TEALS MINECRAFT PROJECT

Lecture 8: Robot Damage & System Time

ROBOT DAMAGE

Lots of things can damage robots. They can fall into water, nearby monsters can attack them, they can attack each other, players can attack them, and so on.

When entities receive damage, the game calls their onEntityDamage() event handler function:

```
// Called whenever the entity receives damage.
public void Entity.onEntityDamage (DamageSource source, float amount);
```

If you implement this method, you can add code that does something whenever the robot is damaged.

ROBOT VISIBILITY

Robots can control their visibility:

```
// Set robot invisible or visible
public void Entity.setInvisible (boolean makeInvisible);
```

(When robots are invisible, you can still hit them, bump into them, and so forth.)

BLOCK DATA

Recall that each block position in the Minecraft world has a block of some type (air, log, grass, water, and so forth).

The following functions provide ways to determine what's at a world position, and ways to set the block at that position (just like you did for prior labs).

```
World world = entity.worldObj;   // Get a reference to entity's world

// Get the block & metadata at a given world position.
Block block = world.getBlock (positionX, positionY, positionZ);
int blockMetadata =
    world.getBlockMetadata (positionX, positionY, positionZ);

// Restore the block we just got above.
world.setBlock (positionX, positionY, positionZ, block);
world.setBlockMetadataWithNotify (
    positionX, positionY, positionZ, blockMetadata, 0);
```

ENTITY POSITION & ORIENTATION

World coordinates are integer, but entities can freely move around, so they get double (real-valued) coordinates.

```
// Get entity position.
double positionX = robot.posX;
double positionY = robot.posY;
double positionZ = robot.posZ;

// Restore entity position and orientation. This also sets the
// yaw (right/left turn) angle to zero, as well as the
// pitch (up/down direction) angle.
entity.setLocationAndAngles (positionX, positionY, positionZ, 0.0, 0.0);
```

TIME

System time is tracked in terms of milliseconds since midnight, January 1, 1970, UTC. This is also known as “Unix time”.

This sentence was typed on 2016 May 13, 08:02:13 UTC, which is 1463126533 in Unix time. That's 1,463,126,533 milliseconds since midnight, January 1, 1970.

Here's how we get the current time in Java:

```
long currentTime = System.currentTimeMillis();
```

Class Exercise: How would you use the above function to figure out if 20 seconds has elapsed from some earlier time?

LAB 8: CHAMELEON ROBOT

In Lab 8, you will create a chameleon robot. This robot slowly wanders around the world. When it receives damage, it disguises itself as a nearby block for some time. After enough time has passed, it restores its form as a chameleon bot and continues wandering.