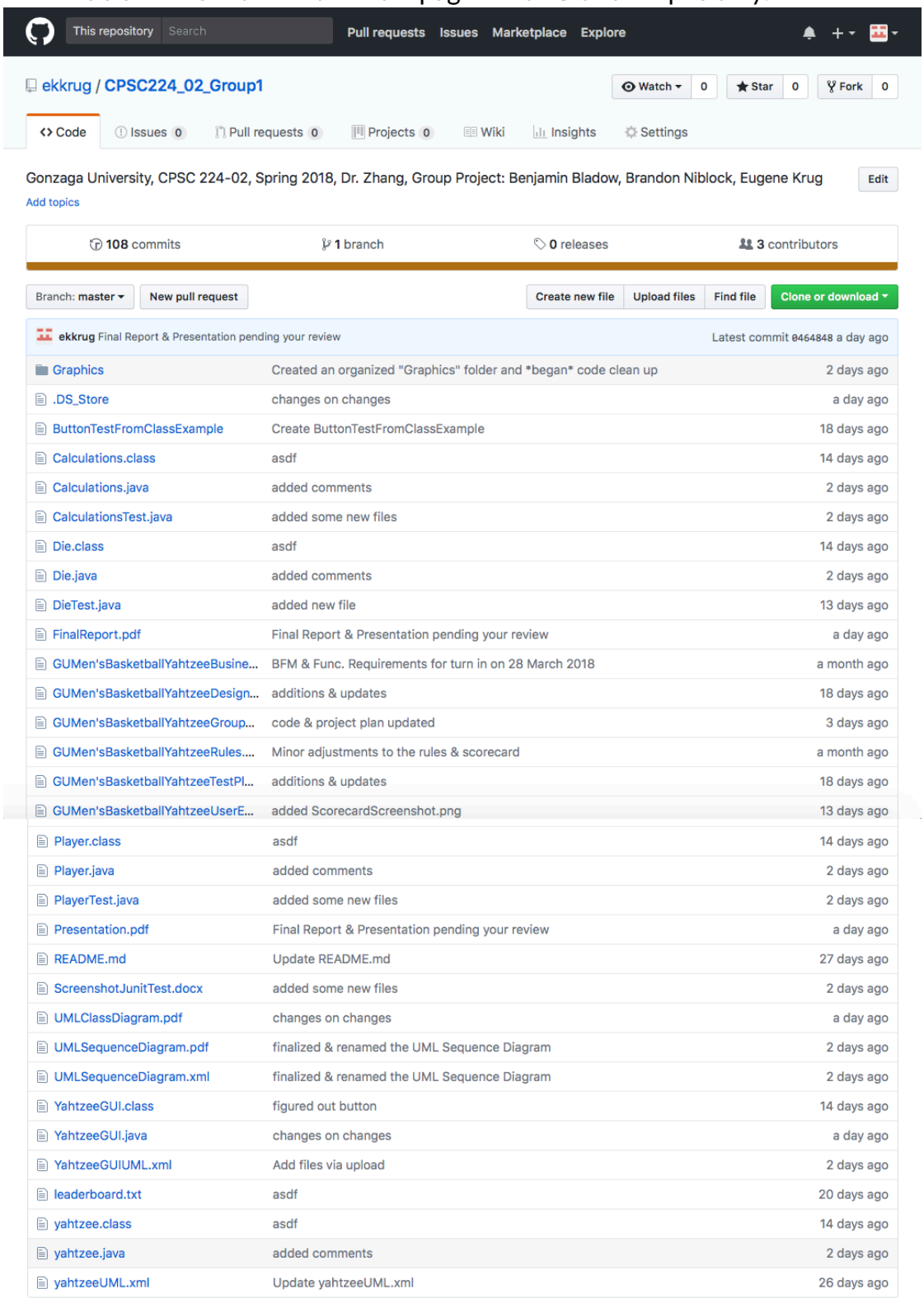
GitHub and JUnit Benjamin Bladow, Eugene Krug, Brandon Niblock

I. GitHub

Link: https://github.com/ekkrug/CPSC224_02_Group1

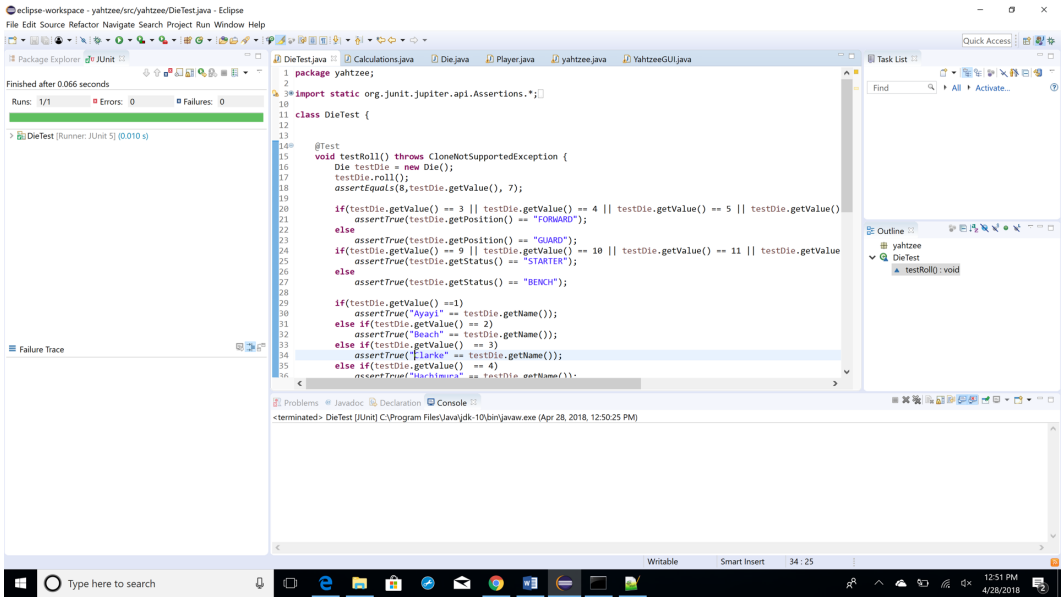
Below is a screenshot from the main page of our GitHub repository.



II. JUnit

A. DieTest.java

```
package yahtzee;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

class DieTest {

	 @Test
	 void test() {
		Die testDie = new Die();
		testDie.roll();
		assertEquals(8,testDie.getValue(), 7);

	if(testDie.getValue() == 3 || testDie.getValue() == 4 || testDie.getValue() == 5 ||
	 testDie.getValue() == 7 || testDie.getValue() == 13 || testDie.getValue() == 15)
	    assertTrue(testDie.getPosition() == "FORWARD");
	else
	    assertTrue(testDie.getPosition() == "GUARD");

	if(testDie.getValue() == 9 || testDie.getValue() == 10 || testDie.getValue() == 11 ||
	 testDie.getValue() == 13 || testDie.getValue() == 15)
	    assertTrue(testDie.getStatus() == "STARTER");
	else
	    assertTrue(testDie.getStatus() == "BENCH");

	     if(testDie.getValue() ==1)
	       assertTrue("Joel Ayayi" == testDie.getName());
	     else if(testDie.getValue() == 2)
	       assertTrue("Jack Beach" == testDie.getName());
	     else if(testDie.getValue()  == 3)
	       assertTrue("Brandon Clarke" == testDie.getName());
	     else if(testDie.getValue()  == 4)
	       assertTrue("Rui Hachimura" == testDie.getName());
	     else if(testDie.getValue()  == 5)
	       assertTrue("Jeremy Jones" == testDie.getName());
	     else if(testDie.getValue()  == 6)
	       assertTrue("Corey Kispert" == testDie.getName());
	     else if(testDie.getValue()  == 7)
	       assertTrue("Jacob Larson" == testDie.getName());
	     else if(testDie.getValue()  == 8)
	       assertTrue("Alex Martin" == testDie.getName());
	     else if(testDie.getValue()  == 9)
	       assertTrue("Silas Melson" == testDie.getName());
	     else if(testDie.getValue()  == 10)
	       assertTrue("Zach Norvell Jr." == testDie.getName());
	     else if(testDie.getValue()  == 11)
	       assertTrue("Josh Perkins" == testDie.getName());
	     else if(testDie.getValue() == 12)
	       assertTrue("Brian Pete" == testDie.getName());
	     else if(testDie.getValue() == 13)
	       assertTrue("Killian Tillie" == testDie.getName());
	     else if(testDie.getValue() == 14)
	       assertTrue("Jesse Wade" == testDie.getName());
	     else if(testDie.getValue() == 15)
	       assertTrue("Jonathan Williams" == testDie.getName());

	 }

}
```



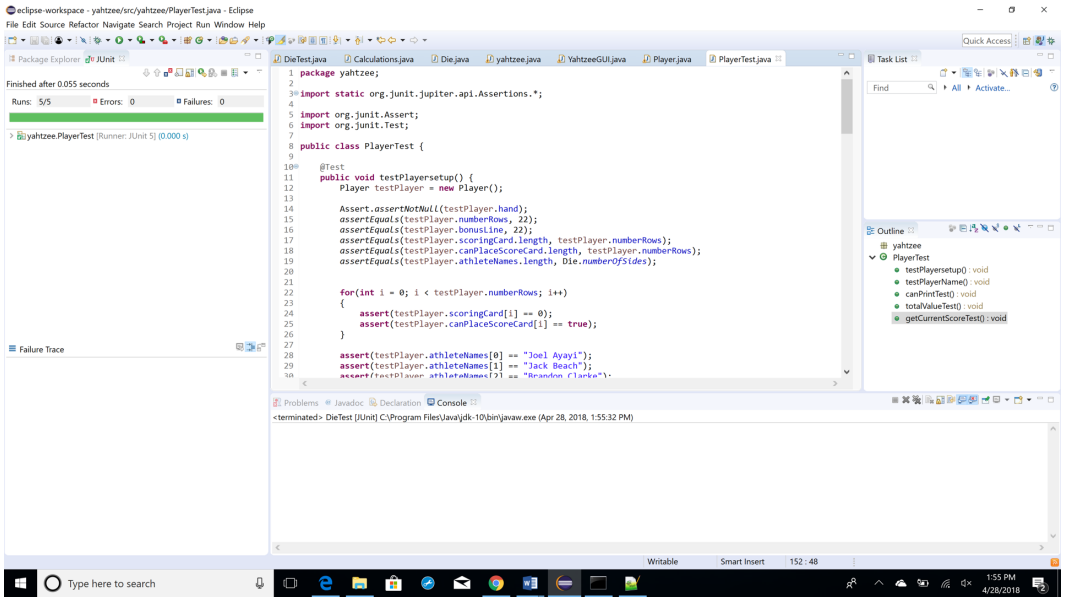
B. PlayerTest.java

(Code is on three pages.)

```
package yahtzee;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.Assert;
import org.junit.Test;

public class PlayerTest {

	@Test
	public void testPlayersetup() {
		Player testPlayer = new Player();

		Assert.assertNotNull(testPlayer.hand);
		assertEquals(testPlayer.numberRows, 22);
		assertEquals(testPlayer.bonusLine, 22);
		assertEquals(testPlayer.scoringCard.length, testPlayer.numberRows);
		assertEquals(testPlayer.canPlaceScoreCard.length, testPlayer.numberRows);
		assertEquals(testPlayer.athleteNames.length, Die.numberOfSides);


		for(int i = 0; i < testPlayer.numberRows; i++)
		{
			assert(testPlayer.scoringCard[i] == 0);
			assert(testPlayer.canPlaceScoreCard[i] == true);
		}

	    assert(testPlayer.athleteNames[0] == "Joel Ayayi");
	    assert(testPlayer.athleteNames[1] == "Jack Beach");
	    assert(testPlayer.athleteNames[2] == "Brandon Clarke");
	    assert(testPlayer.athleteNames[3] == "Rui Hachimura");
	    assert(testPlayer.athleteNames[4] == "Jeremy Jones");
	    assert(testPlayer.athleteNames[5] == "Corey Kispert");
	    assert(testPlayer.athleteNames[6] == "Jacob Larson");
	    assert(testPlayer.athleteNames[7] == "Alex Martin");
	    assert(testPlayer.athleteNames[8] == "Silas Melson");
	    assert(testPlayer.athleteNames[9] == "Zach Norvell Jr.");
	    assert(testPlayer.athleteNames[10] == "Josh Perkins");
	    assert(testPlayer.athleteNames[11] == "Brian Pete");
	    assert(testPlayer.athleteNames[12] == "Killian Tillie");
```

```
        assert(testPlayer.athleteNames[13] == "Jesse Wade");
        assert(testPlayer.athleteNames[14] == "Jonathan Williams");
        }
        @Test
        public void testPlayerName() {
              Player testPlayer = new Player();
              testPlayer.setName("bob");
              assert("bob" == testPlayer.getName());
        }

        @Test
        public void canPrintTest() {
              Player testPlayer = new Player();

              assert(testPlayer.canPrint(5) == true);

              testPlayer.scoringCard[5] = 50;
              testPlayer.canPlaceScoreCard[5] = false;

              assert(testPlayer.canPrint(5) == false);
        }

        @Test
        public void totalValueTest() {
              Player testPlayer = new Player();

              assert(testPlayer.totalValue() == 0);
              testPlayer.scoringCard[0] = 10;
              testPlayer.canPlaceScoreCard[0] = false;
              assert(testPlayer.totalValue() == 10);

              testPlayer.scoringCard[1] = 10;
              testPlayer.canPlaceScoreCard[1] = false;
              assert(testPlayer.totalValue() == 20);

              testPlayer.scoringCard[2] = 10;
              testPlayer.canPlaceScoreCard[2] = false;
              assert(testPlayer.totalValue() == 30);

              testPlayer.scoringCard[3] = 10;
              testPlayer.canPlaceScoreCard[3] = false;
              assert(testPlayer.totalValue() == 40);

              testPlayer.scoringCard[4] = 10;
              testPlayer.canPlaceScoreCard[4] = false;
              assert(testPlayer.totalValue() == 50);

              testPlayer.scoringCard[5] = 10;
              testPlayer.canPlaceScoreCard[5] = false;
              assert(testPlayer.totalValue() == 60);

              testPlayer.scoringCard[6] = 10;
              testPlayer.canPlaceScoreCard[6] = false;
              assert(testPlayer.totalValue() == 70);

              testPlayer.scoringCard[7] = 10;
              testPlayer.canPlaceScoreCard[7] = false;
              assert(testPlayer.totalValue() == 80);

              testPlayer.scoringCard[8] = 10;
              testPlayer.canPlaceScoreCard[8] = false;
              assert(testPlayer.totalValue() == 90);

              testPlayer.scoringCard[9] = 10;
              testPlayer.canPlaceScoreCard[9] = false;
              assert(testPlayer.sumBonus() == 35);
              assert(testPlayer.totalValue() == 135);
        }
```

```
	@Test
	public void getCurrentScoreTest() {
		Player testPlayer = new Player();

		assert(testPlayer.getCurrentScore() == 0);
		testPlayer.scoringCard[0] = 10;
		testPlayer.canPlaceScoreCard[0] = false;
		assert(testPlayer.getCurrentScore() == 10);

		testPlayer.scoringCard[1] = 10;
		testPlayer.canPlaceScoreCard[1] = false;
		assert(testPlayer.getCurrentScore() == 20);

		testPlayer.scoringCard[2] = 10;
		testPlayer.canPlaceScoreCard[2] = false;
		assert(testPlayer.getCurrentScore() == 30);

		testPlayer.scoringCard[3] = 10;
		testPlayer.canPlaceScoreCard[3] = false;
		assert(testPlayer.getCurrentScore() == 40);

		testPlayer.scoringCard[4] = 10;
		testPlayer.canPlaceScoreCard[4] = false;
		assert(testPlayer.getCurrentScore() == 50);

		testPlayer.scoringCard[5] = 10;
		testPlayer.canPlaceScoreCard[5] = false;
		assert(testPlayer.getCurrentScore() == 60);

		testPlayer.scoringCard[6] = 10;
		testPlayer.canPlaceScoreCard[6] = false;
		assert(testPlayer.getCurrentScore() == 70);

		testPlayer.scoringCard[7] = 10;
		testPlayer.canPlaceScoreCard[7] = false;
		assert(testPlayer.getCurrentScore() == 80);

		testPlayer.scoringCard[8] = 10;
		testPlayer.canPlaceScoreCard[8] = false;
		assert(testPlayer.getCurrentScore() == 90);

		testPlayer.scoringCard[9] = 10;
		testPlayer.canPlaceScoreCard[9] = false;
		//Bonus not included until the very end
		assert(testPlayer.getCurrentScore() == 100);
	}
}
```



C. CalculationsTest.java

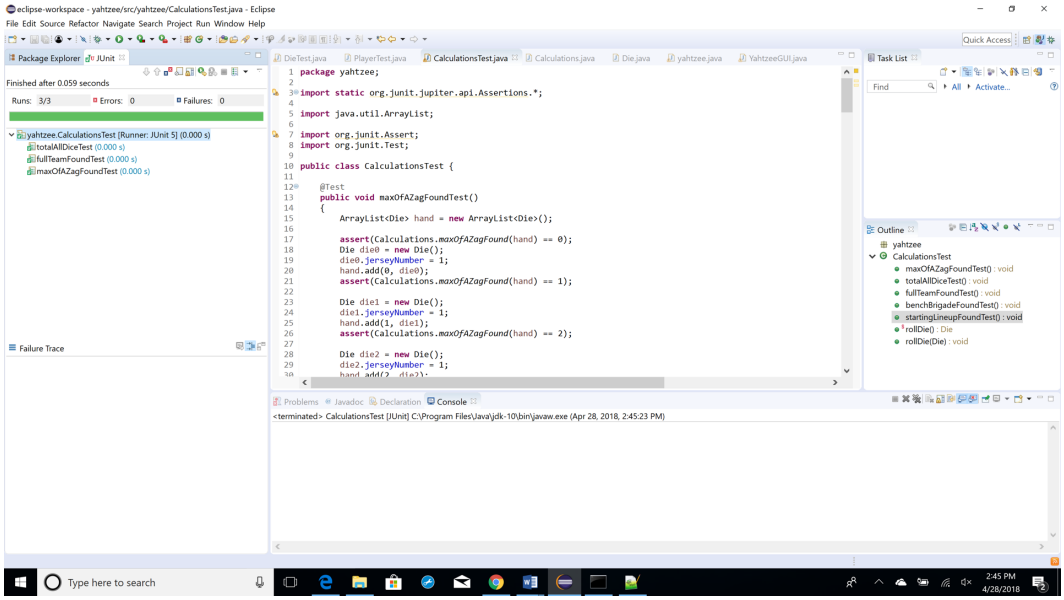
(Code is on four pages.)

```
package yahtzee;

import static org.junit.jupiter.api.Assertions.*;

import java.util.ArrayList;

import org.junit.Assert;
import org.junit.Test;

public class CalculationsTest {

	@Test
	public void maxOfAZagFoundTest()
	{
		ArrayList<Die> hand = new ArrayList<Die>();

		assert(Calculations.maxOfAZagFound(hand) == 0);
		Die die0 = new Die();
		die0.jerseyNumber = 1;
		hand.add(0, die0);
		assert(Calculations.maxOfAZagFound(hand) == 1);

		Die die1 = new Die();
		die1.jerseyNumber = 1;
		hand.add(1, die1);
		assert(Calculations.maxOfAZagFound(hand) == 2);

		Die die2 = new Die();
		die2.jerseyNumber = 1;
		hand.add(2, die2);
		assert(Calculations.maxOfAZagFound(hand) == 3);

		Die die3 = new Die();
		die3.jerseyNumber = 1;
		hand.add(3, die3);
		assert(Calculations.maxOfAZagFound(hand) == 4);
```

```
		Die die4 = new Die();
		die4.jerseyNumber = 1;
		hand.add(4, die4);
		assert(Calculations.maxOfAZagFound(hand) == 5);
	}

	@Test
	public void totalAllDiceTest()
	{
		ArrayList<Die> hand = new ArrayList<Die>();

		assert(Calculations.totalAllDice(hand) == 0);
		Die die0 = new Die();
		die0.jerseyNumber = 1;
		hand.add(0, die0);
		assert(Calculations.totalAllDice(hand) == 1);

		Die die1 = new Die();
		die1.jerseyNumber = 5;
		hand.add(1, die1);
		assert(Calculations.totalAllDice(hand) == 6);

		Die die2 = new Die();
		die2.jerseyNumber = 10;
		hand.add(2, die2);
		assert(Calculations.totalAllDice(hand) == 16);

		Die die3 = new Die();
		die3.jerseyNumber = 14;
		hand.add(3, die3);
		assert(Calculations.totalAllDice(hand) == 30);

		Die die4 = new Die();
		die4.jerseyNumber = 1;
		hand.add(4, die4);
		assert(Calculations.totalAllDice(hand) == 31);
	}

	@Test
	public void fullTeamFoundTest()
	{
		ArrayList<Die> hand = new ArrayList<Die>();

		assert(Calculations.fullTeamFound(hand) == false);
		Die die0 = new Die();
		die0.jerseyNumber = 3;
		rollDie(die0);
		hand.add(0, die0);
		assert(Calculations.fullTeamFound(hand) == false);

		Die die1 = new Die();
		die1.jerseyNumber = 4;
		rollDie(die1);
		hand.add(1, die1);
		assert(Calculations.fullTeamFound(hand) == false);

		Die die2 = new Die();
		die2.jerseyNumber = 10;
		rollDie(die2);
		hand.add(2, die2);
		assert(Calculations.fullTeamFound(hand) == false);

		Die die3 = new Die();
		die3.jerseyNumber = 9;
		rollDie(die3);
		hand.add(3, die3);
		assert(Calculations.fullTeamFound(hand) == false);

		Die die4 = new Die();
		die4.jerseyNumber = 5;
		rollDie(die4);
		hand.add(4, die4);
		assert(Calculations.fullTeamFound(hand) == false);
```

```
		die4.jerseyNumber = 11;
		rollDie(die4);
		hand.set(4, die4);
		assert(Calculations.fullTeamFound(hand) == true);
	}

	public void benchBrigadeFoundTest()
	{
		ArrayList<Die> hand = new ArrayList<Die>();

		assert(Calculations.benchBrigadeFound(hand) == false);
		Die die0 = new Die();
		die0.jerseyNumber = 1;
		rollDie(die0);
		hand.add(0, die0);
		assert(Calculations.benchBrigadeFound(hand) == false);

		Die die1 = new Die();
		die1.jerseyNumber = 2;
		rollDie(die1);
		hand.add(1, die1);
		assert(Calculations.benchBrigadeFound(hand) == false);

		Die die2 = new Die();
		die2.jerseyNumber = 3;
		rollDie(die2);
		hand.add(2, die2);
		assert(Calculations.benchBrigadeFound(hand) == false);

		Die die3 = new Die();
		die3.jerseyNumber = 4;
		rollDie(die3);
		hand.add(3, die3);
		assert(Calculations.benchBrigadeFound(hand) == false);

		Die die4 = new Die();
		die4.jerseyNumber = 10;
		rollDie(die4);
		hand.add(4, die4);
		assert(Calculations.benchBrigadeFound(hand) == false);

		die4.jerseyNumber = 5;
		rollDie(die4);
		hand.set(4, die4);
		assert(Calculations.benchBrigadeFound(hand) == true);
	}

	public void startingLineupFoundTest()
	{
		ArrayList<Die> hand = new ArrayList<Die>();

		assert(Calculations.benchBrigadeFound(hand) == false);
		Die die0 = new Die();
		die0.jerseyNumber = 9;
		rollDie(die0);
		hand.add(0, die0);
		assert(Calculations.startingLineupFound(hand) == false);

		Die die1 = new Die();
		die1.jerseyNumber = 10;
		rollDie(die1);
		hand.add(1, die1);
		assert(Calculations.startingLineupFound(hand) == false);

		Die die2 = new Die();
		die2.jerseyNumber = 11;
		rollDie(die2);
		hand.add(2, die2);
		assert(Calculations.startingLineupFound(hand) == false);
```

```
                Die die3 = new Die();
                die3.jerseyNumber = 13;
                rollDie(die3);
                hand.add(3, die3);
                assert(Calculations.startingLineupFound(hand) == false);

                Die die4 = new Die();
                die4.jerseyNumber = 8;
                rollDie(die4);
                hand.add(4, die4);
                assert(Calculations.startingLineupFound(hand) == false);

                die4.jerseyNumber = 15;
                rollDie(die4);
                hand.set(4, die4);
                assert(Calculations.startingLineupFound(hand) == true);
        }

        public static Die rollDie()
        {
                Die newDie = new Die();
                newDie.roll();
                return newDie;
        }

        public void rollDie(Die die){
       if(die.jerseyNumber == 3 || die.jerseyNumber == 4 || die.jerseyNumber == 5 ||
        die.jerseyNumber == 7 || die.jerseyNumber == 13 || die.jerseyNumber == 15)
                die.position = "FORWARD";
       else
                die.position = "GUARD";

       if(die.jerseyNumber == 9 || die.jerseyNumber == 10 || die.jerseyNumber == 11 ||
        die.jerseyNumber == 13 || die.jerseyNumber == 15)
                die.status = "STARTER";
       else
                die.status = "BENCH";

       if(die.jerseyNumber == 1)
        die.name = "Ayayi";
       else if(die.jerseyNumber == 2)
        die.name = "Beach";
       else if(die.jerseyNumber == 3)
        die.name = "Clarke";
       else if(die.jerseyNumber == 4)
        die.name = "Hachimura";
       else if(die.jerseyNumber == 5)
        die.name = "Jones";
       else if(die.jerseyNumber == 6)
        die.name = "Kispert";
       else if(die.jerseyNumber == 7)
        die.name = "Larsen";
       else if(die.jerseyNumber == 8)
        die.name = "Martin";
       else if(die.jerseyNumber == 9)
        die.name = "Melson";
       else if(die.jerseyNumber == 10)
        die.name = "Norvell";
       else if(die.jerseyNumber == 11)
        die.name = "Perkins";
       else if(die.jerseyNumber == 12)
        die.name = "Pete";
       else if(die.jerseyNumber == 13)
        die.name = "Tillie";
       else if(die.jerseyNumber == 14)
        die.name = "Wade";
       else
        die.name = "Williams";
    }

}
```


eclipse-workspace - yahtzee/src/yahtzee/CalculationsTest.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
JUnit
Finished after 0.059 seconds
Runs: 3/3
Errors: 0
Failures: 0
yahtzee.CalculationsTest [Runner: JUnit 5] (0.000 s)
totalAllDiceTest (0.000 s)
fullTeamFoundTest (0.000 s)
maxOfAZagFoundTest (0.000 s)
Failure Trace
DieTest.java
PlayerTest.java
CalculationsTest.java
Calculations.java
Die.java
yahtzee.java
YahtzeeGUI.java
package yahtzee;
import static org.junit.jupiter.api.Assertions.*;
import java.util.ArrayList;
import org.junit.Assert;
import org.junit.Test;
public class CalculationsTest {
@Test
public void maxOfAZagFoundTest()
{
ArrayList<Die> hand = new ArrayList<Die>();
assert(Calculations.maxOfAZagFound(hand) == 0);
Die die0 = new Die();
die0.jerseyNumber = 1;
hand.add(0, die0);
assert(Calculations.maxOfAZagFound(hand) == 1);
Die die1 = new Die();
die1.jerseyNumber = 1;
hand.add(1, die1);
assert(Calculations.maxOfAZagFound(hand) == 2);
Die die2 = new Die();
die2.jerseyNumber = 1;
Task List
Find
All
Activate...
Outline
yahtzee
CalculationsTest
maxOfAZagFoundTest() : void
totalAllDiceTest() : void
fullTeamFoundTest() : void
benchBrigadeFoundTest() : void
startingLineupFoundTest() : void
rollDie() : Die
rollDie(Die) : void
Problems
Javadoc
Declaration
Console
<terminated> CalculationsTest [JUnit] C:\Program Files\Java\jdk-10\bin\javaw.exe (Apr 28, 2018, 2:45:23 PM)
Type here to search
2:45 PM
4/28/2018