether the user made their shot or not. However, it did not function properly as the ball moved too fast to be visible. Instead now, it ends the users turn when they are done.

```
public void actionPerformed(ActionEvent e)
```

- Method was originally designed to allow the user to click a button to get rid of the splash screen. However, the button was later removed in favor of click the spacebar as it made the game more intuitive.

```
public void barTest()
```

- The barTest method is called by the spaced method and analyzes whether the user stopped the bar in the proper location calculating whether they made or missed the shot.

```
public void compDelayer()
```

- Created a delay before the computer ran its turn so that the user would be able to see that the computer was taking its turn.

```
public void computer()
```

- The computer method lets the user take its turn. It generates a number for the computer to guess and then generates a random guess. If the two numbers are equal then the computer makes it. If not, the computer misses their shot.

```
 public void game()
```

- Keeps track of whether the game is going on and if it is over who won and who lost. Then changes booleans accordingly so that the program will display the proper screen.

Rack Class

- `Cup[] rac;`
    - Creates an array of object cup.
- `int cupCount;`
    - Keeps track of how many cups within the rack have been hit. This is used by the Main class to decide on the difficulty

The constructor for this class does not take anything in and creates a array of ten cups to serve as the rack and sets the cup count to 0 when it is first created.

```
public void makeDead(int i)
```

- Method that tells a cup to become dead if it has been hit and also increases the cup count by 1.
- Takes in int i so that it knows what cup to eliminate.

Cup Class

- `boolean alive;`
    - Keeps track of whether the cup is dead or alive
- `Int location;`
    - Keeps track of what position each cup is in