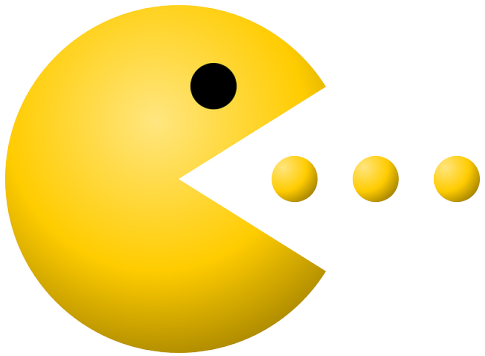
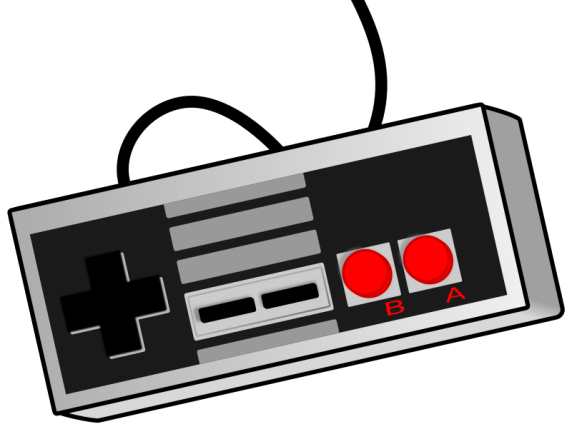




Lecture 02
Randomness and Inputs

PREPARED BY NICOLAS BERGERON

420-141-VA - GAME PROGRAMMING 1 - VANIER COLLEGE

Outline

Randomness in Games

Random numbers in Java

Processing Inputs

Random Numbers

Random numbers generated by computers are pseudo-random numbers in a deterministic sequences of values. The sequence of values after any specific value will be exactly the same

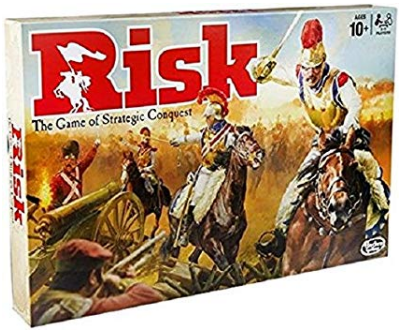
This is useful for tuning a game's mechanics.

- The effects of chance don't change each time the game is played, so the mechanics become deterministic and predictable.

Probabilities – Sum of two dice

Sum of two dice	Outcomes:	Probability
2	(1, 1)	1/36
3	(1, 2) (2, 1)	2/36
4	(1, 3) (2, 2) (3, 1)	3/36
5	(1, 4) (2, 3) (3, 2) (4, 1)	4/36
6	(1, 5) (2, 4) (3, 3) (4, 2) (5, 1)	5/36
7	(1, 6) (2, 5) (3, 4) (4, 3) (5, 2) (6, 1)	6/36
8	(2, 6) (3, 5) (4, 4) (5, 3) (6, 2)	5/36
9	(3, 6) (4, 5) (5, 4) (6, 3)	4/36
10	(4, 6) (5, 5) (6, 4)	3/36
11	(5, 6) (6, 5)	2/36
12	(6, 6)	1/36

Combat Mechanic in Risk



The goal of the game is World Domination

- Players are randomly assigned all the countries in the world
- When game progresses, they can add combat units on their countries
- Each turn, a player can attack an opponent's country to conquer it

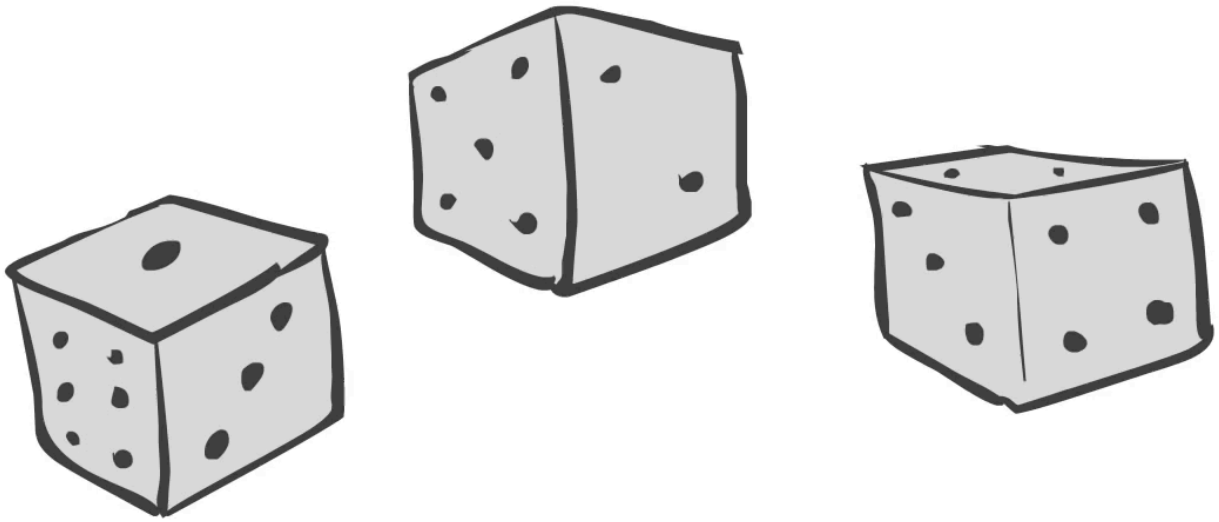
Combat system

- Attacker throws up to 3 dice (limit is the number of attacking units)
- Defender will throw up to 2 dice (limit is the number of defending units)

Dice as game mechanic
Tiny Dice Dungeon

https://www.youtube.com/watch?v=wSWRzmpJbWM

Randomness can get frustrating…



https://www.youtube.com/watch?v=-9ZI9kMsvRQ

Random numbers in Java

Random numbers allow …
Conditional events with probability of happening

Random numbers generated by computers are either:

1. real numbers in the range [0, 1[. (eg: Math.random())
2. integers in the range 0 … value – 1 (eg: in Greenfoot)

Then if an event has a probability of 30% of occurring (e.g., a weapon hitting its target), then

1. if the random number is less or equal to 0.3, the event occurs
2. If random between 0 and 99 is less than 30

Events with probability of 1 always occur; Events with probability of 0 never occur.

Exercises – Let's do it in Java

1. Write a method returning an integer random number between a and b inclusively
2. Write a number returning a real number (double) between a (inclusively) and b (exclusive) [a,b[
3. Write a method returning a boolean value taking the probability of returning true as parameter

Processing Inputs

Get keys in the order they were pressed

getKey

```
public static java.lang.String getKey()
```

Get the most recently pressed key, since the last time this method was called. If no key was pressed since this method was last called, it will return null. If more than one key was pressed, this returns only the most recently pressed key.

Returns:

The name of the most recently pressed key

How can we use this in a game?

Greenfoot.isKeyDown(String key)

isKeyDown

```
public static boolean isKeyDown(java.lang.String keyName)
```

Check whether a given key is currently pressed down.

Parameters:

`keyName - The name of the key to check`

Returns:

`True if the key is down`

Exercises

- Implement a method to determine if a key is up

More about key detection

It is often not enough to know if a key is up or down, other mechanics required to know

1. Is the key just pressed
2. Is the key held
3. Is the key just released

Give example of games using these mechanics.

How can we implement these behaviors?

Greenfoot.mousePressed(Object obj)

mousePressed

```
public static boolean mousePressed(java.lang.Object obj)
```

True if the mouse has been pressed (changed from a non-pressed state to being pressed) on the given object. If the parameter is an Actor the method will only return true if the mouse has been pressed on the given actor. If there are several actors at the same place, only the top most actor will receive the press. If the parameter is a World then true will be returned if the mouse was pressed on the world background. If the parameter is null, then true will be returned for any mouse press, independent of the target pressed on.

Parameters:

`obj - Typically one of Actor, World or null`

Returns:

`True if the mouse has been pressed as explained above`

Greenfoot.mouseClicked(Object obj)

mouseClicked

```
public static boolean mouseClicked(java.lang.Object obj)
```

True if the mouse has been clicked (pressed and released) on the given object. If the parameter is an Actor the method will only return true if the mouse has been clicked on the given actor. If there are several actors at the same place, only the top most actor will receive the click. If the parameter is a World then true will be returned if the mouse was clicked on the world background. If the parameter is null, then true will be returned for any click, independent of the target clicked on.

Parameters:

`obj - Typically one of Actor, World or null`

Returns:

`True if the mouse has been clicked as explained above`

Greenfoot.mouseDragged(Object obj)

mouseDragged

```
public static boolean mouseDragged(java.lang.Object obj)
```

True if the mouse is currently being dragged on the given object. The mouse is considered to be dragged on an object if the drag started on that object - even if the mouse has since been moved outside of that object.

If the parameter is an Actor the method will only return true if the drag started on the given actor. If there are several actors at the same place, only the top most actor will receive the drag. If the parameter is a World then true will be returned if the drag action was started on the world background. If the parameter is null, then true will be returned for any drag action, independent of the target clicked on.

Parameters:

`obj - Typically one of Actor, World or null`

Returns:

`True if the mouse has been dragged as explained above`

Greenfoot.mouseDragEnded(Object obj)

mouseDragEnded

```
public static boolean mouseDragEnded(java.lang.Object obj)
```

True if a mouse drag has ended. This happens when the mouse has been dragged and the mouse button released.

If the parameter is an Actor the method will only return true if the drag started on the given actor. If there are several actors at the same place, only the top most actor will receive the drag. If the parameter is a World then true will be returned if the drag action was started on the world background. If the parameter is null, then true will be returned for any drag action, independent of the target clicked on.

Parameters:

`obj - Typically one of Actor, World or null`

Returns:

`True if the mouse has been dragged as explained above`

Greenfoot.mouseMoved(Object obj)

mouseMoved

```
public static boolean mouseMoved(java.lang.Object obj)
```

True if the mouse has been moved on the given object. The mouse is considered to be moved on an object if the mouse pointer is above that object.

If the parameter is an Actor the method will only return true if the move is on the given actor. If there are several actors at the same place, only the top most actor will receive the move. If the parameter is a World then true will be returned if the move was on the world background. If the parameter is null, then true will be returned for any move, independent of the target under the move location.

Parameters:

`obj - Typically one of Actor, World or null`

Returns:

`True if the mouse has been moved as explained above`

Microphone Volume Greenfoot.getMicLevel()

getMicLevel

```
public static int getMicLevel()
```

Get the microphone input level. This level is an approximation of the loudness any noise that is currently being received by the microphone.

Returns:

The microphone input level (between 0 and 100, inclusive).

Do you know any game mechanic using the Microphone?

Questions

?