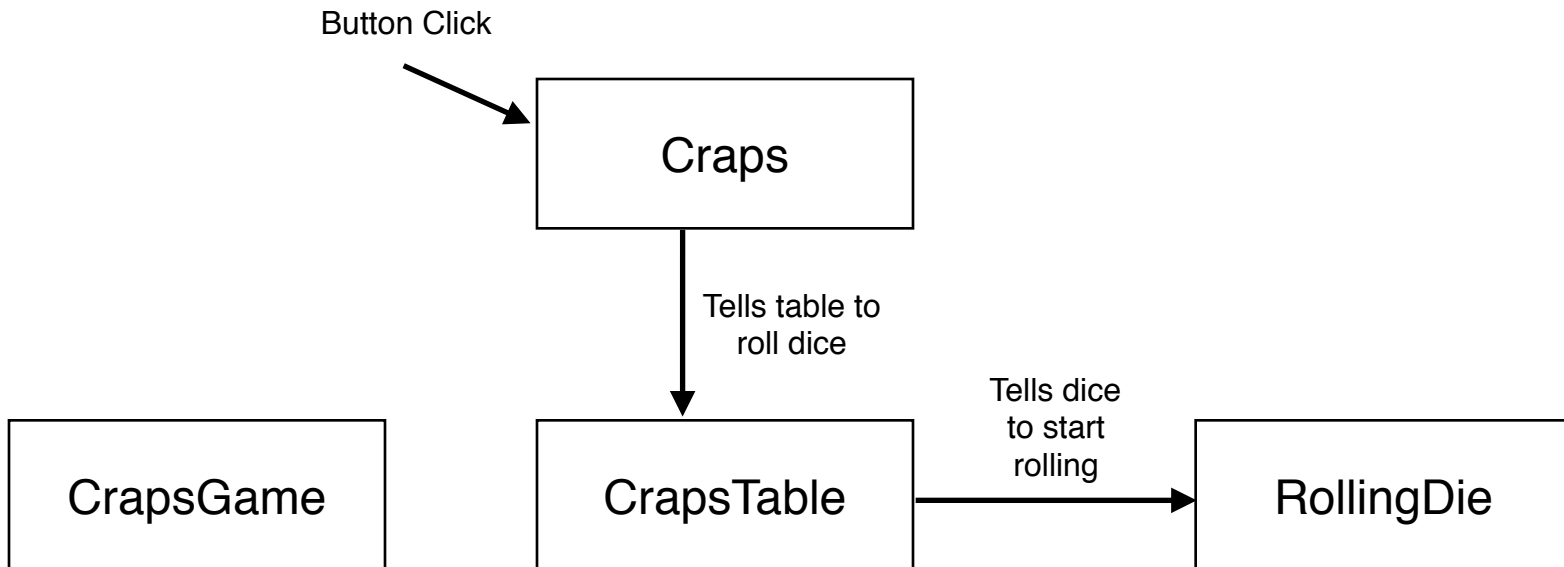
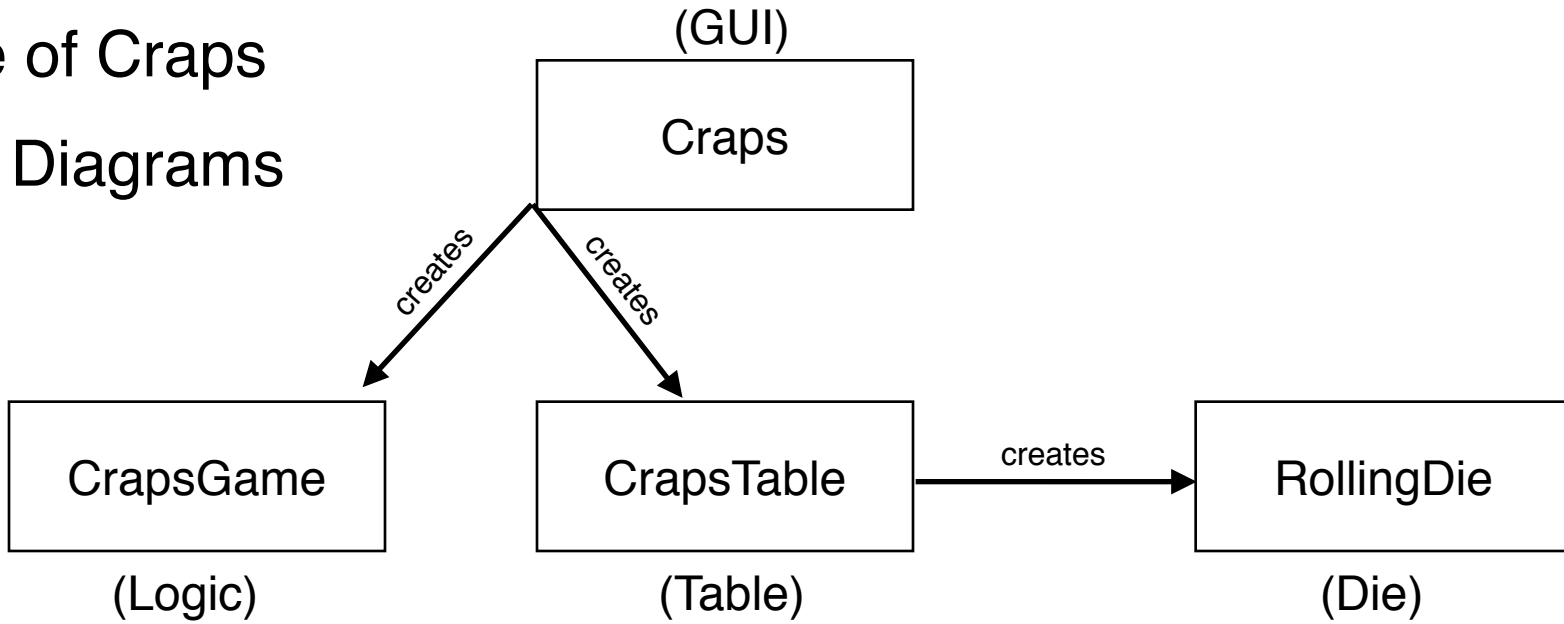
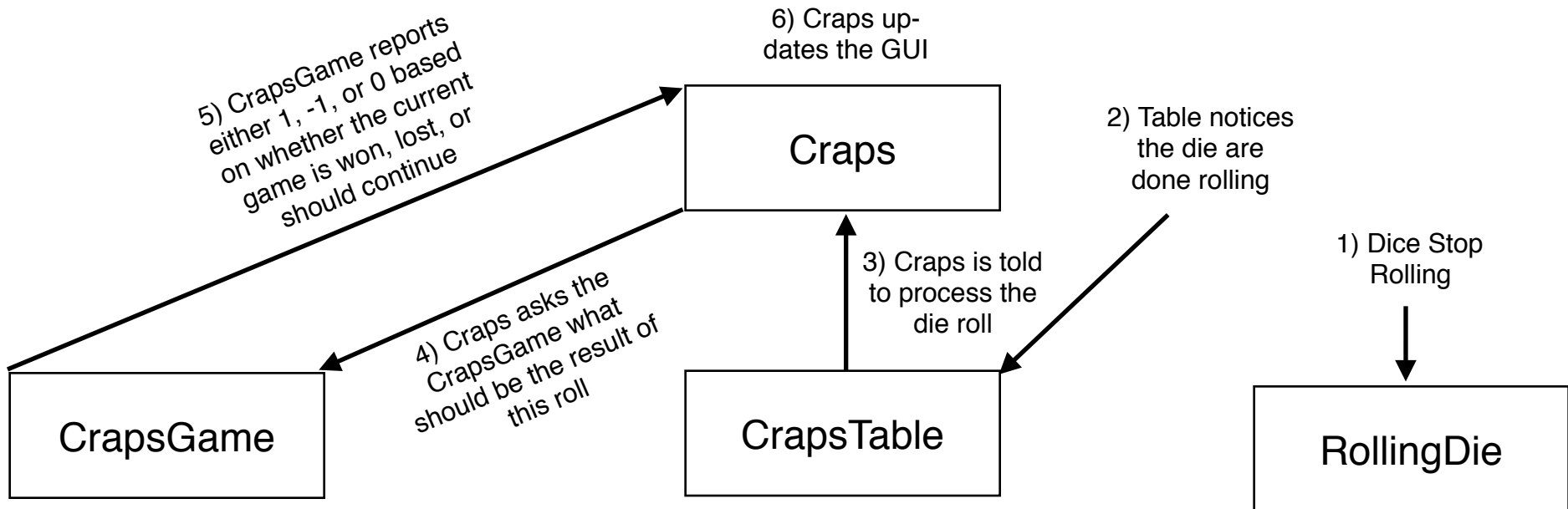


Game of Craps

Class Diagrams





Craps

Complete!

- Program starts in init() method
- Creates the GUI for the program
- Creates the CrapsGame and CrapsTable
- When a button is clicked, tells the table to roll the dice
- Is notified when the dice are done rolling and decides what to do with the score

```
//button has been pressed
public void actionPerformed(ActionEvent e)
{
    if (!table.diceAreRolling())
        table.rollDice();
}
```

```
//time to decide what to do after a roll
public void processRoll(int pts)
{
    int result;

    result = game.nextRoll(pts);

    if (result > 0)
    {
        //....
    }
}
```

CrapsTable

- Represents the Green Table that the die roll around in
- Creates the two RollingDie

Complete!

- When told to roll the die, it tells each of the RollingDie to roll
- Continually checks to see if the die are done rolling, in which case it tells Craps to process the roll

```
//makes sure the die are rolling
//when done, tells Craps to process
//the roll
public void actionPerformed(ActionEvent e)
{
    if (diceAreRolling()) {
        if (!clock.isRunning())
            clock.restart();

        if (die1.isRolling())
            die1.avoidCollision(die2);
        else if (die2.isRolling())
            die2.avoidCollision(die1);
    }
    else
    {
        clock.stop();
        craps.processRoll(die1.getPoints() + die2.getPoints());
        //...
    }
}
```

```
//tells both of the dice to start rolling
//also starts a timer to keep the dice rolling
public void rollDice()
{
    die1.roll(0, getWidth(), 0, getHeight());
    die2.roll(0, getWidth(), 0, getHeight());
    clock.start();
}
```


RollingDie

- Represents one die
- Has a face value between 1 and 6

Incomplete!

- Moves and bounces around the CrapsTable it is created within

```
//a random number between 1
and 6 (inclusive)
public int randomFaceValue()
{
    return -1;
}
```

	x1	x2	x3
y1			
y2			
y3			

```
//Finish this method that draws the dots on the dice
private void drawDots(Graphics g, int x, int y, int points)
{
    int x1, x2, x3, y1, y2, y3, dotSize;
    g.setColor(Color.white);

    //code that determines where to put the circles
    //for drawing the die face (there are 9 possible
    //locations for the white dots)
    dotSize = dieSize / 4;
    int step = dieSize / 8;
    x1 = x + step - 1;
    x2 = x + 3*step;
    //...

    if(points == 1)
        g.fillOval(x2, y2, dotSize, dotSize);
    //finish the other cases!
    else
        g.drawString(points + "", x2, y2);
}
```

CrapsGame

- Represents the logic of the game
- Stores the “point value”

Incomplete!

- Is asked to process a roll of the dice and to decide the result based on the state of the game
- Also must keep the value of point correct after a roll

```
public int nextRoll(int sumOfDie)
{
    int result; //needs to be assigned some value...
    // calculate result AND set point appropriately
    // Do not use other return statements

    result = -1; //you should change this

    //don't change below here!
    return result;
}
```