CS 101
Randomness

Lecture 21
Randomness

• In most programming languages (Processing included) there are ways to generate random numbers
Randomness

- In most programming languages (Processing included) there are ways to generate random numbers
- When writing programs, when do you think randomness might come in handy?
Randomness

- In most programming languages (Processing included) there are ways to generate random numbers
- *When writing programs, when do you think randomness might come in handy?*
Randomness

- In most programming languages (Processing included) there are ways to generate random numbers
- When writing programs, when do you think randomness might come in handy?
Randomness

- In most programming languages (Processing included) there are ways to generate random numbers
- When writing programs, when do you think randomness might come in handy?
  - Simulations (Physical, Medical, Population, etc)
Randomness

• In most programming languages (Processing included) there are ways to generate random numbers
• When writing programs, when do you think randomness might come in handy?
  ○ Simulations (Physical, Medical, Population, etc)
  ○ Games (Artificial Intelligence, Variety)
Randomness

- In most programming languages (Processing included) there are ways to generate random numbers
- When writing programs, when do you think randomness might come in handy?
  - Simulations (Physical, Medical, Population, etc)
  - Games (Artificial Intelligence, Variety)
  - Security (Passwords, Cryptography)
Randomness

- But . . . how can a computer generate a random number?
- Computers only do exactly what human programmers tell them to do
- Do they know what randomness is?
Randomness

- How can a computer generate something random?
- Don’t computers just do what humans tell them to do?
Randomness

- Processing simulates randomness with *Pseudo-Random Number Generation (PRNG)*
- Essentially, **PRNGs** are algorithms that use mathematical formulae or simply precalculated tables to produce sequences of numbers that appear random [random.org]
- Use simple math formulas to go from the current number to the next number
Randomness

- Complete this code
- The `setXToRandomNumber` function should replace `x` with some random number
- How?

```java
int x = 20;

void setup () {
    for (int i = 0 ; i < 100; i ++) {
        setXToRandomNumber();
        println("x is: " + x);
    }
}

void setXToRandomNumber() {
    // What could go here?
}
```
Randomness

- Complete this code
- The `setXToRandomNumber` function should replace `x` with some random number
- How?

```java
int x = 20;

void setup () {
    for (int i = 0 ; i < 100; i ++) {
        setXToRandomNumber();
        println("x is: " + x);
    }
}

void setXToRandomNumber() {
    x += 7;
}
```
Randomness

- Complete this code
- The `setXToRandomNumber` function should replace `x` with some random number
- How?

```java
int x = 20;

void setup () {
  for (int i = 0 ; i < 100; i ++) {
    setXToRandomNumber();
    println("x is: " + x);
  }
}

void setXToRandomNumber() {
  x = (x*17) + 7 / 100;
}
```
Randomness

- Complete this code
- The `setXToRandomNumber` function should replace `x` with some random number
- What if we want to restrict the range of numbers to be between 0 and 100?

```java
int x = 20;

void setup () {
    for (int i = 0 ; i < 100; i ++) {
        setXToRandomNumber();
        println("x is: " + x);
    }
}

void setXToRandomNumber() {
    x = (x*17) + 7 / 100;
}
```
Modulus

- The modulus operator produces the remainder of division
- For example:
  4 % 2 = 0
  5 % 2 = 1
  6 % 2 = 0
  7 % 2 = 1
  8 % 2 = 0
Modulus

- The modulus operator produces the remainder of division
- For example:
  
  \[
  \begin{align*}
  4 \% 3 &= 1 \\
  5 \% 3 &= 2 \\
  6 \% 3 &= 0 \\
  7 \% 3 &= 1 \\
  8 \% 3 &= 2 \\
  \end{align*}
  \]
Modulus

- The modulus operator produces the remainder of division
- For example:
  
  \[
  \begin{align*}
  4 \ % \ 4 & = 0 \\
  5 \ % \ 4 & = 1 \\
  6 \ % \ 4 & = 2 \\
  7 \ % \ 4 & = 3 \\
  8 \ % \ 4 & = 0
  \end{align*}
  \]
Modulus

What is the result of the below expressions?

```c
int x = 10 % 3 + 1;
int y = 5 % 5;
int z = 15 % 7 - 2;
```
Modulus

- Modulus is great for restricting numbers to a particular range
- IF you mod by N, the result will be restricted to the range 0 through N-1
Randomness

- Complete this code
- The `setXToRandomNumber` function should replace `x` with some random number
- What if we want to restrict the range of numbers to be between 0 and 100?

```java
int x = 20;

void setup () {
    for (int i = 0 ; i < 100; i ++) {
        setXToRandomNumber();
        println("x is: " + x);
    }
}

void setXToRandomNumber() {
    x = (x*17) + 7 / 100;
}
```
Randomness

- Complete this code
- The `setXToRandomNumber` function should replace `x` with some random number
- What if we want to restrict the range of numbers to be between 0 and 100?

```java
int x = 20;

void setup () {
  for (int i = 0 ; i < 100; i ++) {
    setXToRandomNumber();
    println("x is: " + x);
  }
}

void setXToRandomNumber() {
  if (x == 0) {
    x += 7;
  }
  x = (x * 7 / x + 11 * x) % 100;
}
```
random()

- We will use the built-in function random()
- random(lower, upper)
  returns a random number between lower and upper
does not include upper
the number returned is of type float
- Example:
  random(0, 5) - returns numbers from 0 up to 5, not including 5

- [https://processing.org/reference/random_.html](https://processing.org/reference/random_.html)
Let’s do a few programming exercises with random()
random()

- Write a program that displays a sequence of random numbers between 0 and 100, one-per-second.
random()

void setup () {
  size(300, 200);
  textSize(50);
  frameRate(1);
}

void draw () {
  background(100);
  float r = random(0, 100);
  text(r, 25, 110);
}
random()

- Modify the program so that the decimal places are not displayed
random()

void setup () {
  size(300, 200);
  textSize(50);
  frameRate(1);
}

void draw () {
  background(100);
  float r = random(0, 100);
  int r2 = int(r);
  text(r2, 25, 110);
}
Fill out the program below that draws a rectangle of random size and starts in a random position 20 times per second.

```cpp
void setup() {
  size(400,400);
  frameRate(20);
}
void draw() {
  float x = ?
  float y = ?
  float w = ?
  float h = ?
  rect(x,y,w,h);
}
```
void setup() {
  size(400,400);
  frameRate(20);
}

void draw() {
  float x = random(0,400);
  float y = random(0,400);
  float w = random(40,100);
  float h = random(40,100);
  rect(x,y,w,h);
}
random()

- Modify `spacehip_target_practice` so that the target moves randomly
- Let’s try a few different types of movement
random()

- Modify `spaceship_target_practice` so that the target moves randomly
- Let’s try a few different types of movement
random()

- Modify `spaceship_target_practice` so that the target moves randomly
- Let’s try a few different types of movement
random()

- Modify `spaceship_target_practice` so that the target moves randomly
- Let’s try a few different types of movement
However, **PRNGs** need to be given a value to initialize the randomness

It will use this initial value to determine the random number(s) it provides

The **initial value** is also called the **seed value**

The **seed value** is a number that is the starting point

By default, seed is the current time (in milliseconds) on a computer

Can set it ourselves:

[https://processing.org/reference/randomSeed_.html](https://processing.org/reference/randomSeed_.html)
Random Seed

- Use the `randomSeed()` function
- Takes one argument - The seed to initialize with
- Let’s try this out in the examples we just did . . .
random()

- Add a seed value to spaceship_target_practice
- Run the program a few times
- What do you notice?
Materials

- Required Materials
  - [https://www.random.org/randomness/](https://www.random.org/randomness/)