private String placeBet(int whichBet) {
    String result = "";
    if (whichBet == 0) {
        Set<String> choices = new TreeSet<String>();
        choices.add(Wheel.BLACK);
        choices.add(Wheel.RED);
        result = ConsoleReader.promptOneOf("Please bet", choices);
    } else if (whichBet == 1) {
        Set<String> choices = new TreeSet<String>();
        choices.add("even");
        choices.add("odd");
        result = ConsoleReader.promptOneOf("Please bet", choices);
    } else if (whichBet == 2) {
        //...
        System.out.println();
        return result;
    }
}
A Look at the Bet classes

```java
public HighLowBet() {
    super("High or Low");
    choices.add(HIGH);
    choices.add(LOW);
}

@Override
public boolean betIsWon(Wheel wheel) {
    if (wheel.onGreen()) {
        return false;
    }
    int wheelNumber = wheel.getNumber();
    return (wheelNumber > SEP & userChoice.equals(HIGH)) ||
           (wheelNumber <= SEP & userChoice.equals(LOW));
}
```

```java
public class ConsecutiveBet extends Bet {
    private int numConsecutive;

    public ConsecutiveBet(int numConsecutive, int odds) {
        super(numConsecutive + " in a row", odds);
        this.numConsecutive = numConsecutive;
    }

    public ConsecutiveBet(int odds) {
        this(1, odds);
    }

    public void placeBet() {
        userChoice = ";
        int start = Integer.parseInt(userChoice);
        return (start <= wheel.getNumber() && wheel.getNumber() <
                start + this.numConsecutive);
    }
}
```

Effect on Game Class

```java
private Bet[] myPossibleBets = {
    new RedBlackBet(),
    new OddEvenBet(),
    new HighLowBet(),
    new ConsecutiveBet(1, 35),
    new ConsecutiveBet(2, 17),
    new ConsecutiveBet(3, 11)
};
```

Discussion

- Benefits of the refactored hierarchy
- Drawbacks of the refactored hierarchy

Benefits of The Refactored Hierarchy

- Benefits of the refactored hierarchy
  - Where is the logic about the bets?
    - In the Bet classes
    - Game can manage the game, not be responsible for bets
  - Easier to add a new Bet

- Drawbacks of the refactored hierarchy
  - Adds more classes, hierarchy, abstraction

Oh, the places you have been!

- What Have You Learned This Semester?
Project Notes

- Project Analysis: Make sure you understand the others' design/code/parts
  - At least at a high level

Extensions Discussion

- What would you need to do to create random expressions?

PICASSO DEMO

Picasso Demo

- Let's see this baby in action!
  - Discuss any design issues/challenges you’ve met so far
    - Interesting discussions/conclusions
    - How you’d change if you were to do something similar later
# Picasso Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines of Code</td>
<td>3327</td>
</tr>
<tr>
<td>Methods</td>
<td>318</td>
</tr>
<tr>
<td>Classes</td>
<td>138</td>
</tr>
<tr>
<td>Packages</td>
<td>13</td>
</tr>
</tbody>
</table>