Conditional and Repeating Actions

Control Structure

1. Sequential Structures
2. Selection Structures
3. Repetition Structures (loops)

condition - true or false

Selection Structures

1. if-else
2. if-then
3. multi-way-if

Conditions using sensor methods

	Sensor Methods	
Method	**Purpose**	**Example**
`hasFlower()`	Does this Jeroo have any flowers?	`dean.hasFlower()`
`isClear(relativeDirection)`	Is there a clear space in the indicated direction? A clear space contains no flower, no net, no water, and no Jeroo. [`isClear(HERE)` is meaningless]	`dean.isClear(LEFT)`
`isFacing(compassDirection)`	Is this Jeroo facing in the indicated direction?	`dean.isFacing(NORTH)`
`seesFlower(relativeDirection)`	Is there a flower in the indicated direction?	`dean.seesFlower(HERE)`
`seesJeroo(relativeDirection)`	Is there another Jeroo in the indicated direction? [`seesJeroo(HERE)` is meaningless]	`dean.seesJeroo(AHEAD)`
`seesNet(relativeDirection)`	Is there a net in the indicated direction? [`seesNet(HERE)` is meaningless]	`dean.seesJeroo(RIGHT)`
`seesWater(relativeDirection)`	Is there water in the indicated direction? [`seesWater(AHEAD)` is meaningless]	`dean.seesWater(AHEAD)`

- Notably, when you see relativeDirection, your choices of direction include HERE, LEFT, RIGHT, and AHEAD. When you see compassDirection, your choices of direction are: NORTH, SOUTH, EAST, and WEST.

If-then-else structure syntax in java

```java
if (condition)
{
    // statements if true
```

```
}
else
{
    // statements if false
}

// example
if (timmy.seesNet(AHEAD))
{
    timmy.toss();
    timmy.turn(LEFT);
    timmy.turn(LEFT);
}
else
{
    timmy.turn(RIGHT);
}
```

if-then structure

```java
if (condition)
{
    // statements if and only if true
}

// example
if (jessica.seesNet(RIGHT))
{
    jessica.turn(RIGHT);
    jessica.toss();
    jessica.turn(LEFT);
}
```

multi-way structure

```java
if (condition1)
{
    x
}
else if (condition2) // you can as many else ifs as you want
{
    x
}
else
{
    x
}

//example
if (louisa.seesFlower(AHEAD))
{
    louisa.hop();
    louisa.pick();
```

```
}
else if (louisa.seesNet(AHEAD))
{
    louisa.toss();
}
else
{
    louisa.hop();
    louisa.turn(LEFT);
}
```

Compound Conditions

		Operators for conditions
Operator	**Java Symbol**	**Meaning**
Negation	`!` (exclamation point)	NOT
Conjunction	`&&` (2 keystrokes; no space between)	AND
Disjunction	`||` (2 keystrokes; no space between)	OR

Loops

while loop - continues while condition is true

until loop - continues until condition is true, terminates afterwards

pretest loop - condition is checked before loop starts for the first time

posttest loop - condition is tested after one run of the loop

```java
while (condition)
{
    // body statements
}

//example
while (kim.seesFlower(AHEAD))
{
    kim.hop();
    kim.pick();
}
// do loop, basically a while loop
do{
    kim.hop();
}while(name.isBlank());
```