

Greenfoot Chapter 3

3.1: `if(Grennfoot.getRandomNumber(100) < 10){}`

3.2: `if(Greenfoot.getRandomNumber(100)<7){}`

3.3: `public void act()
{
 if(isAtEdge())
 {
 turn(35);
 }
 if(Greenfoot.getRandomNumber(100)<25){
 turn(15);
 }
 move(5);
}`

3.4: Not only does the crab turn randomly, but the turn degree is random as well.

3.5: `public void act()
{
 if(isAtEdge())
 {
 turn(35);
 }
 if(Greenfoot.getRandomNumber(100)<25){
 if(Greenfoot.getRandomNumber(101)>50){`

```

        turn(Greenfoot.getRandomNumber(45));
    }
    else{
        turn(-Greenfoot.getRandomNumber(45));
    }
}
move(5);
}

```

3.6: With multiple crabs, they do not turn all at once because the random number only allows them to turn 25% of the time.

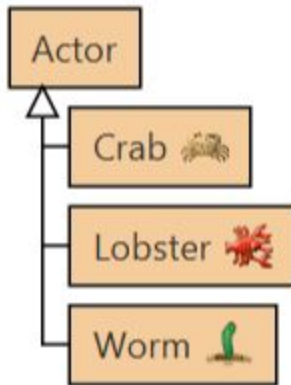
3.7: The crabs move around and turn randomly while the worms do nothing. When a crab meets a worm, nothing happens.

3.8:

```

public void randomTurn(){
    if(Greenfoot.getRandomNumber(100)<25){
        if(Greenfoot.getRandomNumber(101)>50){
            turn(Greenfoot.getRandomNumber(45));
        }
        else{
            turn(-Greenfoot.getRandomNumber(45));
        }
    }
}

```



3.9:

3.10: Without code lobster will do nothing, and nothing is what they did.

3.11: `public void act()`

```
{
    move(5);
    turnAtEdge();
    randomTurn();
    lookForWorm();
}
public void lookForWorm(){
    if(isTouching(Worm.class)){
        removeTouching(Worm.class);
    }
}
public void randomTurn(){
    if(Greenfoot.getRandomNumber(100)<25){
        if(Greenfoot.getRandomNumber(101)>50){
            turn(Greenfoot.getRandomNumber(45));
        }
        else{
```

```

        turn(-Greenfoot.getRandomNumber(45));
    }
}
}
public void turnAtEdge(){
    if( isAtEdge())
    {
        turn(35);
    }
}
}

```

3.12:

```
public void lookForCrab(){
    if(isTouching(Crab.class)){
        removeTouching(Crab.class);
    }
}
```

3.13: The crab was unable to eat all of the worms before the lobsters got to them.

3.14 - 3.18:

```
public void act()
{
    move(5);
    lookForWorm();
    turnCrab();
}
public void turnCrab(){
    if(Greenfoot.isKeyDown("left")){
```

```

        turn(-5);
    }
    if(Greenfoot.isKeyDown("right")){
        turn(5);
    }
}
public void lookForWorm(){
    if(isTouching(Worm.class)){
        removeTouching(Worm.class);
    }
}
}

```

3.19: Crab = move(2);
 Lobster = move(3);

3.20: static void stop();
 Method name is stop.

3.21: stop has no parameters and returns nothing

```

3.22: public void lookForCrab(){
        if(isTouching(Crab.class)){
            removeTouching(Crab.class);
            Greenfoot.stop();
        }
    }
}

```

3.23: Greenfoot.playsound(`java.lang.String` soundFile)

The name is playSound and it expects 1 parameters and returns nothing

```
3.24: public void lookForWorm(){  
    if(isTouching(Worm.class)){  
        removeTouching(Worm.class);  
        Greenfoot.playSound("slurp.wav");  
    }  
}
```

```
3.25: public void lookForCrab(){  
    if(isTouching(Crab.class)){  
        removeTouching(Crab.class);  
        Greenfoot.stop();  
        Greenfoot.playSound("oof.wav");  
    }  
}
```

3.26: Greenfoot.getMicLevel(); has no parameters.

3.27 getMicLevel returns an integer. This means the level could be increased or decreased.

3.28: Any positive number.

3.29: The method is static which means It will not reset its value when you use the reset button.

3.30: Greenfoot.getMicLevel();

3.31: move(5);

3.32, 3.33: public void act()

```
{  
    if(Greenfoot.getMicLevel() > 3){  
        move(-10);  
    }  
    else{  
        move(5);  
    }  
}
```

3.34: public void moveLeftNoise(){

```
    if(Greenfoot.getMicLevel() > 3){  
        move(-10);  
    }  
    else{  
        move(5);  
    }  
}
```

3.35: public void gameOver(){

```
    if(isAtEdge()){  
        Greenfoot.playSound("oof.wav");  
        Greenfoot.stop();  
    }
```

```
}
```

```
3.36: public void gameOver(){
    Greenfoot.playSound("oof.wav");
    Greenfoot.stop();
}
public void checkGameOver(){
    if(isAtEdge()){
        gameOver();
    }
}
```

```
3.38, 3.39: public void act()
{
    checkGameOver();
    move(5);
    reset();
}
public void reset(){
    if(isAtEdge()){
        setRotation(180);
    }
}
public void gameOver(){
    Greenfoot.playSound("oof.wav");
    Greenfoot.stop();
}
public void checkGameOver(){
```



```
        if(isTouching(Stickman.class)){  
            gameOver();  
        }  
    }  
}
```