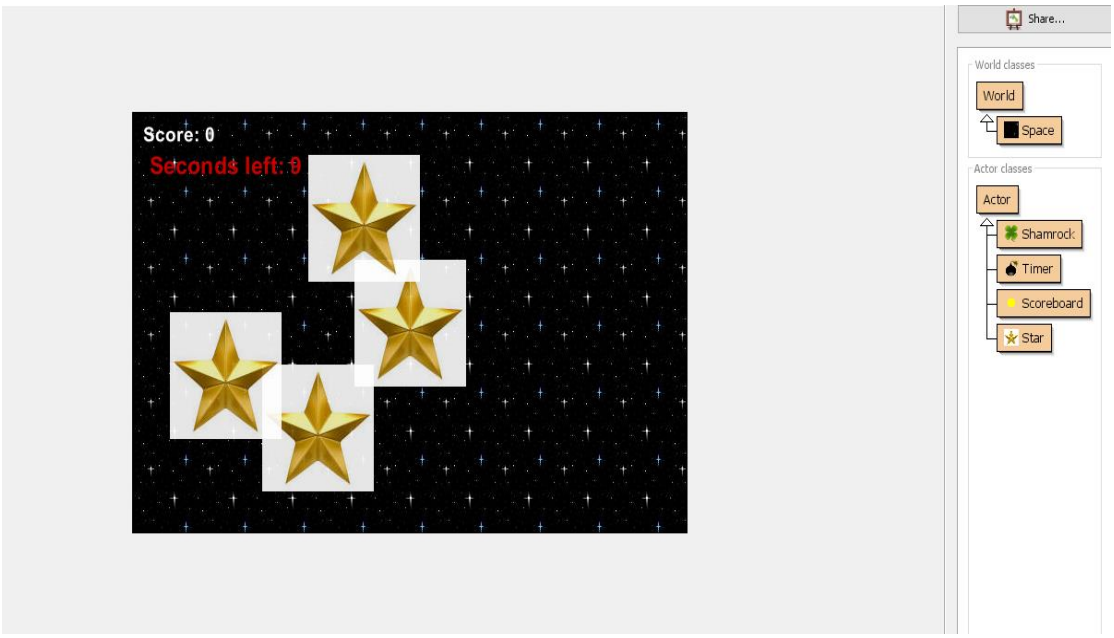
Άσκηση 6η

Δημιουργήστε τις κλάσεις και τις υποκλάσεις όπως απεικονίζονται παρακάτω, γράψτε τους κώδικες στους αντίστοιχους editor και «τρέξτε» το πρόγραμμα….

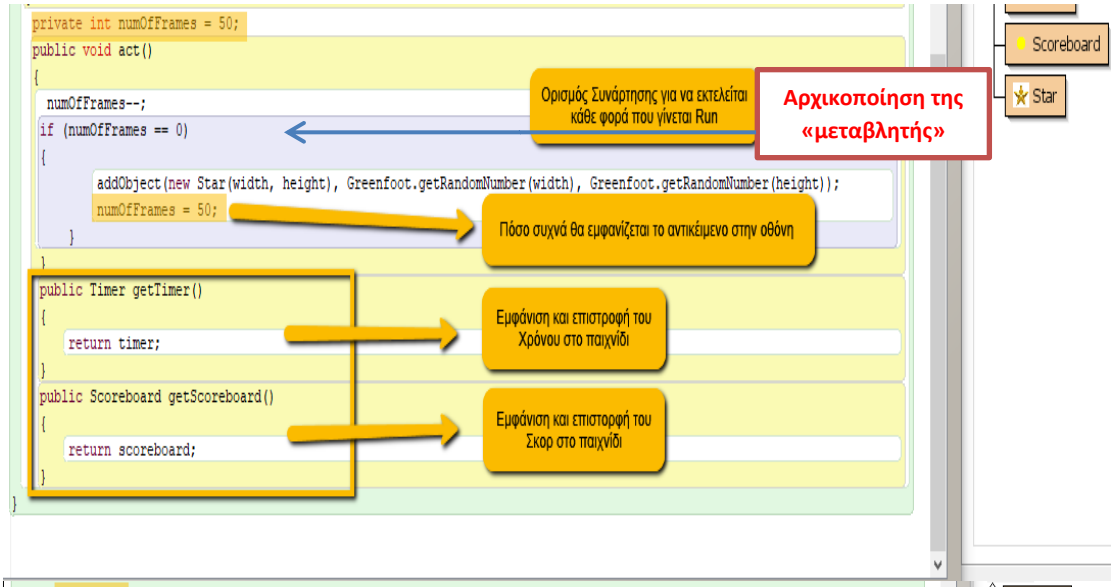


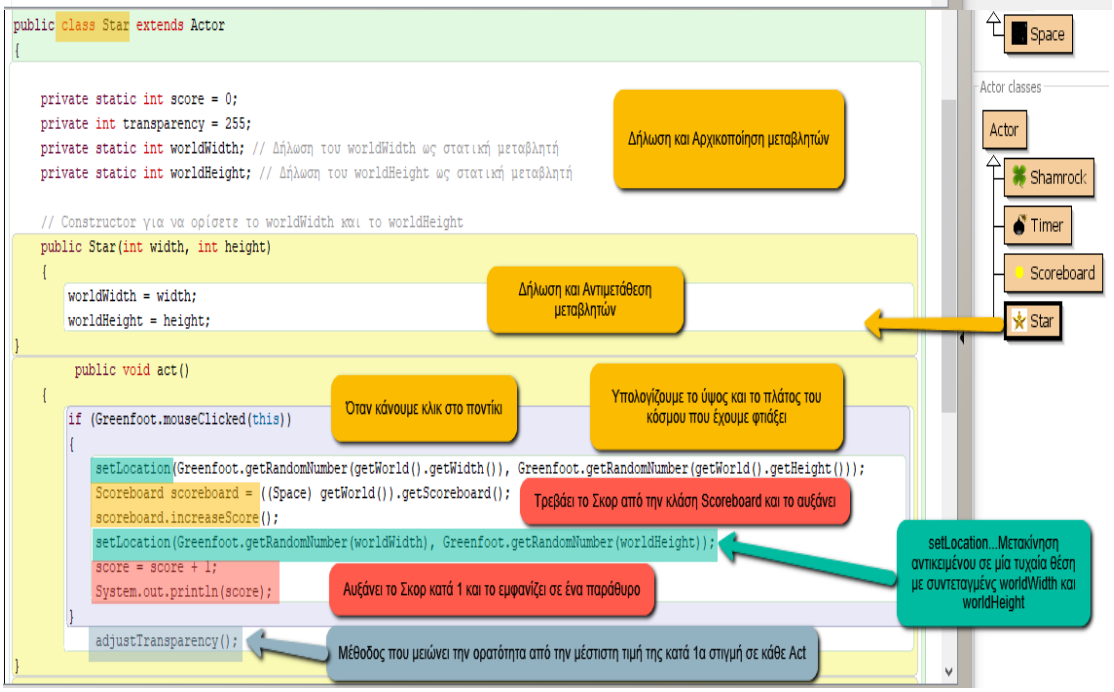

```
import greenfoot.*;

/**
 * Write a description of class Space here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Space extends World
{
    private static int width = 600;
    private static int height = 400;
    private Timer timer;
    private Scoreboard scoreboard;
    /**
     * Constructor for objects of class Space.
     *
     */
    public Space()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(600, 400, 1);
        addObject(new Star(width, height), 300, 200);
        addObject(new Star(width, height), 200, 300);
        addObject(new Star(width, height), 100, 250);
        addObject(new Star(width, height), 250, 100);
        timer = new Timer(); // Δημιουργήστε το αντικείμενο Timer
        addObject(timer, 100, 50); // Προσθέστε το στον κόσμο στις συγκεκριμένες συντεταγμένες
        addObject(timer, getWidth() - 50, 20);
        scoreboard = new Scoreboard();
        addObject(scoreboard, 50, 20);
    }
```

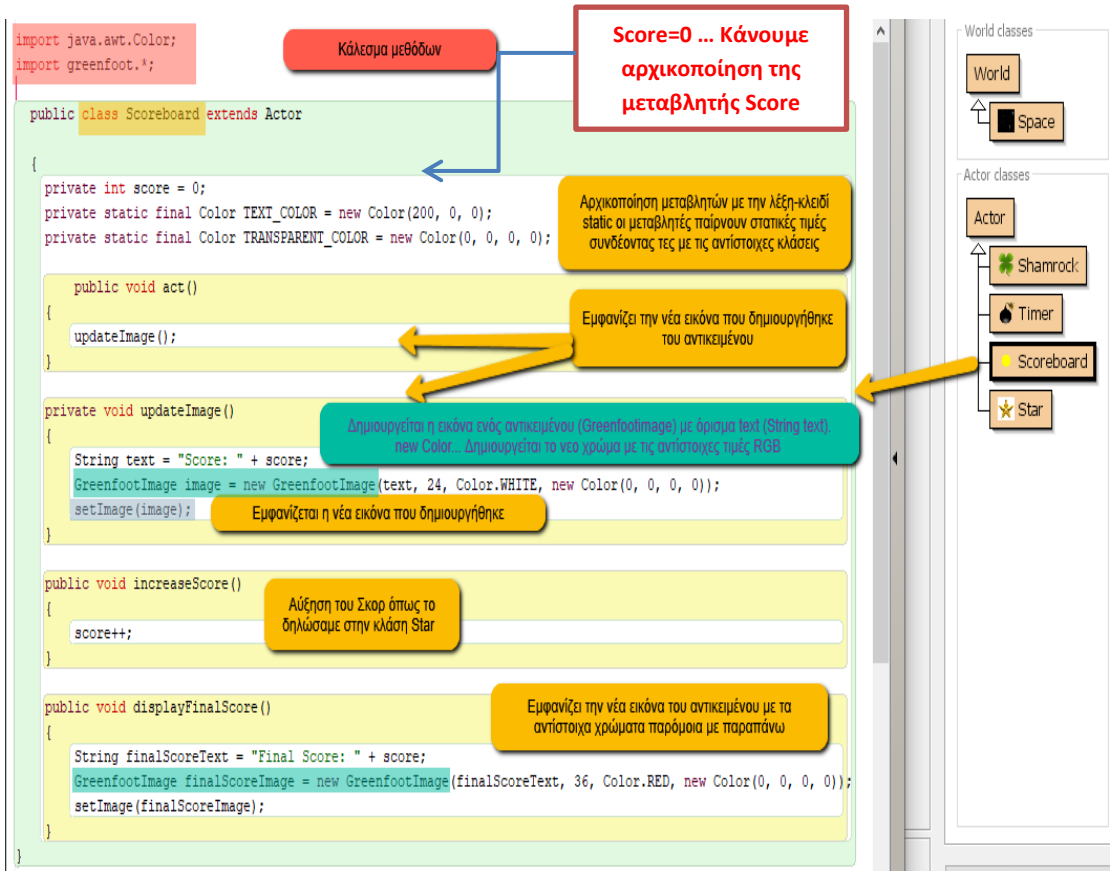
Ορίζουμε τις ιδιοτικές μεταβλητές Static ορίζουμε την μεταβλητή με την ίδια την κλάση

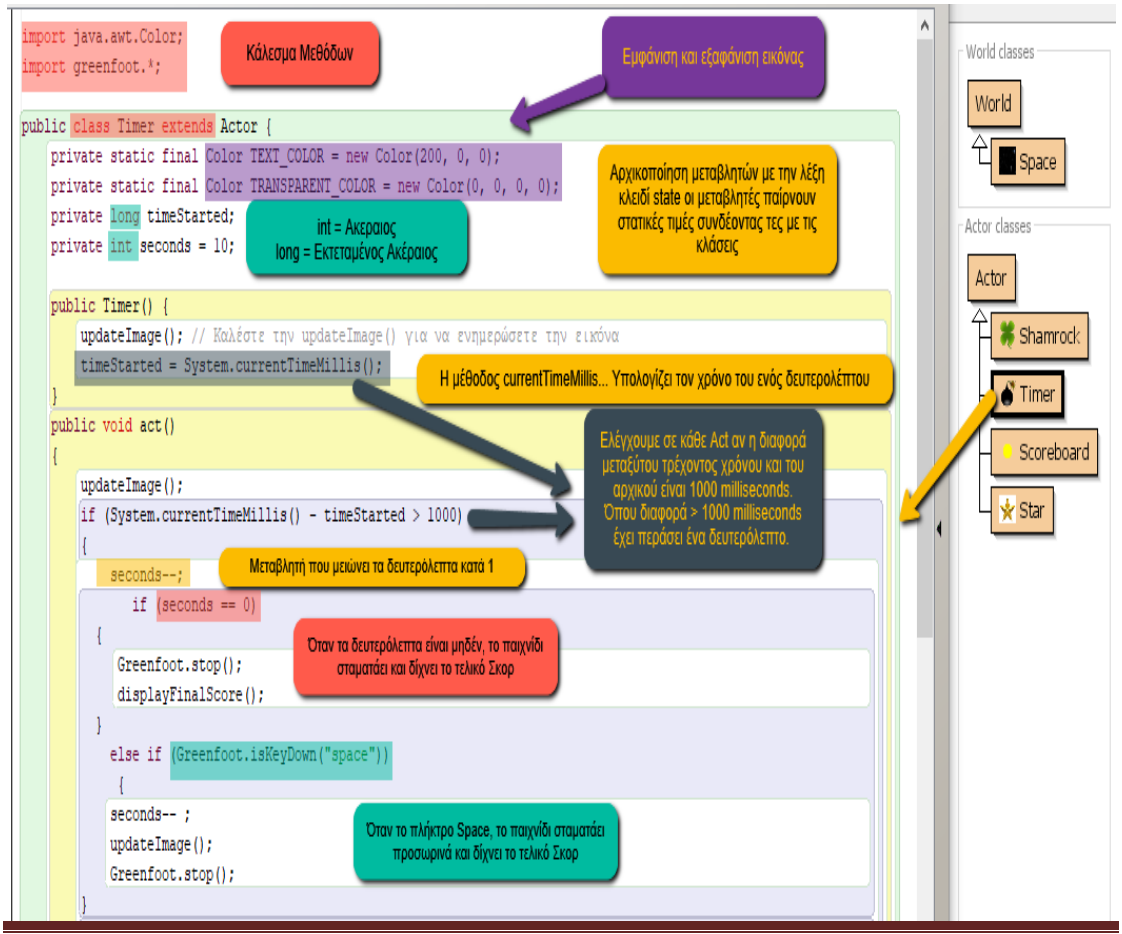
Δημιουργούμε τα αντικείμενα μας στο χώρο του παιχνιδιου με συγκεκριμένες συντεταγμένες

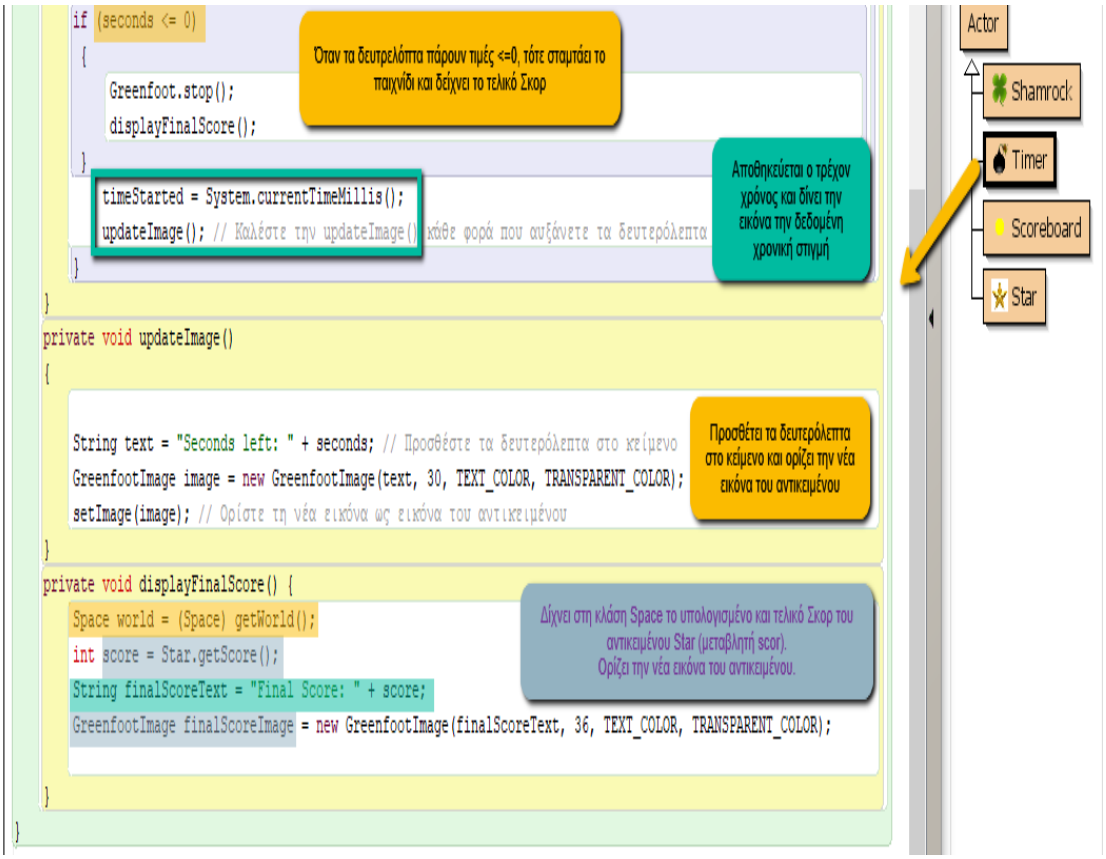

private int numOfFrames = 50;
public void act()
{
numOfFrames--;
if (numOfFrames == 0)
{
addObject(new Star(width, height), Greenfoot.getRandomNumber(width), Greenfoot.getRandomNumber(height));
numOfFrames = 50;
}
}
public Timer getTimer()
{
return timer;
}
public Scoreboard getScoreboard()
{
return scoreboard;
}
}
Ορισμός Συνάρτησης για να εκτελείται κάθε φορά που γίνεται Run
Αρχικοποίηση της «μεταβλητής»
Πόσο συχνά θα εμφανίζεται το αντικείμενο στην οθόνη
Εμφάνιση και επιστροφή του Χρόνου στο παιχνίδι
Εμφάνιση και επιστροφή του Σκορ στο παιχνίδι
Scoreboard
Star

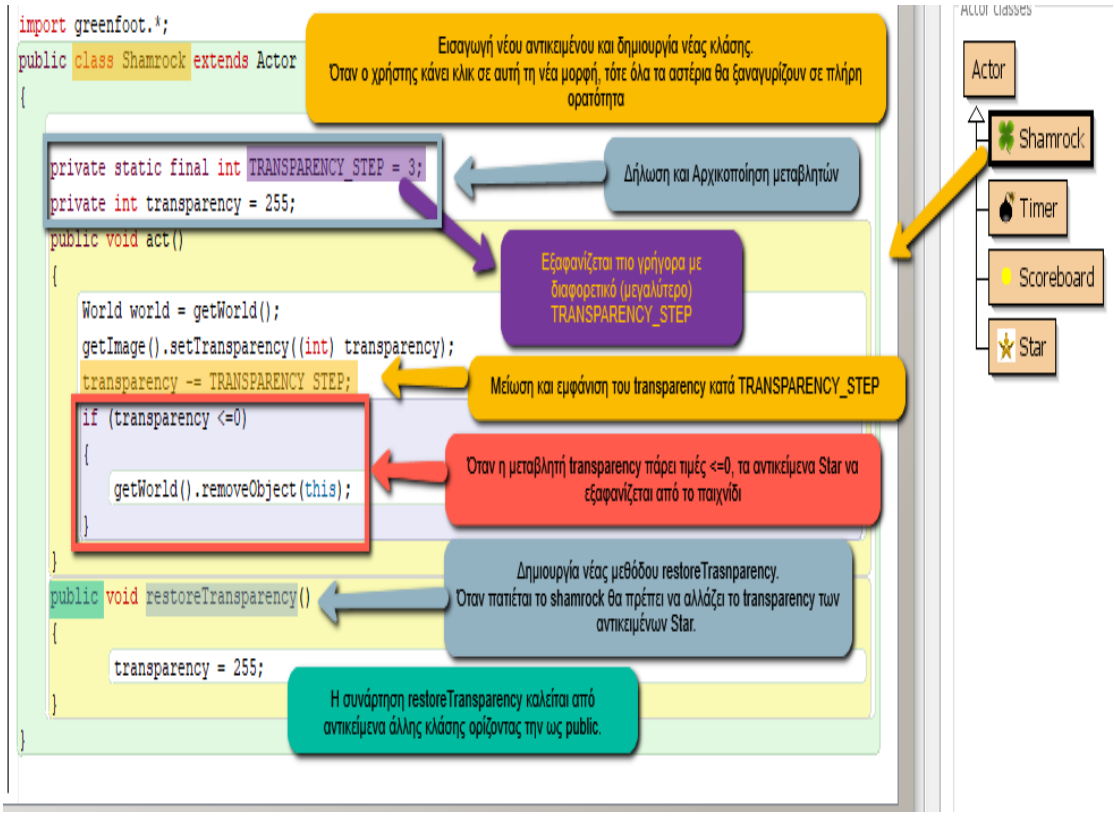

public class Star extends Actor
{
private static int score = 0;
private int transparency = 255;
private static int worldWidth; // Δήλωση του worldWidth ως στατική μεταβλητή
private static int worldHeight; // Δήλωση του worldHeight ως στατική μεταβλητή
// Constructor για να ορίσετε το worldWidth και το worldHeight
public Star(int width, int height)
{
worldWidth = width;
worldHeight = height;
}
public void act()
{
if (Greenfoot.mouseClicked(this))
{
setLocation(Greenfoot.getRandomNumber(getWorld().getWidth()), Greenfoot.getRandomNumber(getWorld().getHeight()));
Scoreboard scoreboard = ((Space) getWorld()).getScoreboard();
scoreboard.increaseScore();
setLocation(Greenfoot.getRandomNumber(worldWidth), Greenfoot.getRandomNumber(worldHeight));
score = score + 1;
System.out.println(score);
}
adjustTransparency();
}
Δήλωση και Αρχικοποίηση μεταβλητών
Δήλωση και Αντιμετάθεση μεταβλητών
Όταν κάνουμε κλικ στο ποντίκι
Υπολογίζουμε το ύψος και το πλάτος του κόσμου που έχουμε φτιάξει
Τρεβάει το Σκορ από την κλάση Scoreboard και το αυξάνει
setLocation...Μετακίνηση αντικειμένου σε μία τυχαία θέση με συντεταγμένες worldWidth και worldHeight
Αυξάνει το Σκορ κατά 1 και το εμφανίζει σε ένα παράθυρο
Μέθοδος που μειώνει την ορατότητα από την μέστιστη τιμή της κατά 1α στιγμή σε κάθε Act
Space
Actor classes
Actor
Shamrock
Timer
Scoreboard
Star


private void adjustTransparency()
{
getImage().setTransparency(transparency);
if (transparency <=0)
{
transparency = 0;
}
else
{transparency = transparency - 1;
}
}
public static int getScore()
{
return score;
}
}
Κατασκευάζουμε την μέθοδο που θα μειώνει τη διαφάνεια της εικόνας κατά 1 Act
Εφαρμόζει την εικόνα του αντικειμένου Star με την νέα τιμή της
Ελέγχει και Εξασφαλίζει ότι δεν θα πάρει τιμή μικρότερη από 0.
Το GreenFoot Δεν δέχεται αρνητικές τιμές για διαφάνεια εικόνας.
Προγραμματίζουμε το Star να μειώνει την διαφάνεια της εικόνας κατά 1 Act
Επιστροφή του Σκορ στην κάθε δεδομένη χρονική στιγμή
Shamrock
Timer
Scoreboard
Star


Κάλεσμα μεθόδων
Score=0 ... Κάνουμε αρχικοποίηση της μεταβλητής Score
Αρχικοποίηση μεταβλητών με την λέξη-κλειδί static οι μεταβλητές παίρνουν στατικές τιμές συνδέοντας τες με τις αντίστοιχες κλάσεις
Εμφανίζει την νέα εικόνα που δημιουργήθηκε του αντικειμένου
Δημιουργείται η εικόνα ενός αντικειμένου (GreenfootImage) με όρισμα text (String text), new Color... Δημιουργείται το νεο χρώμα με τις αντίστοιχες τιμές RGB
Εμφανίζεται η νέα εικόνα που δημιουργήθηκε
Αύξηση του Σκορ όπως το δηλώσαμε στην κλάση Star
Εμφανίζει την νέα εικόνα του αντικειμένου με τα αντίστοιχα χρώματα παρόμοια με παραπάνω
World classes
World
Space
Actor classes
Actor
Shamrock
Timer
Scoreboard
Star


Κάλεσμα Μεθόδων
Εμφάνιση και εξαφάνιση εικόνας
Αρχικοποίηση μεταβλητών με την λέξη κλειδί state οι μεταβλητές παίρνουν στατικές τιμές συνδέοντας τες με τις κλάσεις
int = Ακεραιος
long = Εκτεταμένος Ακέραιος
Η μέθοδος currentTimeMillis... Υπολογίζει τον χρόνο του ενός δευτερολέπτου
Ελέγχουμε σε κάθε Act αν η διαφορά μεταξύτου τρέχοντος χρόνου και του αρχικού είναι 1000 milliseconds. Όπου διαφορά > 1000 milliseconds έχει περάσει ένα δευτερόλεπτο.
Μεταβλητή που μειώνει τα δευτερόλεπτα κατά 1
Όταν τα δευτερόλεπτα είναι μηδέν, το παιχνίδι σταματάει και δίχνει το τελικό Σκορ
Όταν το πλήκτρο Space, το παιχνίδι σταματάει προσωρινά και δίχνει το τελικό Σκορ
World classes
World
Space
Actor classes
Actor
Shamrock
Timer
Scoreboard
Star


if (seconds <= 0)
{
Greenfoot.stop();
displayFinalScore();
}
timeStarted = System.currentTimeMillis();
updateImage(); // Καλέστε την updateImage() κάθε φορά που αυξάνετε τα δευτερόλεπτα
}
}
private void updateImage()
{
String text = "Seconds left: " + seconds; // Προσθέστε τα δευτερόλεπτα στο κείμενο
GreenfootImage image = new GreenfootImage(text, 30, TEXT_COLOR, TRANSPARENT_COLOR);
setImage(image); // Ορίστε τη νέα εικόνα ως εικόνα του αντικειμένου
}
private void displayFinalScore() {
Space world = (Space) getWorld();
int score = Star.getScore();
String finalScoreText = "Final Score: " + score;
GreenfootImage finalScoreImage = new GreenfootImage(finalScoreText, 36, TEXT_COLOR, TRANSPARENT_COLOR);
}
}
Όταν τα δευτρελόπτα πάρουν τιμές <=0, τότε σταμτάει το παιχνίδι και δείχνει το τελικό Σκορ
Αποθηκεύεται ο τρέχον χρόνος και δίνει την εικόνα την δεδομένη χρονική στιγμή
Προσθέτει τα δευτερόλεπτα στο κείμενο και ορίζει την νέα εικόνα του αντικειμένου
Δίχνει στη κλάση Space το υπολογισμένο και τελικό Σκορ του αντικειμένου Star (μεταβλητή scor).
Ορίζει την νέα εικόνα του αντικειμένου.
Actor
Shamrock
Timer
Scoreboard
Star


import greenfoot.*;
public class Shamrock extends Actor
{
private static final int TRANSPARENCY_STEP = 3;
private int transparency = 255;
public void act()
{
World world = getWorld();
getImage().setTransparency((int) transparency);
transparency -= TRANSPARENCY_STEP;
if (transparency <=0)
{
getWorld().removeObject(this);
}
}
public void restoreTransparency()
{
transparency = 255;
}
}
Εισαγωγή νέου αντικειμένου και δημιουργία νέας κλάσης.
Όταν ο χρήστης κάνει κλικ σε αυτή τη νέα μορφή, τότε όλα τα αστέρια θα ξαναγυρίζουν σε πλήρη ορατότητα
Δήλωση και Αρχικοποίηση μεταβλητών
Εξαφανίζεται πιο γρήγορα με διαφορετικό (μεγαλύτερο) TRANSPARENCY_STEP
Μείωση και εμφάνιση του transparency κατά TRANSPARENCY_STEP
Όταν η μεταβλητή transparency πάρει τιμές <=0, τα αντικείμενα Star να εξαφανίζεται από το παιχνίδι
Δημιουργία νέας μεθόδου restoreTrasnparency.
Όταν πατέται το shamrock θα πρέπει να αλλάζει το transparency των αντικειμένων Star.
Η συνάρτηση restoreTransparency καλείται από αντικείμενα άλλης κλάσης ορίζοντας την ως public.
Actor
Shamrock
Timer
Scoreboard
Star

Αναφορές

- Νικολός Δημήτριος, *Σημειώσεις GreenFoot*