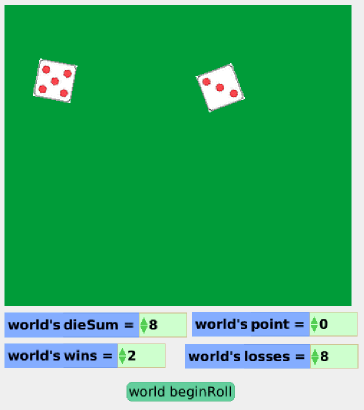
**Challenge**:

Create a function game of [Craps](https://en.wikipedia.org/wiki/Craps) (without betting).



**Things you’ll need to know**:

* How to animate the image of an object using holders (for the dice)
* The rules of the game of Craps and how to use test statements
* How to use variables
* Starting and stopping scripts

**Things to think about**:

* How can you tell whether a roll is a first roll (in which case 7 wins) or a subsequent roll (in which case 7 loses)
* How can you make rolling dice that bounce off each other and slow down?

**How to to it**:

* See below!

**Extensions**:

* Automate the rolling so we can see what happens after one million rolls. How often do you win? How often do you win on a first roll versus a second roll? Is the game fair?

**Craps Lessons**

**Part A: Random Dice Rolls**

Note: there are a LOT of ways to do this, some using several more sub scripts and variables.

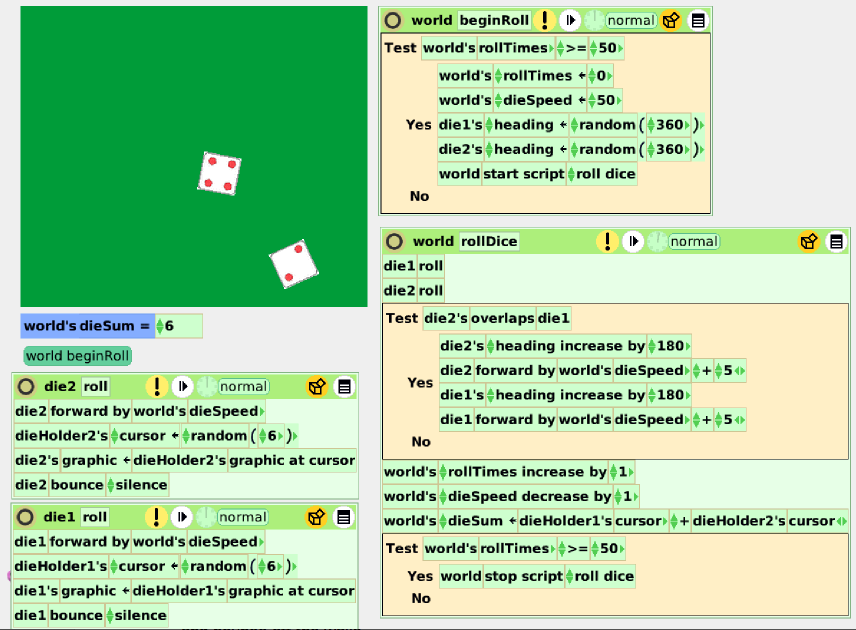
1. Create a Holder object and add six image objects to the holder that represent the 6 faces of a regular die. Name the Holder **dieHolder1**.
2. Draw a blob object (it doesn’t really matter what it looks like) and name it **die1.**
3. Add a variable to World called **rollTimes**. It should be a number that is initially 0.
4. Create a script for **die1** named **roll**. Have **dieHolder1’s** cursor be set to a random number between 1 and 6. Set **die1**’s graphic to be the same as the graphic at the current cursor in **dieHolder1**.
5. Create a script for the World named **rollDice**. It should tell **die1** to roll. It should also increase **rollTimes** by 1. It should also see test to see if the World’s **rollTimes** is 50 or greater. If it is, the **rollDice** script should be stopped.
6. Create a script for the World named **beginRoll**. It should set the **rollTimes** to be 0. It should also tell the script **rollDice** to start.
7. Make a duplicate of both **dieHolder1** and **die1**. This will give you a second die named **die2**.
8. Now edit the World’s **rollDice** script so that it starts **die2**’s roll script.
9. Create a button for your **beginRoll** script. You can hide your three scripts and the two holders.

**Bonus #1**

1. Create a Playfield and put the dice inside. For each roll scripts, have the dice move forward 50 and bounce off the walls.
2. In the World’s rollDice, randomize the headings of both of the dice.
3. For even more fun, make a variable for the dice speed. Make how far the dice moves be dependent on this variable. As the dice roll, decrement this value.
4. CHALLENGE: Can you make it so that the dice can’t stop on top of each other?
5. CHALLENGE 2: Can you make it so that you can’t start rolling again until the dice have stopped?

**Part B: Die Sum**

1. Add a number variable to the World called **dieSum**.
2. Drag a monitor for **dieSum** to the World area so you can see the value change.
3. In the World’s **rollDice** script, set the value of **dieSum** to be the sum of **holder1**’s cursor and **holder2**’s cursor [alternatively, you could have added another variable to each die to keep track of the value]



**Part C: Implementing Game Logic**

1. Create a World variable called **point**. It should initially be 0. Drag a monitor to the World.
2. Create two more World variables called **wins** and **losses**. They should initially be 0. Drag monitors to the World
3. Create two World’s scripts named **win** and **lose**.
4. In the **win** script, increase the **wins** variable by 1 and set the **point** variable to 0.
5. In the **lose** script, increase the **losses** variable by 1 and set the **point** variable to 0.
6. Create a World script called **processRoll**.
7. Have **processRoll** either call the **win** or **lose** script under the following conditions. If the point is 0, you lose if you roll a 2, 3, or 12, you win if you roll a 7 or 11, and you set point to match the dieSum. If the point is not 0, you lose if you roll a 7, you win if your roll matches the point, and nothing happens otherwise.
8. Edit the World’s **rollDice** script. Make it call **processRoll** when the **rollDice** script is stopped.

