

## Άσκηση: Κίνηση στον χώρο 2 - αλληλεπίδραση

Στη συνέχεια να δημιουργήσουμε ένα απλό είδος παιχνιδιού. Στον κόσμο μας θα εμφανίζονται τυχαία αντικείμενα όπως «κρανία» και «βόμβες» τις οποίες θα καλείται ο χρήστης να καταστρέψει κατευθύνοντας την σφαίρα επάνω τους. Αρχικά δημιουργήστε 2 νέες υποκλάσεις κάνοντας *δεξιά κλικ* στην κλάση *Actor* και επιλέγοντας *Νέα υποκλάση*, δώστε για όνομα *Skull* και για εικόνα από την κατηγορία εικόνων *symbols* επιλέξτε την *skull.png*. Στη συνέχεια με αντίστοιχη μεθοδολογία δημιουργήστε την κλάση *Bomb* επιλέγοντας αυτή τη φορά από την κατηγορία εικόνων *objects* την βόμβα. Στη συνέχεια στην ενότητα *myWorld* κατασκευάστε αντίγραφα των κλάσεων *Skull* και *Bomb*:

```
public myWorld()
{
    super(600, 400, 1);           //έχει δημιουργηθεί
    Ball Ball1 = new Ball();     //έχει δημιουργηθεί
    Skull Skull1 = new Skull();
    Bomb Bomb1 = new Bomb();
    addObject(Ball1,150,150);    //έχει δημιουργηθεί
}
```

Τα κρανία και οι βόμβες θέλουμε να δημιουργούνται αυτόματα και με τυχαίο τρόπο στον κόσμο μας. Για αυτό χρησιμοποιούμε την ενσωματωμένη μέθοδο του περιβάλλοντος `Greenfoot.getRandomNumber()` η οποία επιστρέφει τυχαίους αριθμούς για τις συντεταγμένες  $x_1$ ,  $y_1$  της κλάσης *Skull* και  $x_2$ ,  $y_2$  της *Bomb* : (δημιουργήστε την μέθοδο `act()` μέσα στην ενότητα *myWorld*)

```
public void act()
{
    int x1,y1,x2,y2,t1,t2;
    t1=Greenfoot.getRandomNumber(200);
    if (t1<=3){ //πιθανότητα 3/200 να εμφανιστεί
        x1 = 10 + Greenfoot.getRandomNumber(540); //για να είναι ορατό εντός του κόσμου
        y1 = 10 + Greenfoot.getRandomNumber(360);
        addObject(new Skull(), x1,y1);
    }
    t2=Greenfoot.getRandomNumber(350);
```

```
if (t2<=__) { //πιθανότητα 3/350 να εμφανιστεί
```

```
x2 = _____ (____);
```

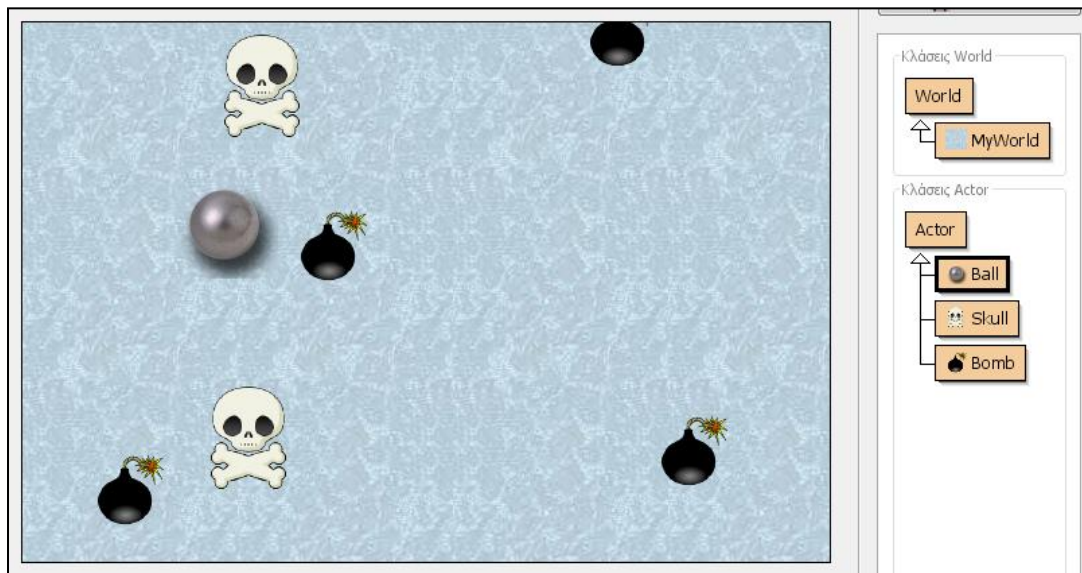
```
y2 = _____ (____);
```

```
addObject(_____, ____, ____);
```

```
}
```

```
}
```

Στη συνέχεια μεταγλωττίστε κάθε κλάση ξεχωριστά (ή πατήστε ξανά το πλήκτρο «Μεταγλώττιση Όλων» - ανάλογα την έκδοση) για να δημιουργηθεί ο κόσμος με τις μορφές του στο περιβάλλον του Greenfoot. Αν υπάρχουν συντακτικά λάθη προβείτε στις αντίστοιχες διορθώσεις.



Τώρα στην κλάση της σφαίρας (Ball) θα πρέπει να προσθέσετε μια μέθοδο η οποία κάθε φορά που ακουμπάει η σφαίρα το κρανίο ή την βόμβα τα αντικείμενα αυτά να εξαφανίζονται από τον κόσμο. Η μέθοδος αυτή να ονομαστεί hitEnemy() και να καλείται μέσα στην act(). Ο ορισμός της γίνεται ως εξής (μέσα στην public class Ball extends Actor):

```
public void hitEnemy()  
{  
    if (isTouching(Skull.class)){  
        removeTouching(Skull.class);  
    }  
}
```

```
}
```

Με τον παραπάνω κώδικα εξαφανίζονται τα «κρανία». Προσθέστε επιπλέον κώδικα και για τις «βόμβες».

Για τερματίζει το παιχνίδι να το προγραμματίσετε να σταματάει όταν η σφαίρα «εξολοθρεύσει» συνολικά 30 αντικείμενα κλάσης «κρανίου» και «βόμβας». Θα πρέπει να οριστούν 3 μετρητές (καθολικές μεταβλητές) στην αρχή του κώδικα της κλάσης Ball:

```
int counter1=0,counter2=0;
int counter3 = 0;
```

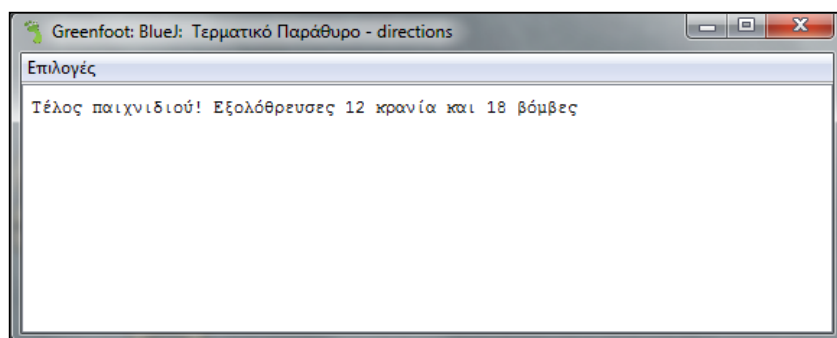
Στη συνέχεια στην μέθοδο hitEnemy() οι μετρητές να αυξάνονται μοναδιαία κάθε φορά που συμβαίνει σύγκρουση:

```
if (isTouching(Skull.class)){
    removeTouching(Skull.class);
    counter1++;
    checkEnd();
}
```

Η μέθοδος checkEnd() καλείται να ελέγξει το σύνολο των «βομβών» και «κρανίων» που έχει ακουμπήσει η σφαίρα έχει φτάσει τον αριθμό 30:

```
public void checkEnd(){
    counter3=_____+_____;
    if (counter3==_____){
        Greenfoot.stop();
        System.out.println("Τέλος παιχνιδιού! Εξολόθρευσες " +counter1 + " κρανία και "
+counter2 + " βόμβες");
    }
}
```

Αν αυτό συμβαίνει το παιχνίδι τερματίζει και εμφανίζεται παράθυρο μηνύματος στην οθόνη:



Συμπληρώστε τον κώδικα, μεταγλωττίστε και τρέξτε το έργο. Προβείτε σε όποιες απαραίτητες τροποποιήσεις απαιτούνται. Κατόπιν βελτιώστε το παιχνίδι προσθέτοντας 3<sup>ο</sup> αντικείμενο «βαρέλι» (barel) το οποίο θα εμφανίζεται με