

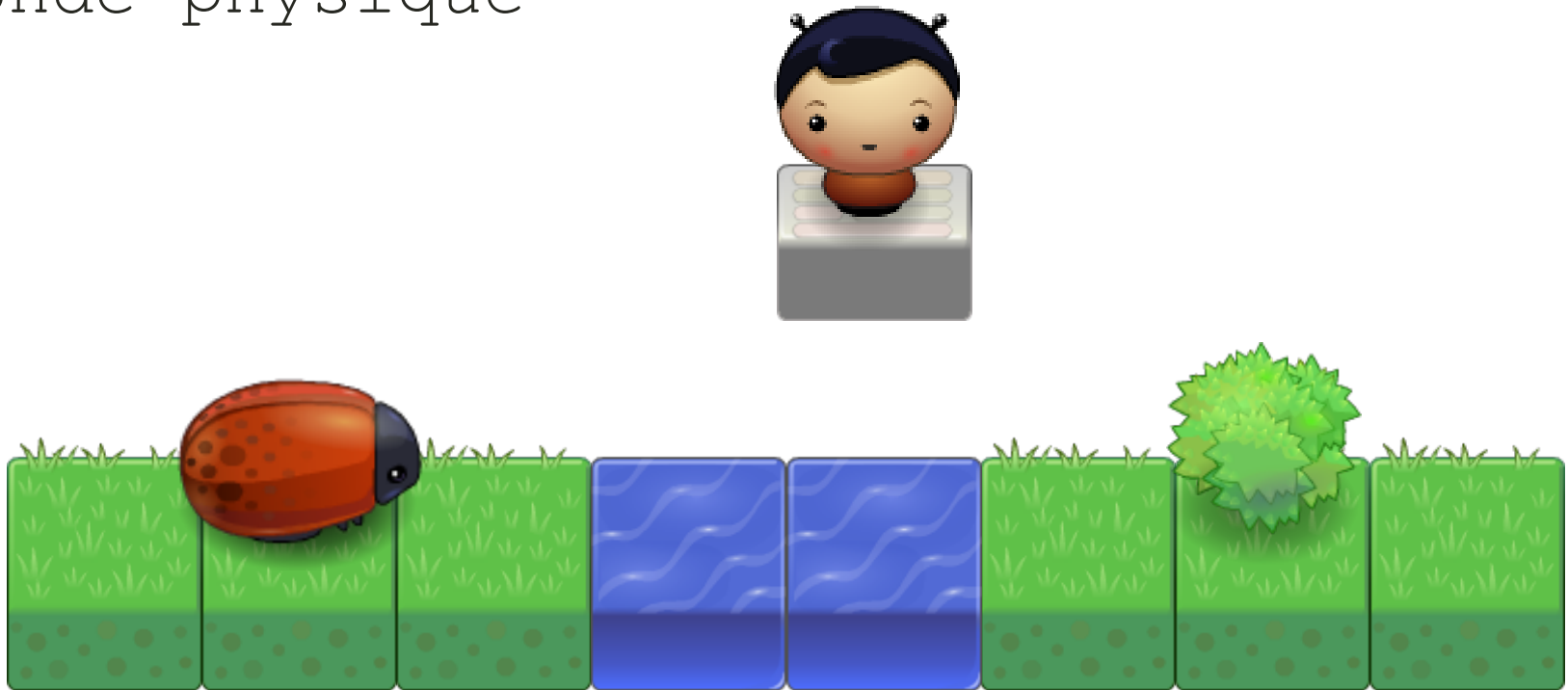
Algoid - ALGEA
www.algoid.net
Créer par Yann Caron

DEVOXXTM
4KIDS



ALGEA :: Actor

- Dans les jeux vidéos, les acteurs sont des objets graphiques
- Qui interagissent entre eux dans un monde physique



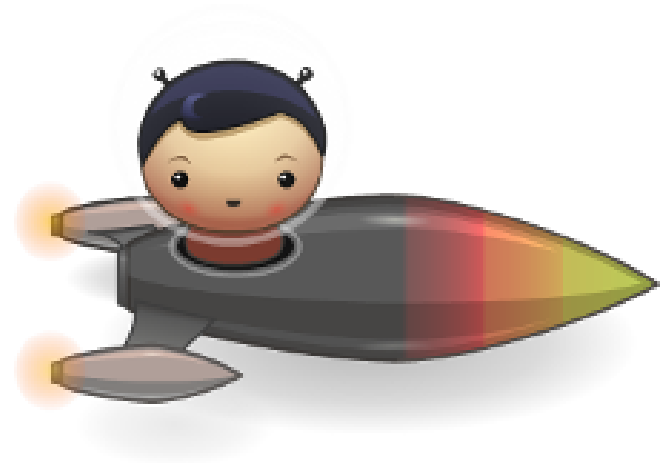
Rappel :: Les objets

- Un acteur (ex : le hero) est un objet
- Il possède des propriétés (des variables)
 - Image
 - Position
 - Hitbox
 - Type physique
- **Qu'est ce qu'un acteur ?**



Rappel :: Les méthodes

- Un acteur a des méthodes (fonctions)
 - Bouger (move)
 - Tourner (rotate)
 - Dire (say)
 - Pousser (applyForce)
 - Lancer (impulse)



- **Qu'est-ce qu'un acteur peu faire ?**

Rappel :: Les évènements

- Et des évènements
 - Toucher (`onContact`)
 - Cliquer (`onTouch`)
- Qu'est-ce qu'il peut se passer avec un acteur ?



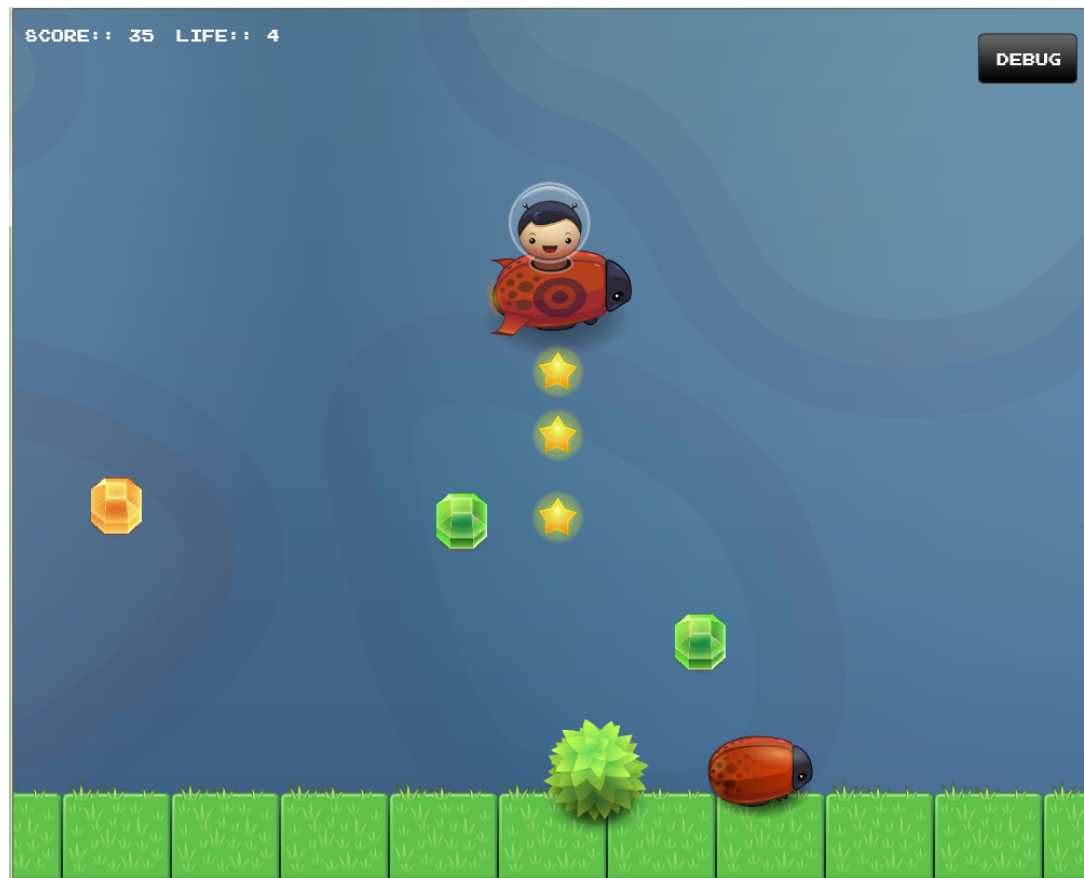
Inspiration

- Jetpack Joyride like



Notre jeu :: CuteJet

Jetpack Joyride like



Notre héros

```
set hero = gameLayer.createActor(  
    "PlanetCute/beetleship");  
  
hero.move(200, 400);
```



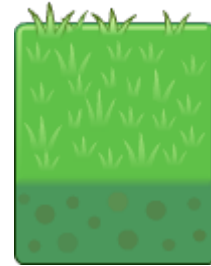
Parler

```
hero.say(font, "Allons y !");
```



Le niveau

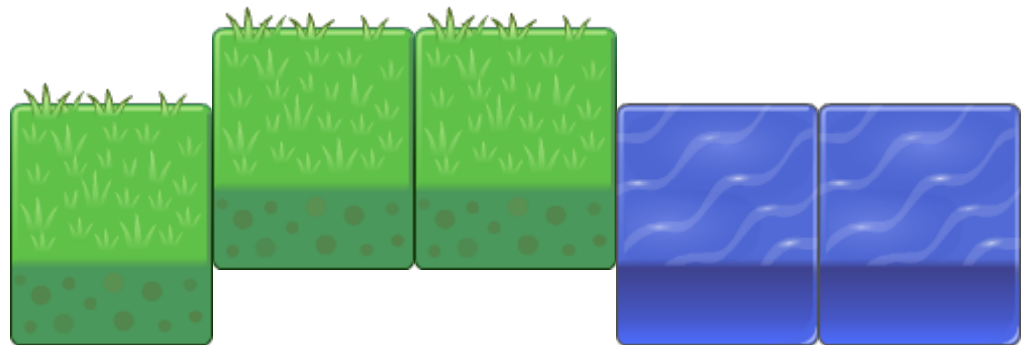
- Dans le menu à gauche, on choisit PlanetCute / Grass Block
- Mais il en faut plusieurs pour dessiner le niveau
- Comment faire ?



Le niveau

```
set x = 0;  
loop(50) {  
    set ground = gameLayer.createActor("PlanetCute/Grass Block");  
    ground.move(x, 50);  
  
    x = x + ground.getWidth();  
}
```

- Une boucle
- Une variable
- On incrémente
- D'autres blocks ?



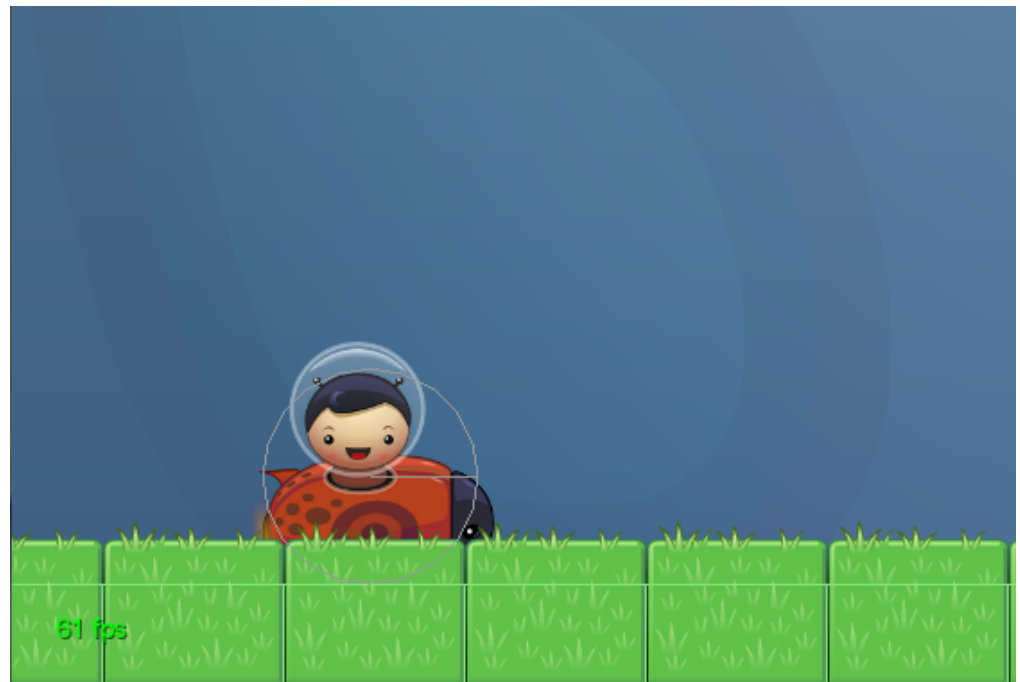
Physique

- Les acteurs interagissent avec le monde
- Type d'acteur
 - `typeStatic()`, `typeDynamique()`, `typeSensor()`
- Static
 - Les murs, le sol, le décor
- Dynamique
 - Notre héros, les ennemies
- Sensor
 - Les gems

Physique :: exemple

- Notre héros :: `hero.typeDynamic().hitCircle(15);`
- Le sol :: `ground.typeStatic().hitSquare(75);`

On appuie sur
le bouton Debug



Animation

- Maintenant on veut que notre héros avance tout seul !
- Et ça marche !!!

```
// le heros avance  
hero.applyForceX(25);
```

Fonction et évènement

- Quand on clique, on veut que notre héros saute

```
set jump = function() {  
    hero.impulseY(15);  
}
```

```
stage.onTap(jump);
```

- Il est vivant ! Il vole !

Et voilà le travail !



Une petite pause ?

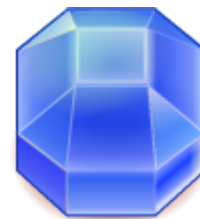


Les gemmes

```
set gem = gameLayer.createActor(  
    "PlanetCute/Gem Green")
```

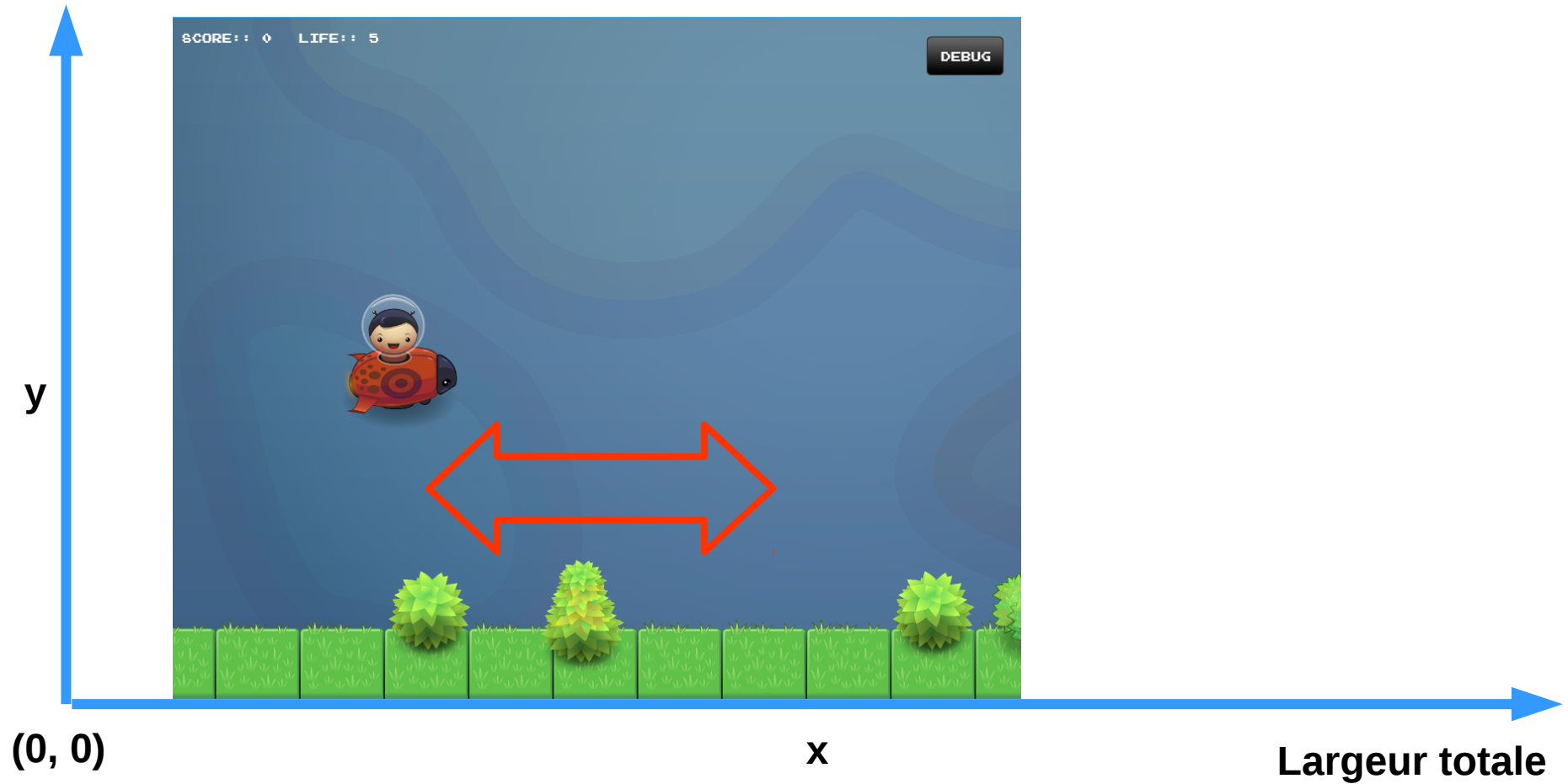
```
gem.scale(0.5)
```

```
gem.move(200, 400)
```



Coordonnées et scrolling

Hauteur totale



Gemme au centre du jeu

```
set x = gameLayer.getWidth() / 2  
set y = gameLayer.getHeight() / 2  
  
gem.move(x, y)
```

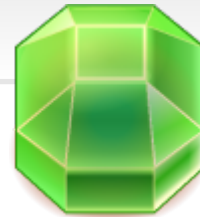


Jeu de hasard

- La fonction `math.random(10)`
 - Choisit un nombre aléatoirement entre 0 et 10. Ou tout autre nombre.
- On veut placer aléatoirement les gemmes
- Comment faire ???

Jeu de hasard

```
set x = math.random(gameLayer.getWidth())  
set y = math.random(gameLayer.getHeight())  
  
gem.move(x, y)
```



Plusieurs gemmes

- On veut créer 25 gemmes dans notre niveau
- Comment faire ???



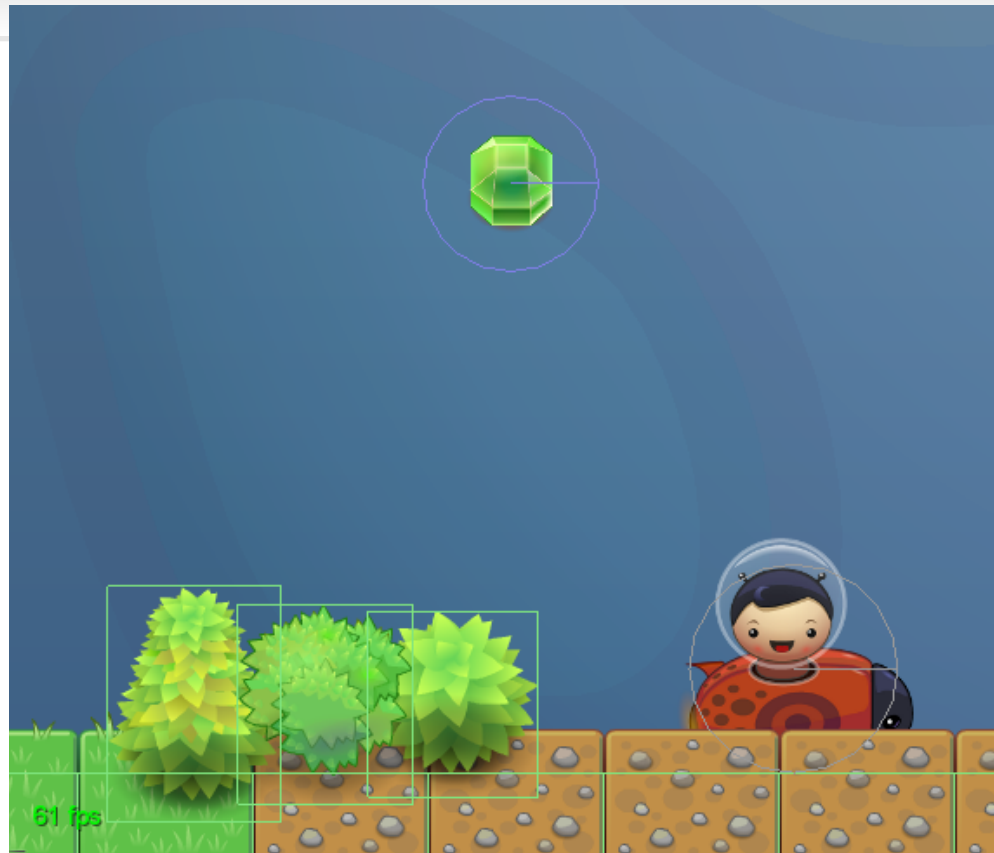
Plusieurs gemmes

```
loop(25) {  
    set x = math.random(gameLayer.getWidth())  
    set y = math.random(gameLayer.getHeight())  
  
    set gem = gameLayer.createActor(  
        "PlanetCute/Gem Orange")  
  
    gem.move(x, y)  
    gem.scale(0.5)  
    gem.typeSensor()  
    gem.hitCircle()  
}
```



Physique des gemmes

```
gem.typeSensor()  
gem.hitCircle()
```



Fonction et événement

- Quand on touche le héro, on veut qu'il gagne des points

```
set winGem = function() {  
    gameLayer.remove(gem)  
}  
  
gem.onContactWith(hero, winGem)
```

- Et ça marche !

Score

- Comment augmenter le score du joueur lorsqu'il touche un gemme ?
- Une variable à disposition

```
score
```

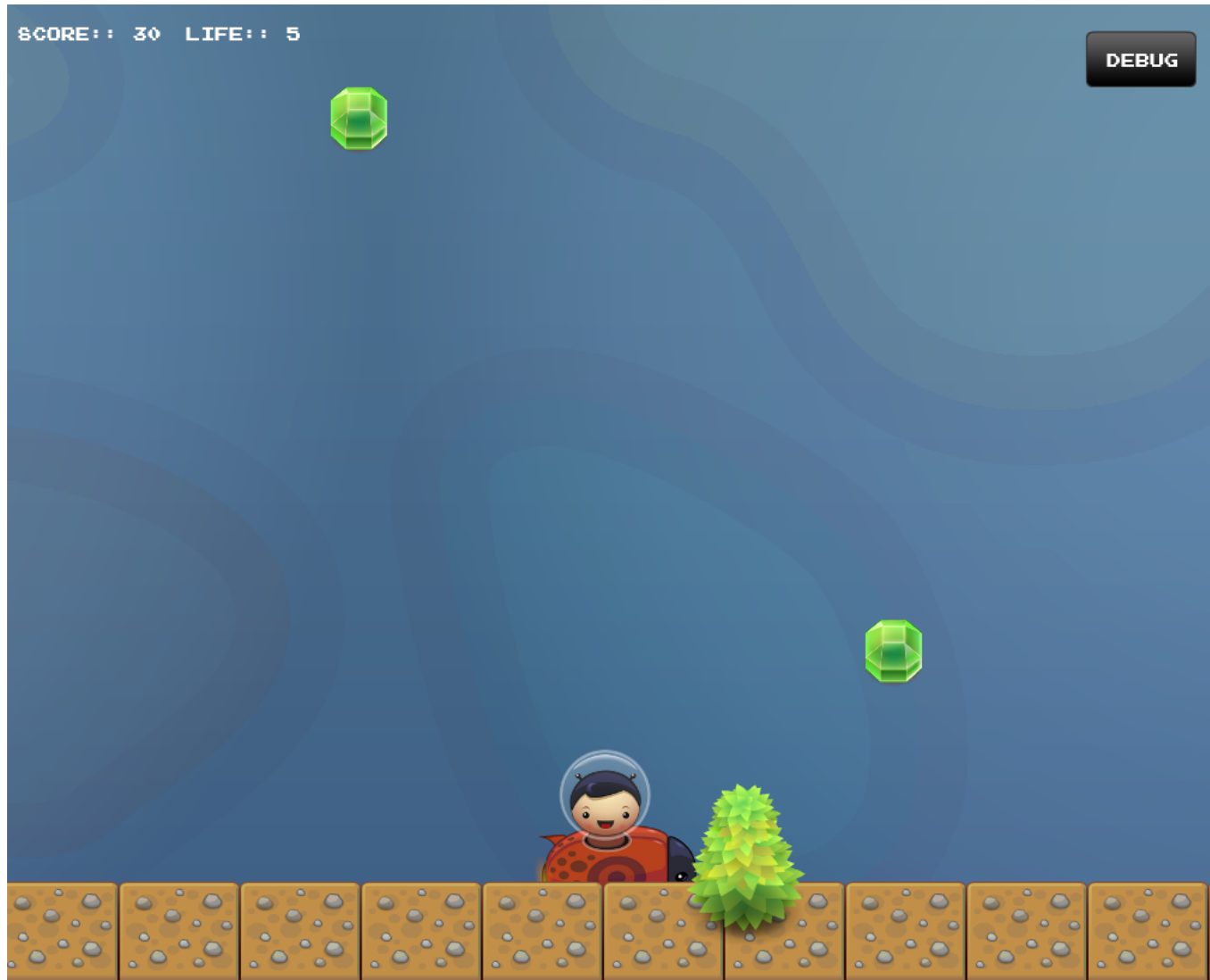
Score

- Une incrémentation

```
score = score + 10;
```

- C'est tout !!!!

Et voilà le travail



Dangers

- Maintenant il faut éviter des cailloux

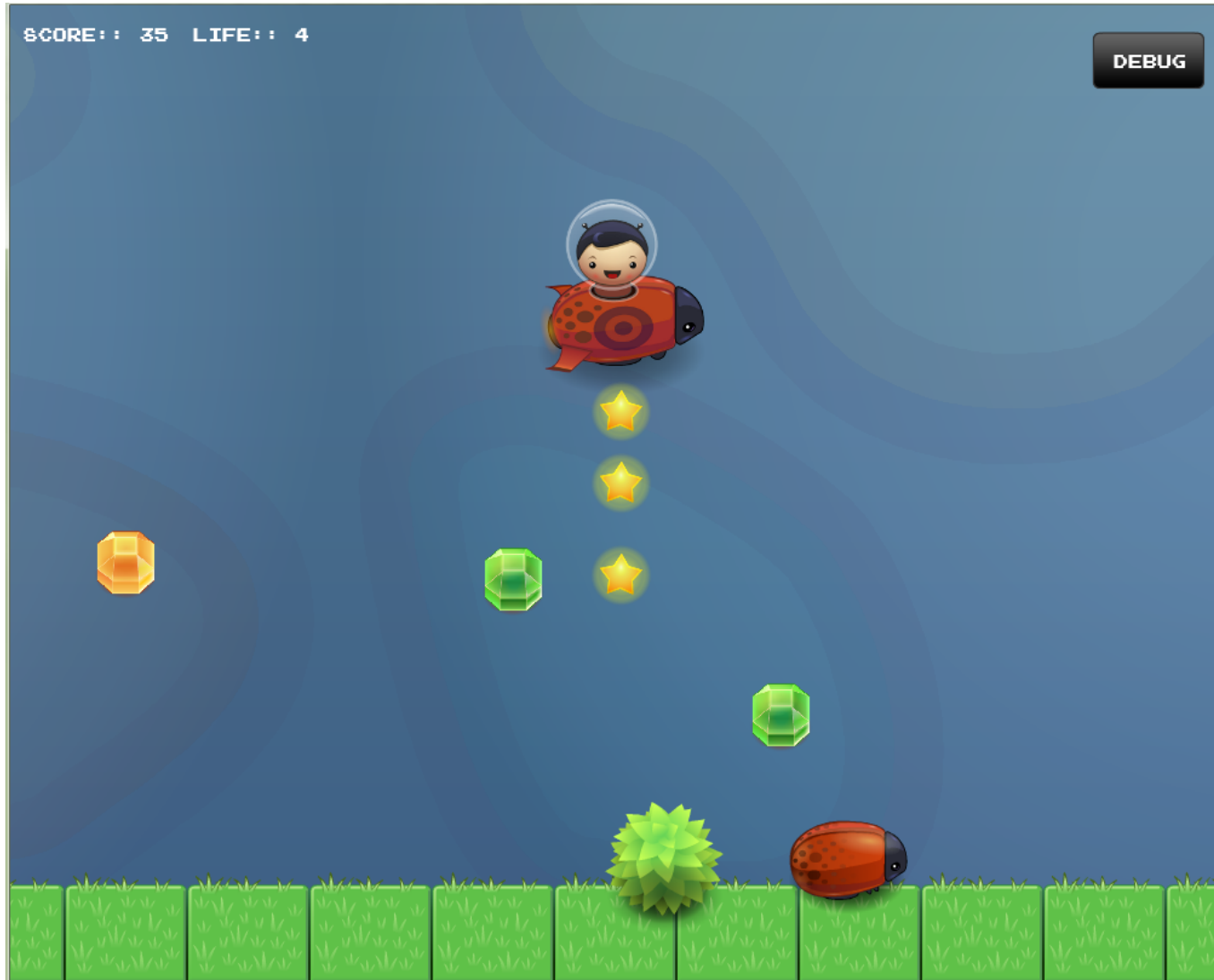
```
set throwRock = function () {  
  set y = math.random(gameLayer.getHeight())  
  set rock = gameLayer.createActor("PlanetCute/Rock", 1500, y)  
  
  rock.typeSensor()  
  rock.hitCircle()  
  rock.setXVelocity(-5)  
  
  rock.onContactWith(hero, gameOver)  
}  
hero.onPulse(throwRock, 1500)
```



Ça y est, on sait faire un jeu !

- Pour aller plus loin, imaginons :
 - Que l'on crée des ennemis
 - Qu'on veuille tirer sur ces ennemis
 - Qu'on fasse marcher les ennemis

Et voilà le travail



Le son

- On crée un son
- Et on le joue en boucle

```
set music = game.createSound("Loop/Vibraphone")  
music.playLoop();
```

- Ou une seul fois:: play
- On le contrôle:: pause, resume, stop

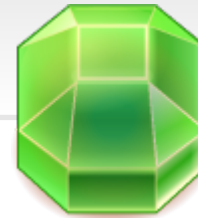
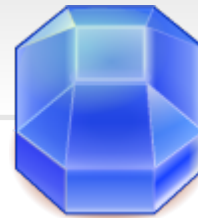
Pour aller plus loin

Les pierres précieuses au hasard

- On veut choisir parmi 3 couleurs de pierres (**tableaux**, **concaténation**)

```
set gems = array {"Gem Blue", "Gem Green", "Gem Orange"};
loop(25) {
  // ....
  set g = math.random(gems.length());

  set gem = gameLayer.createActor("PlanetCute/" .. gems[g]);
  // ....
}
```



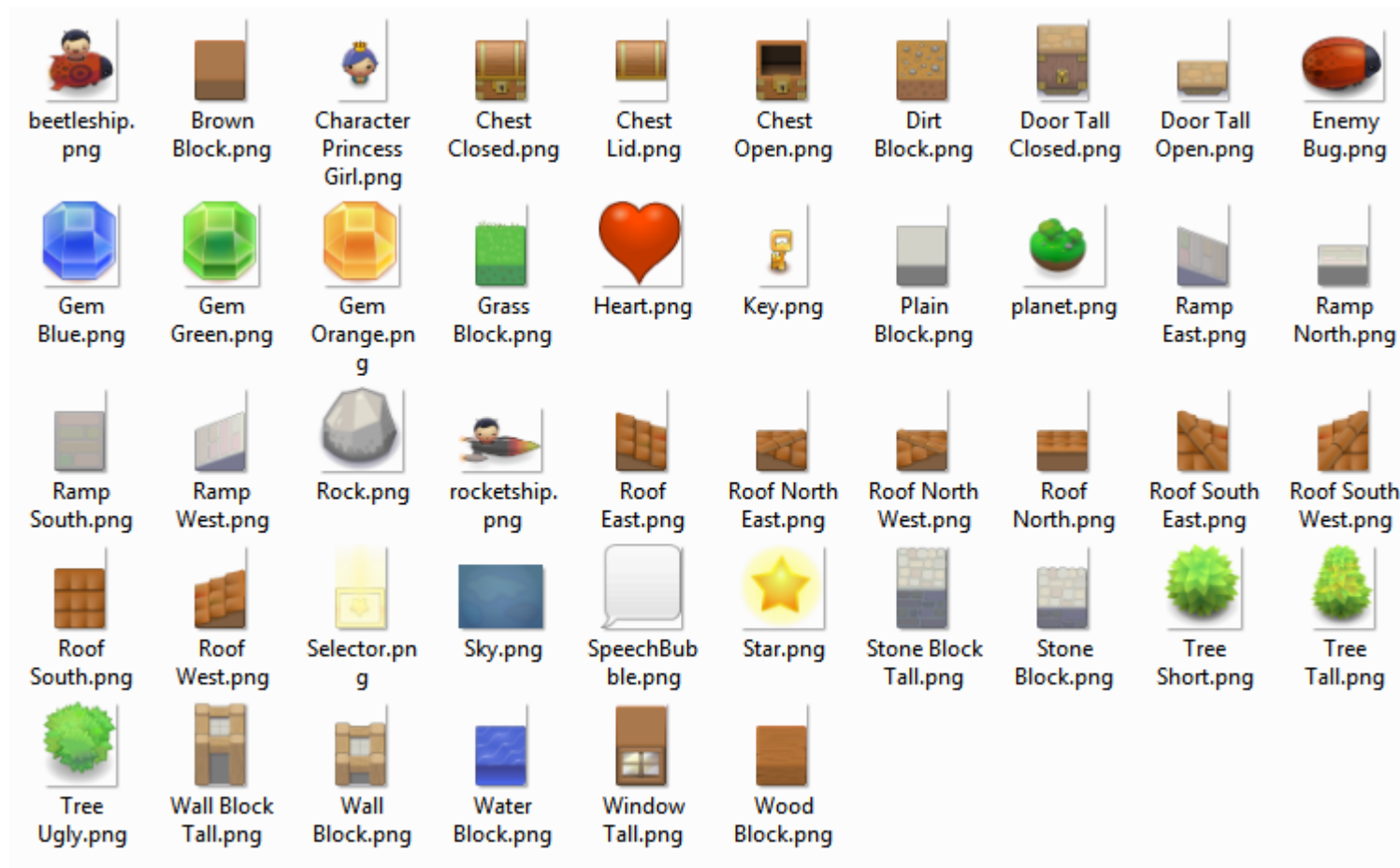
Collision selon l'acteur ou son nom

- Gérer la collision selon ce que l'on touche

```
set contact = function (enemy, other) {  
  if (other.getName() == "PlanetCute/Star") {  
    // Si c'est une étoile  
  } elseif (other.getName() != "PlanetCute/Grass Block") {  
    // Les autres sauf l'herbe  
  }  
}  
  
enemy.onContact(contact);
```

De quoi disposons-nous ?

- Dans le répertoire ALGEA/assets img ou sound



PlanetCut



Merci à Daniel Cook pour son game-
design

<http://www.lostgarden.com/2007/05/>