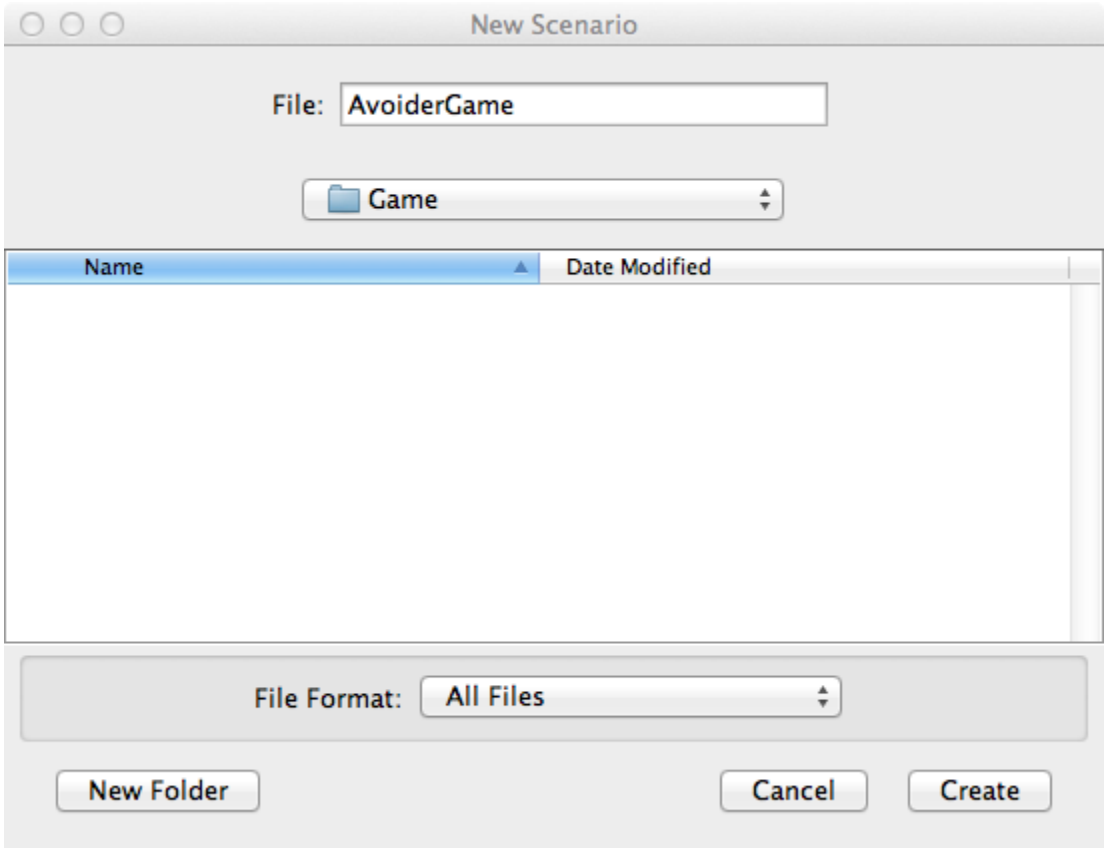
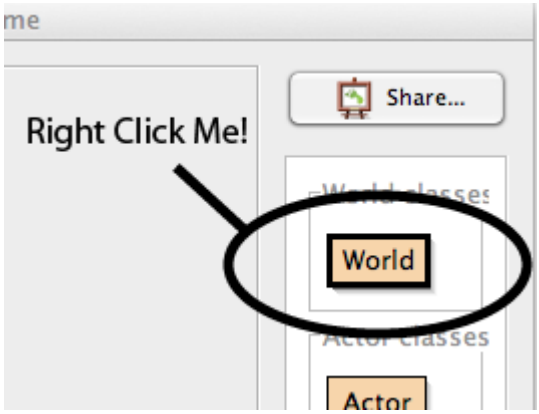
Δημιουργία παιχνιδιού Avoider

Δημιουργήστε ένα νέο σενάριο μες το όνομα «AvoiderGame» και πατήστε Create.



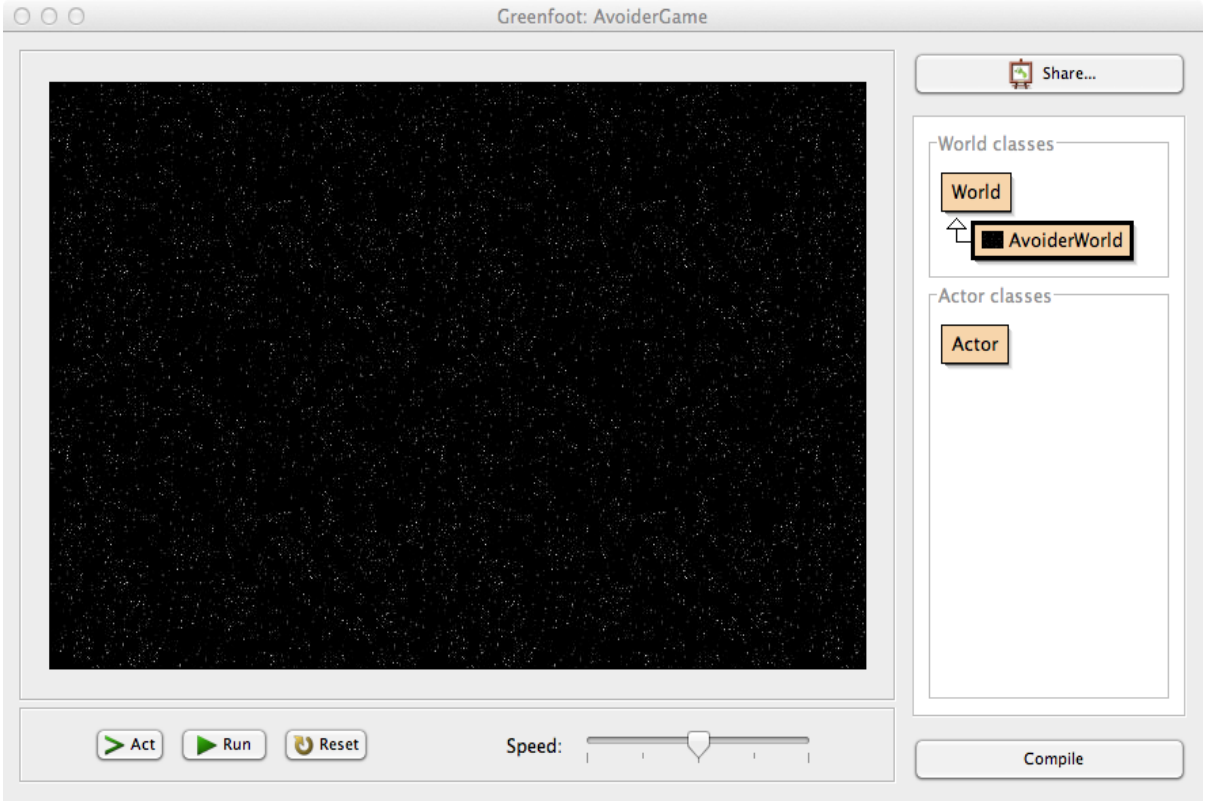
Name the scenario "AvoiderGame" and then hit the "Create" button.

Κάντε δεξί κλικ στο *World* class καιεπιλέξτε "New subclass...".



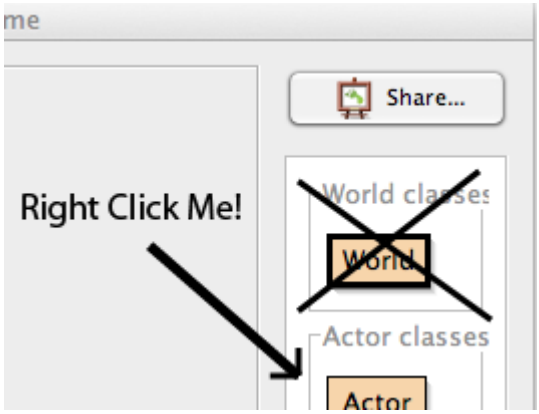
Στο παράθυρο που εμφανίζεται δώστε όνομα στην class "AvoiderWorld" και μετά επιλέξτε "backgrounds→space1.jpg" ως εικόνα της

Πατήστε "Compile" στο κυρίως παράθυρο:



Δημιουργία των χαρακτήρων (Avatar – Enemy).

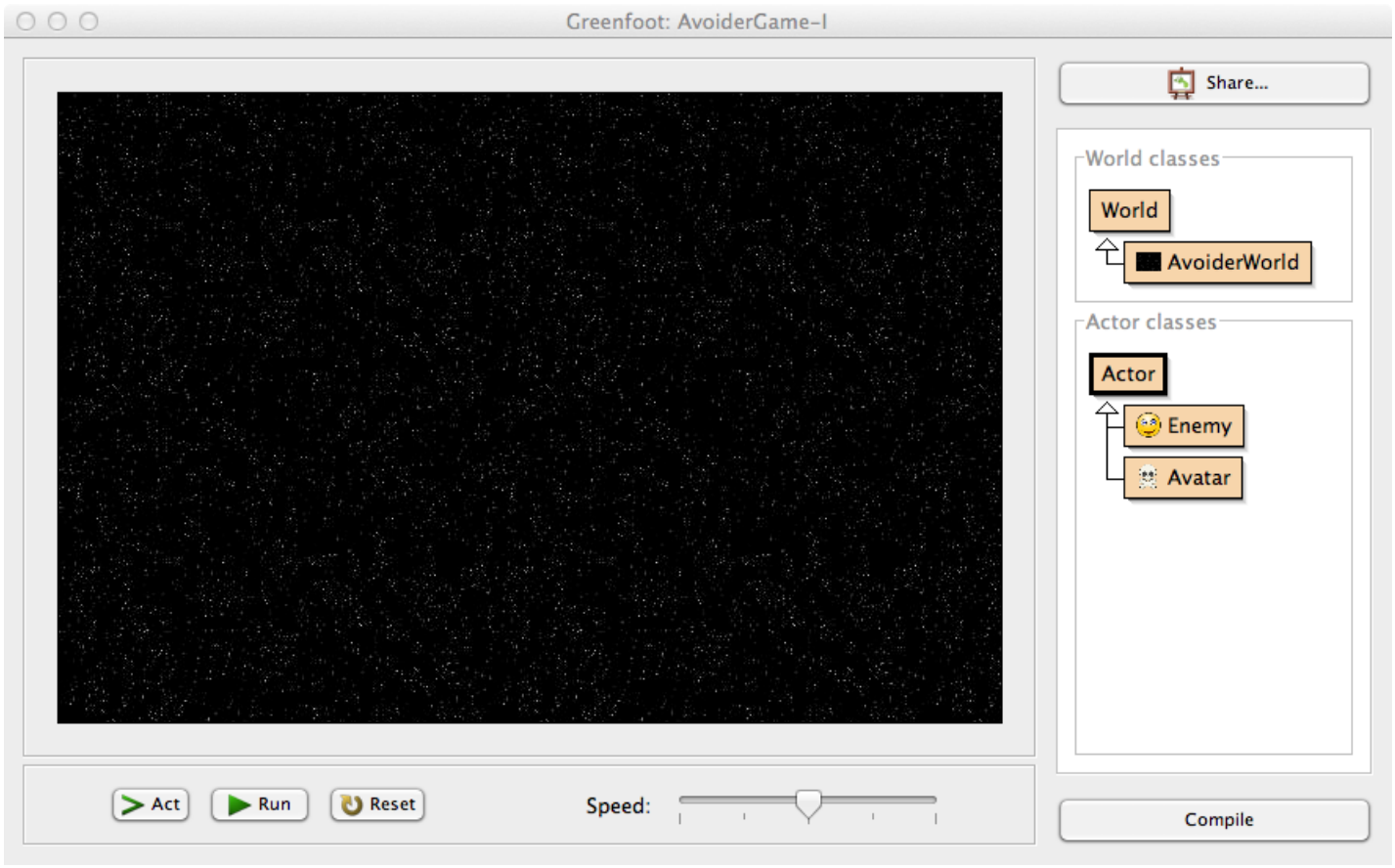
Δεξί κλικ στο *Actor* class και επιλέξτε "Newsubclass...".



Στο "New class" popup window, δώστεόνομα "**Avatar**" και επιλέξτε την εικόνα "symbols→skull.png". Στο κυρίως παράθυρο πατήστε "Compile".

Τώρα με τα ίδια βήματα δημιουργήστε τον εχθρό→Δεξί κλικ στο *Actor* class και επιλέξτε "New subclass...", δώστε όνομα "**Enemy**" και επιλέξτε την εικόνα→"symbols→Smiley2.png" Στο κυρίως παράθυρο πατήστε "Compile".

Θα πρέπει να δείτε κάτι όπως παρακάτω:



Τι έχουμε κάνει ως τώρα;

Το Greenfoot βλέπει ένα πρόγραμμα σαν *World* που περιέχει *Actors*. Ο World δημιουργεί, προσθέτει και καταργεί *Actors* από την οθόνη, και περιοδικά καλεί την *act()* method του κάθε *Actor*. Ο κάθε *Actor* παίζει τον ρόλο του, δηλαδή την *act()* method που περιγράφει τι πρέπει να κάνει.

Προσθήκη του ήρωα Avatar

Τώρα θα προσθέσουμε τον ήρωά μας στο παιχνίδι.

ΠΡΟΣΟΧΗ!!! Εκτελέστε προσεκτικά κάθε βήμα:

Για να το πετύχετε αυτό κάντε δεξί κλικ στην Avatar class, και επιλέξτε select new Avatar() από το μενού που εμφανίζεται, τοποθετήστε την εικόνα του στο κέντρο της οθόνης και κάντε αριστερό κλικ.

Τώρα κάντε δεξί κλικ οπουδήποτε στην οθόνη του παιχνιδιού και επιλέξτε «Save the world» στο μενού που εμφανίζεται.

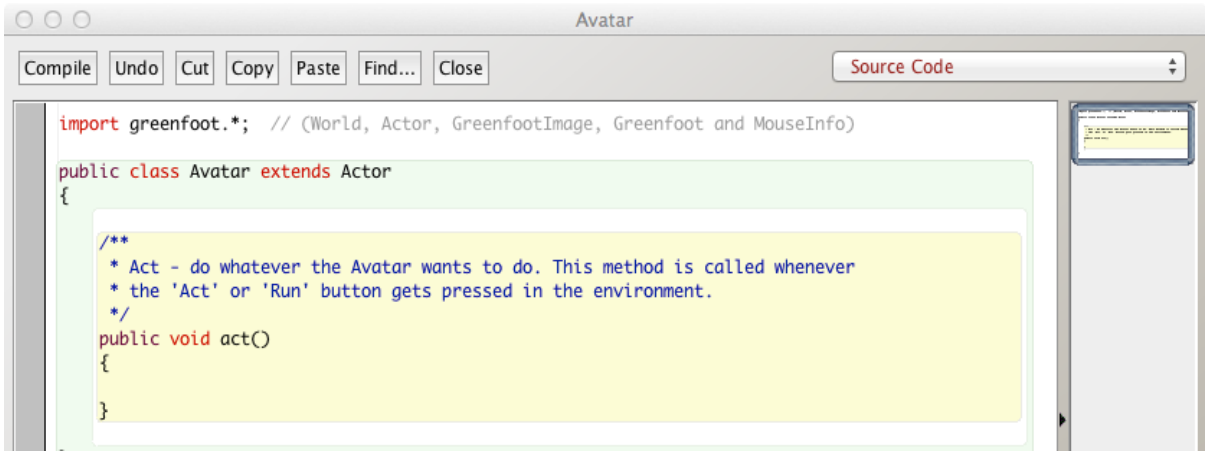
Με αυτό, ο ήρωάς μας τοποθετείται αυτόματα στην οθόνη του παιχνιδιού.

Αν πατήσετε τώρα το κουμπί Reset θα δείτε τον ήρωά μας τοποθετημένο στο κέντρο της οθόνης.

ΈλεγχοςτουήρωαAvatar

Προσθήκη κώδικα στην *Avatar* class που θα μας επιτρέπει να τον ελέγχουμε με το ποντίκι.

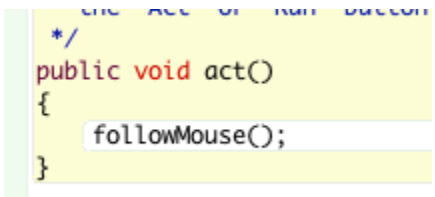
➔ Διπλό κλικ ή Δεξί κλικ →"openeditor" στο *Avatar*:



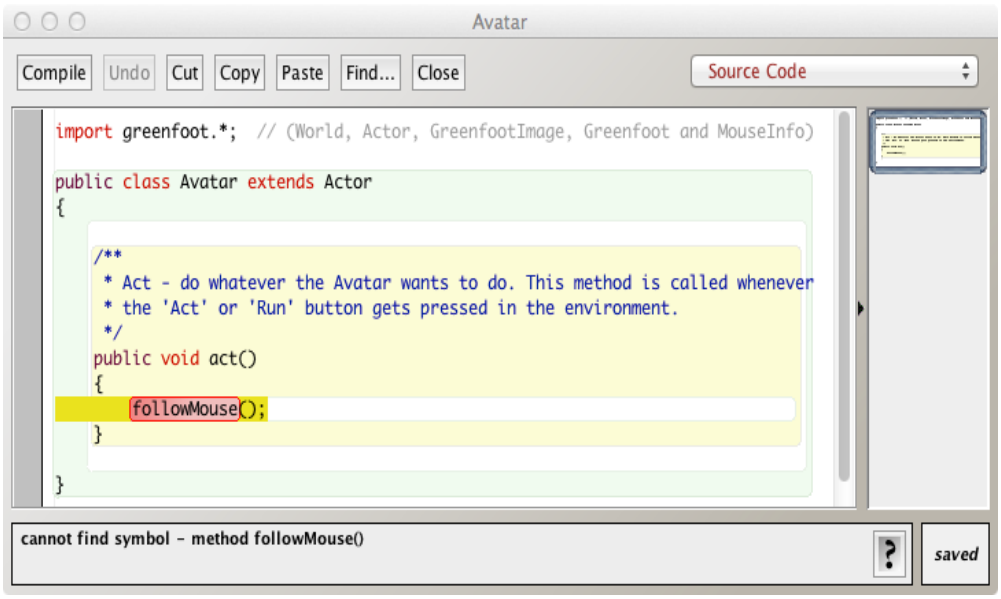
Τώρα βλέπετε την *act()* method η οποία δεν περιέχει κώδικα.

Θέλουμε να κάνουμε το ***Avatar*** να ακολουθεί το ποντίκι.

Μέσα στην *act()* method, πληκτρολογήστε: " followMouse();". Όπως φαίνεται παρακάτω:

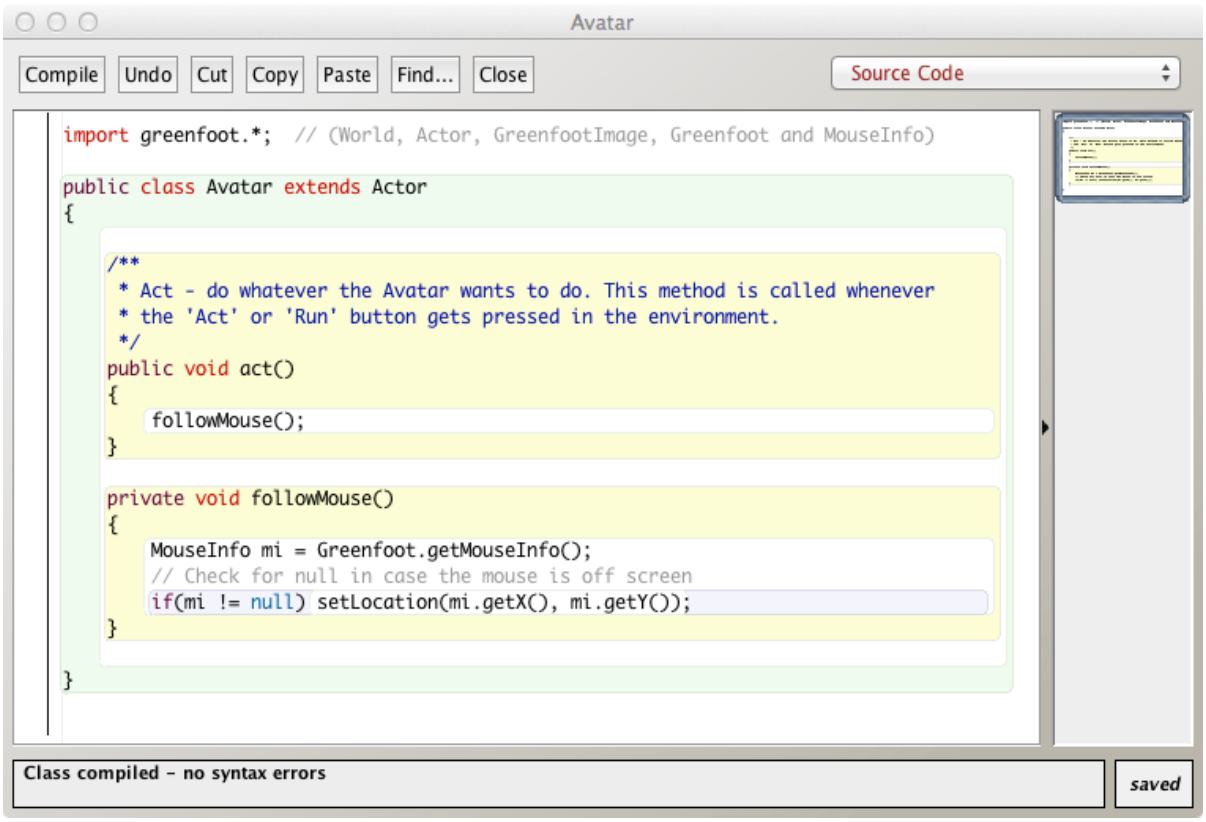


Πατήστε το πλήκτρο **"Compile"** για να δείτε τι θα συμβεί. Τι παρατηρείτε;



Το Greenfoot μας εμφανίζει ένα μήνυμα σφάλματος, επισημαίνοντας τη λέξη **followMouse();** Αυτό συμβαίνει επειδή η **συνάρτηση followMouse()** δεν υπάρχει, γι' αυτό λοιπόν θα τη δημιουργήσουμε εμείς και θα ορίσουμε τι ακριβώς θα κάνει.

Γράψτε προσεκτικά τον παρακάτω κώδικα όπως φαίνεται στην εικόνα για να ορίσετε τι ακριβώς θα κάνει η συνάρτηση **followMouse()**:



Πατήστε το πλήκτρο **"Compile"**. Δοκιμάστε την εφαρμογή σας μετακινώντας το ποντίκι. Τι συμβαίνει;

Ο κώδικας της class **"MouseInfo"** κάνει τα εξής: Πρώτα καλούμε ένα αντικείμενο γράφοντας **"Greenfoot.getMouseInfo();"**το οποίο περιέχει πληροφορίες που αφορούν το ποντίκι και το χρησιμοποιώ για να πάρω και να χρησιμοποιήσω τις συντεταγμένες X και Y του ποντικιού στην οθόνη **"setLocation(x,y)".**

Γράφω επίσης το **"if(mi != null)"** γιατί αν κατα λάθος κουνήσετε το ποντίκι εκτός του παραθύρου του Greenfoot, δεν θα υπάρχει πλέον διαθέσιμη η πληροφορία θέσης του ποντικιού και θα είχαμε σφάλμα.

Τι θα κάνουμε με τον enemy;

Ας κάνουμε λοιπόν και τον ***Enemy*** να κινείται. Κάντεδιπλόκλικ(ή δεξί κλικ →Openeditor…) στην***Enemy*** **class**καιγράψτε τον κώδικαστην***act()*** **method**όπως φαίνεται παρακάτω:

```
 */
public void act()
{
    setLocation(getX(), getY() + 1);
}


/*public void act()
```

Η **setLocation()** είναι μια class *του* **Avatar** που χρησιμοποιήσαμε πριν λίγο. Τι κάνει αυτός ο κώδικας τώρα; Δίνει κίνηση. Καθώς το πρόγραμμα τρέχει, ο ***Enemy*** κινείται ένα pixelπρος τα κάτω συνεχώς.

Πατήστε το πλήκτρο **"Compile"**, και τοποθετήστε έναν **Enemy** στην οθόνη. Δοκιμάστε να δείτε τι συμβαίνει;

Στρατός! Enemies...Enemies...Enemies...

Ας γράψουμε έναν κώδικα που θα δημιουργεί στρατό από ***enemies*** που θα επιτίθενται στον **Avatar.**

Κάντε διπλό κλικ (ή δεξί κλικ →Open editor…) στον **AvoiderWorld** και δημιουργήστε την παρακάτω method όπως φαίνεται στην εικόνα:

```
public void act()
{
    // Randomly add enemies to the world
    if(Greenfoot.getRandomNumber(1000) < 20)
    {
        Enemy e = new Enemy();
        addObject(e, Greenfoot.getRandomNumber(getWidth()-20)+10, -30);
    }

}
```

Compileandrun. Τισυμβαίνει;

Αν θέλουμε οι enemiesνα εξαφανίζονται από την οθόνη τότε θα πρέπει να αλλάξουμε τον κώδικα στο AvoiderWorld από super(600, 400, 1); Σε super(600, 400, 1, false);

Αλλάξτε το κώδικα σώστε (Compileandrun) και ξαναδοκιμάστε. Τι συμβαίνει;

Διαχείριση μνήμης

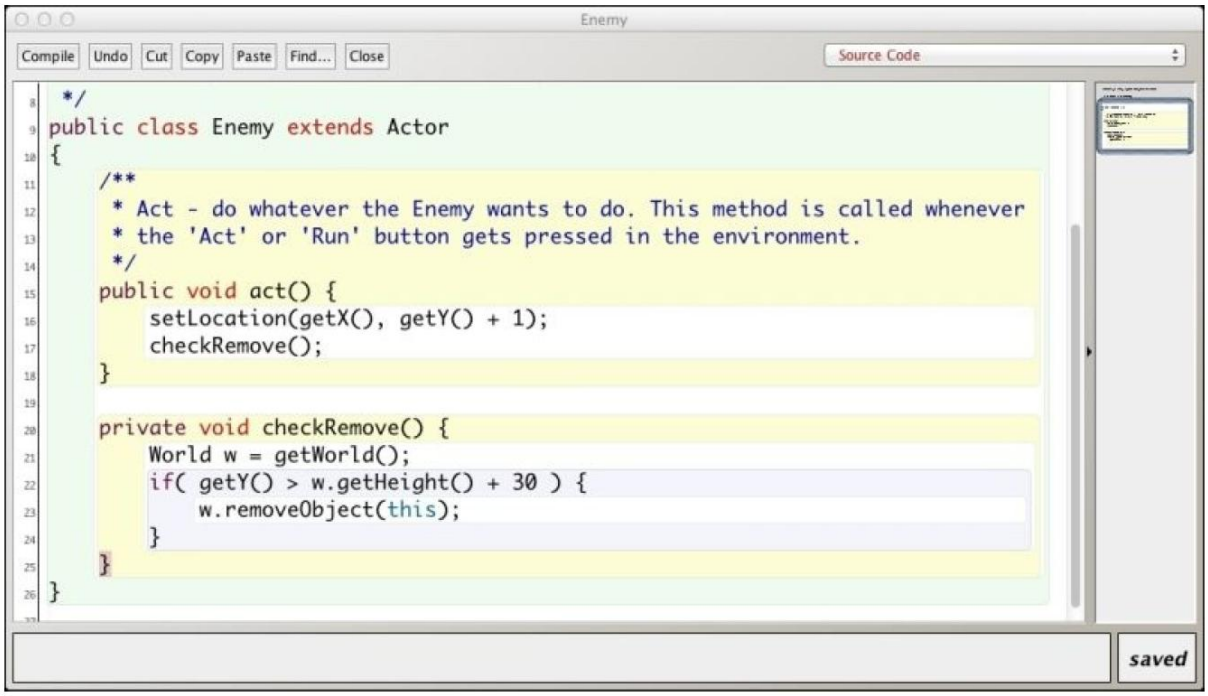
Αν θέλουμε να διαχειριστούμε σωστά τη μνήμη του υπολογιστή κατά τη διάρκεια του παιχνιδιού θα πρέπει να φροντίσουμε, οι enemiesπου δημιουργούνται, περνάνε από την οθόνη και εξαφανίζονται, να μην καταναλώνουν πόρους του υπολογιστή παραμένοντας στη μνήμη του. Αυτό μπορούμε να το πετύχουμε **καλώντας τη methodremoveObject(),** η οποία αφαιρεί και από τη μνήμη του υπολογιστή όποιον enemyεξαφανιστεί από την οθόνη μας. Το πιο σωστό σημείο για να καλέσουμε τη **methodremoveObject(),** είναι μέσα από τον ίδιο τον enemy.

Ανοίξτε τον editortης κλάσης **Enemy**και προσθέστε στο τέλος του κώδικα:

```
checkRemove();
```

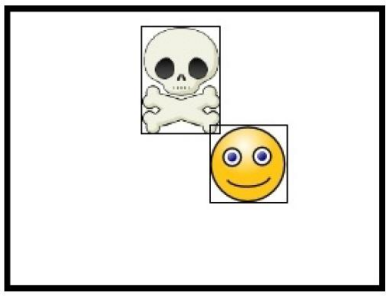
Και τώρα προσθέστε την **checkRemove() method**, γράφοντας κάτω από την **act() method** τον παρακάτω κώδικα:

```
private void checkRemove() {
        World w = getWorld();
        if(getY() >w.getHeight() + 30 ) {
        w.removeObject(this);
        }
}
```



Compile and run. Σώστε και δοκιμάστε το παιχνίδι.

Έλεγχος επαφής-σύγκρουσης. Collisiondetection



Θέλουμε το παιχνίδι μας να ελέγχει, αν ο Avatarέρθει σε επαφή με έναν εχθρό enemyνα σταματάει, να εξαφανίζεται ο Avatar και το παιχνίδι να σταματάει.

Για να το πετύχουμε αυτό πρέπει να χρησιμοποιήσουμε την methodgetOneIntersectingObject()και να τροποποιήσουμε την **Avatar class** προσθέτοντας τον παρακάτω κώδικα:

```
public void act() {
followMouse();
checkForCollisions();
}
```

Και προσθέστε τον κώδικα της checkForCollisions() method κάτω από την act() method:

```
private void checkForCollisions() {
        Actor enemy = getOneIntersectingObject(Enemy.class);
        if( enemy != null ) {
        getWorld().removeObject(this);
        Greenfoot.stop();
        }
}
```

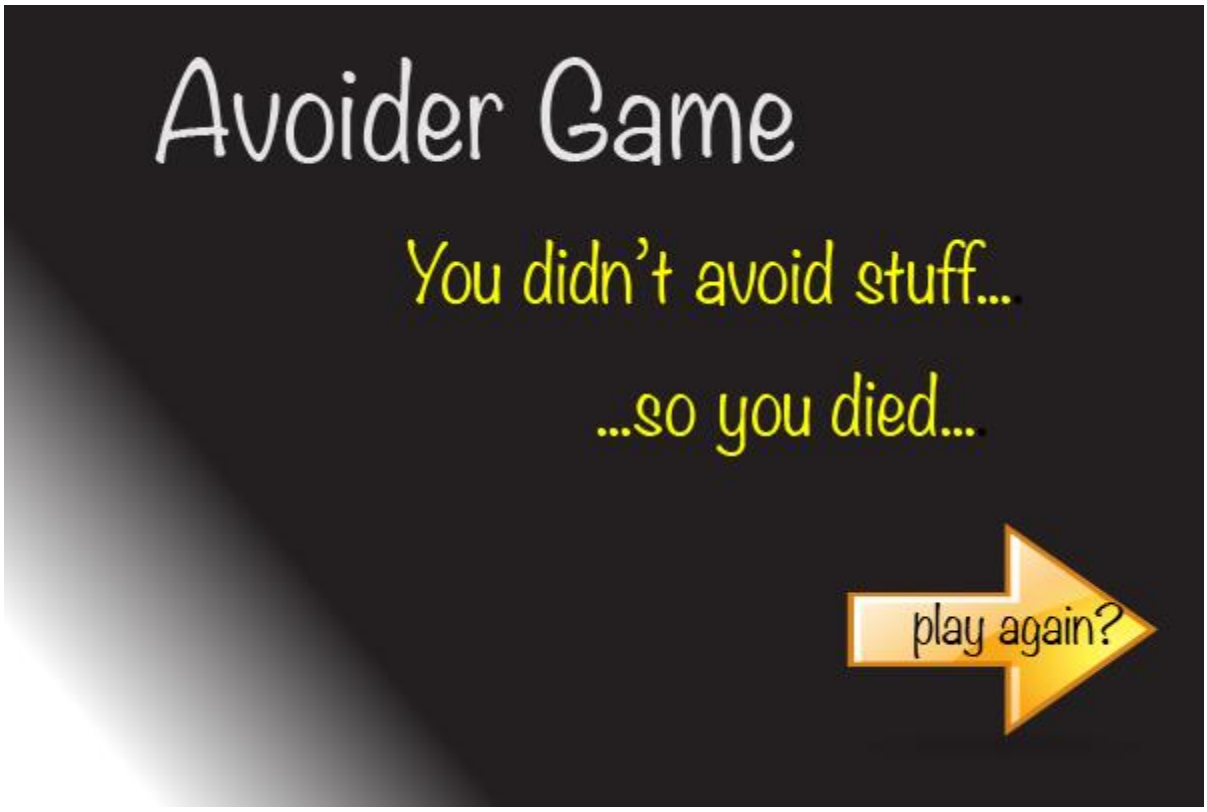
Compile and Run.

Σώστε και τρέξτε το παιχνίδι σας. Δοκιμάστε το. Προσθέστε έναν Avatarστην οθόνη με δεξί κλικ, μετά κλικ στην οθόνη, και προσπαθήστε να αποφύγετε τους εχθρούς που κατεβαίνουν. Χάστε και ξαναπροσπαθήστε το ίδιο μερικές φορές ακόμη.

Τι συμβαίνει;

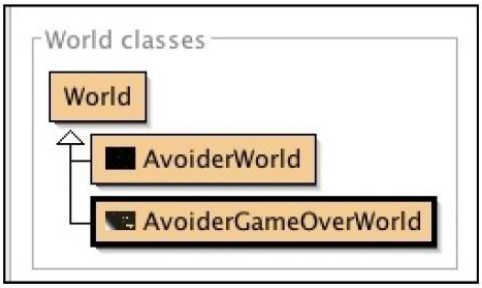
Οθόνη GameOverScreen

Θα δημιουργήσουμε τώρα μια οθόνη Game Over η οποία θα εμφανίζεται μόλις τελειώσει το παιχνίδι μας. Κατεβάστε και αποθηκεύστε την εικόνα που χρειαζόμαστε από εδώ: https://tinyurl.com/tp47pvt



Ένας νέος κόσμος (NewWorld) ...

Ακολουθώντας τα ίδια βήματα με τα οποία δημιουργήσατε το **AvoiderWorld**, δημιουργήστε ένα νέο κόσμο και ονομάστε τον **"AvoiderGameOverWorld"** στον οποίο θα θέσετε ως εικόνα την εικόνα που κατεβάσατε πριν.



Για να το πετύχουμε αυτό πρέπει να αλλάξουμε τον κώδικα μέσα στην **checkForCollisions()**όπως φαίνεται παρακάτω. Η αλλαγή πρέπει να γίνει μέσα στη δομή if:

```
private void checkForCollisions() {
        Actor enemy = getOneIntersectingObject(Enemy.class);
        if( enemy != null ) {
        AvoiderWorld world = (AvoiderWorld) getWorld();
        world.endGame();
        }
}
```

Η διαφορά είναι, ότι, τώρα με τον κώδικα τερματίζουμε το παιχνίδι αντί να εξαφανίσουμε τον Avatar και θέλουμε να εμφανιστεί και η οθόνη GameOver.

Επίσης μέσα στην **AvoiderWorld** προσθέστε τον εξής κώδικα όπως φαίνεται παρακάτω:

```
public void endGame() {
    AvoiderGameOverWorld go = new AvoiderGameOverWorld();
    Greenfoot.setWorld(go);
}
```

Compile and Run.

Δοκιμάστε ξανά το παιχνίδι σας. Προσθέστε έναν Avatarπροσπαθήστε να αποφύγετε τους εχθρούς και δείτε τι συμβαίνει.

Εμφανίζεται η οθόνη GameOver;

Ξαναπαίξε - PlayAgain.

Θέλουμε να κάνουμε το παιχνίδι να ξαναρχίζει όταν πατήσουμε στην οθόνη "GameOver".

Μέσω της *AvoiderGameOverWorld* θα ελέγξουμε αν γίνει κάποιο κλικ και αν ναι, να εκκινεί ξανά το παιχνίδι στο *AvoiderWorld*. Για να το πετύχουμε θα χρησιμοποιήσουμε τη συνάρτηση mouseClicked(), την οποία θα πληκτρολογήσουμε μέσα στην act() methodως εξής:

```
public void act()
{
    // Restart the game if they user clicks the mouse anywhere in the screen
    if( Greenfoot.mouseClicked(this))
    {
        AvoiderWorld aworld = new AvoiderWorld();
        Greenfoot.setWorld(aworld);
    }
}
```

Compile και run. Δουλεύουν όλα κανονικά; Αν υπάρχει κάποιο λάθος στον κώδικα, διορθώστε το, και επαναλάβετε.

Θα πρέπει να ξεκινά ένα παιχνίδι, να εμφανίζονται κατεβαίνοντας οι εχθροί, να μπορείτε να προσθέσετε έναν Avatar, να αποφεύγετε τους εχθρούς και αν αγγίξετε κάποιον να τελειώνει το παιχνίδι με την εμφάνιση της οθόνης GameOver. Με κλικ στην οθόνη θα πρέπει να ξεκινά ένα νέο παιχνίδι. Δουλεύουν όλα κανονικά;

Οθόνη έναρξης

Κατεβάστε την παραπάνω εικόνα από εδώ και αποθηκεύστε την: https://tinyurl.com/wrr7c2b

Δημιουργήστε μια subclass της Worldμε το όνομα «AvoiderGameIntroScreen», και θέστε ως εικόνα αυτήν που μόλις κατεβάσατε.

Θέλουμε το παιχνίδι μας να ξεκινάει με αυτή την οθόνη, και για να το πετύχουμε αυτό κάνουμε δεξί κλικ στην subclass«AvoiderGameIntroScreen», που μόλις δημιουργήσαμε και επιλέγουμε newAvoiderGameIntroScreen().

Compile and Run. Δοκιμάστε το παιχνίδι σας. Τι συμβαίνει;

Πρέπει τώρα να κάνουμε το παιχνίδι μας να ξεκινάει με ένα κλικ στην αρχική οθόνη. Τι πρέπει να κάνουμε;

Προσθέστε τον παρακάτω κώδικα μέσα στην **subclass «AvoiderGameIntroScreen»**

```
public void act() {
        // Start the game if the user clicks the mouse anywhere
        if(Greenfoot.mouseClicked(this) ) {
        AvoiderWorld world = new AvoiderWorld();
        Greenfoot.setWorld(world);
        }
}
```

Compile and Run. Δοκιμάστε το παιχνίδι σας. Τι συμβαίνει;

Σκορ!

Το παιχνίδι γίνεται πιο ενδιαφέρον αν υπάρχει σκορ και διαφορετικά επίπεδα δυσκολίας. Θα ξεκινήσουμε προσθέτοντας τη δυνατότητα καταγραφής του σκορ.

Αυτό θα πρέπει να εισάγουμε την Counter class του Greenfoot. Για να την εισάγουμε, κάνουμε τα εξής:

Στο μενού Edit του Greenfoot επιλέξτε **Import class…** και από τις επιλογές που εμφανίζονται επιλέξτε την κλάση Counterκαι πατήστε OK.

Τώρα θα δείτε την κλάση Counter στη λίστα των Actors, μαζί με τις κλάσεις Avatarκαι Enemy.

Θέλουμε το σκορ να εμφανίζεται άμεσα στην οθόνη, με την εκκίνηση του παιχνιδιού.

Μπείτε στον κώδικα του AvoiderWorld class, βρείτε την prepare( ) method και αλλάξτε την ως εξής:

```
private void prepare() {
        Avatar avatar = new Avatar();
        addObject(avatar, 287, 232);
        scoreBoard = new Counter("Score: ");
        addObject(scoreBoard, 70, 20);
}
```

Οι δύο πρώτες γραμμές υπάρχουν ήδη, προσθέστε τις επόμενες δύο, οι οποίες θα εμφανίζουν το σκορ μας στο πάνω αριστερά μέρος της οθόνης. Επίσης πρέπει να ορίσουμε τη μεταβλητή scoreboard στην αρχή της κλάσης. Γράψτε μέσα στον κώδικα του AvoiderWorld όπως φαίνεται παρακάτω:

Στην αρχή, κάτω από τη φράση → public class AvoiderWorld extends World

Ορίστε τη μεταβλητή scoreboard γράφοντας τον παρακάτω κώδικα.

private Counter scoreBoard;

Compile and Run. Δοκιμάστε το παιχνίδι σας. Τι συμβαίνει;

Πως θα ανεβαίνει το σκορ;

Πρέπει να καλέσουμε την setValue( ) έτσι ώστε η μεταβλητή scoreBoard να αυξάνεται όσο περνάει ο χρόνος. Μπορούμε να την τοποθετήσουμε στην AvoiderWorld sub class. Στόχος είναι να παίρνει πόντους ο παίχτης για κάθε εχθρό που δημιουργείται.

Αλλάξτε λοιπόν τον κώδικά μέσα στην AvoiderWorld subclass ως εξής, προσθέτοντας τις δύο τελευταίες γραμμές:

```
public void act() {
        // Randomly add enemies to the world
        if(Greenfoot.getRandomNumber(1000) < 20) {
        Enemy e = new Enemy();
        addObject( e, Greenfoot.getRandomNumber(getWidth()-20)+10, -30);
        // Give us one point for every new enemy
        scoreBoard.setValue(scoreBoard.getValue() + 1);
}
}
```

Αποθηκεύστε το παιχνίδι σας.

Παρατηρήστε τον κώδικα που μόλις προσθέσατε. Μπορείτε να εξηγήσετε τι κάνει;

Compile and Run. Τρέξτε το παιχνίδι. Τι παρατηρείτε;

Αύξηση βαθμού δυσκολίας

Η δυσκολία του παιχνιδιού μπορεί να διαμορφωθεί, αλλάζοντας το ρυθμό δημιουργίας και εμφάνισης των εχθρών (enemies) καθώς και την ταχύτητά τους. Για να το πετύχουμε αυτό, πρέπει να οριστούν οι μεταβλητές enemySpawnRateandenemySpeed οι οποίες θα ρυθμίζουν το ρυθμό εμφάνισης εχθρών και την ταχύτητά τους.

Αλλάξτε την αρχή του κώδικα της κλάσης AvoiderWorld προσθέτοντας τις παρακάτω μεταβλητές, όπως φαίνεται στην εικόνα:

```
2
3  public class AvoiderWorld extends World
4  {
6      private Counter scoreBoard;
7      private int enemySpawnRate = 20;
8      private int enemySpeed = 1;
9
```

Θέλουμε να αυξάνεται ο βαθμός δυσκολίας καθώς το σκορ ανεβαίνει. Για να το πετύχουμε αυτό πρέπει να προσθέσουμε την παρακάτω methodστον κώδικα της κλάσης AvoiderWorld.

```
private void increaseLevel() {
        int score = scoreBoard.getValue();
        if( score >nextLevel ) {
        enemySpawnRate += 2;
        enemySpeed++;
        nextLevel += 100;
}
}
```

Όπως βλέπετε, έχουμε μια νέα μεταβλητή, την nextLevel, μέσα στη method increaseLevel(), και πρέπει να την ορίσουμε στην αρχή του κώδικα της AvoiderWorld class.

Ορίστε την στην αρχή, μαζί με τις άλλες δύο μεταβλητές →private int nextLevel = 100;

Όπως φαίνεται από τον κώδικα στην increaseLevel(), αυξάνουμε τη μεταβλητή enemySpawnRate και τη μεταβλητή enemySpeed όσο το σκορ του παίκτη ανεβαίνει. Το τελευταίο πράγμα που πρέπει να κάνουμε είναι να χρησιμοποιήσουμε τις μεταβλητές enemySpawnRate και enemySpeed στη δημιουργία των enemies και να καλέσουμε την increaseLevel() από την act() method στην κλάση AvoiderWorld.

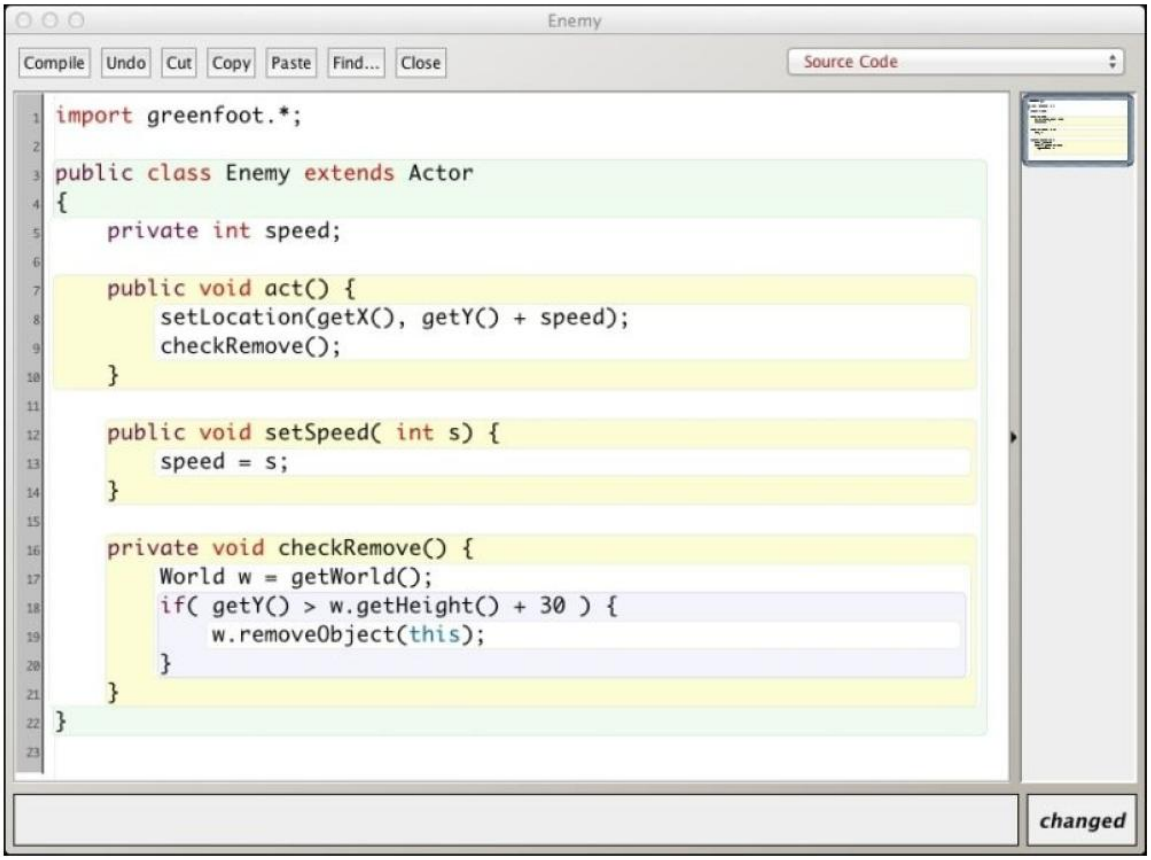
Αλλάξτε λοιπόν τον κώδικα της act()method όπως παρακάτω:

```
public void act() {
        // Randomly add enemies to the world
        if(Greenfoot.getRandomNumber(1000) <enemySpawnRate) {
        Enemy e = new Enemy();
        e.setSpeed(enemySpeed);
        addObject( e, Greenfoot.getRandomNumber(getWidth()-20)+10, -30);
        // Give us some points for facing yet another enemy
        scoreBoard.setValue(scoreBoard.getValue() + 1);
        }
        increaseLevel();
}
```

Κάντε Compile. Τι παρατηρείτε; Βλέπετε ένα μήνυμα σφάλματος;

Στηνact()method, χρησιμοποιούμε τη γραμμή κώδικα **e.setSpeed(enemySpeed);** Για να αλλάξουμε την ταχύτητα των enemies, όμως δεν έχουμε κάνει καμία αναφορά αυτής της methodμέσα στην Enemy class. Άρα χρειάζεται να αλλάξουμε λίγο την Enemyclass για να αλλάζει η ταχύτητα.

Αλλάξτε τον κώδικα μέσα στην Enemyclass όπως φαίνεται παρακάτω:



Όπως βλέπετε κάναμε λίγες αλλαγές στην Enemy class. Προσθέσαμε την setSpeed() method, η οποία δέχεται μία ακέραια (integer)παράμετρο και την χρησιμοποιεί για να αλλάξει την ταχύτητα μέσα από τη μεταβλητή speedπου έχουμε ορίσει στην αρχή της κλάσης.

Στηνact() method, χρησιμοποιούμε την τιμή της μεταβλητής speedvariable στην setLocation( ), και έτσι αυξάνεται η ταχύτητα στην κάθετη (y) συντεταγμένη.

Αποθηκεύστε! Αποθηκεύστε!Αποθηκεύστε!

Compile and run and enjoy your new game!

Άσκηση 1

1. Με την προϋπόθεση ότι έχετε αποθηκεύσει, ότι όλα είναι εντάξει, και το παιχνίδι μας τρέχει κανονικά, κάντε το εξής:

→Αποθηκεύστε το παιχνίδι σας με το όνομα «AvoiderGame_Askisi»

→Αλλάξτετορυθμόμετονοποίο γεννιούνται νέοι εχθροί.

→Αλλάξτε την ταχύτητα με την οποία τρέχουν οι εχθροί

→Προσθέστε turn(5); στην act() method του Enemy class. Compile and run. Τι συμβαίνει;

→Δοκιμάστε διαφορετικές τιμές αντί για 5 μέσα στην turn().Τι συμβαίνει;