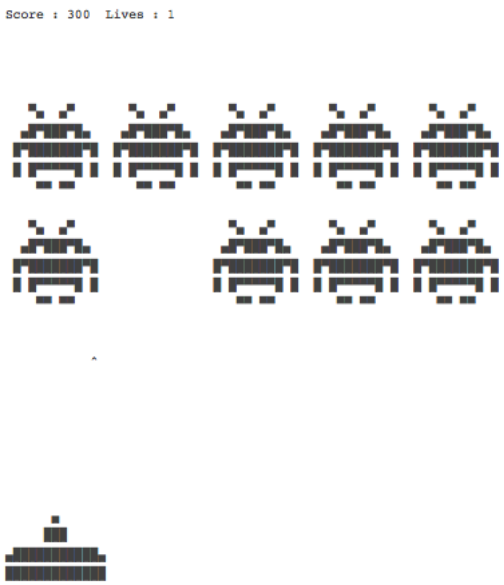
SAÉ 2.01
Space Invader

L'objectif de cette SAE, est de réaliser un jeu de Space Invader en ASCII art.



Les dessins sont tirés de la page

`http://textart4u.blogspot.fr/2014/04/space-invaders-copy-paste-ascii-text-art.html`

1 Prise en main

Un dessin ASCII est la donnée d'un ensemble de chaines, chacune venant avec sa position. Par exemple :

est donné par l'ensemble `("####',1,2), ("°°",5,2),("uu",1,1),('[redacted]',0,0)` . Remarquez qu'on utilise ici des caractères qui ne sont pas sur votre clavier. Je les ai obtenus par copier/coller.

Pour représenter un tel ensemble de chaines positionnées, on a deux classes : `ChainePositionnee` et `EnsembleChaines` :

```
public class ChainePositionnee{
    int x,y;
    String c;
    public ChainePositionnee(int a,int b, String d){x=a; y=b; c=d;}
}
```

```
import java.util.ArrayList;
public class EnsembleChaines {
    ArrayList<ChainePositionnee> chaines;
    public EnsembleChaines(){chaines= new ArrayList<ChainePositionnee
       >(); }

    public void ajouteChaine(int x, int y, String c){
        chaines.add(new ChainePositionnee(x,y,c));}

    public void union(EnsembleChaines e){
        for(ChainePositionnee c : e.chaines)
            chaines.add(c);
    }

}
```

Pour la gestion du dessin, on vous donne une classe Executable dont nul n'est besoin de lire le code et téléchargeable sur Célène. Cette classe orchestrera la gestion du clavier et l'affichage.

Pour que cela fonctionne, vous devez "seulement" coder le jeu, en ayant une classe "GestionJeu" fournissant les méthodes suivantes :

- `public int getHauteur()` et `public int getLargeur()` renvoyant la largeur et la hauteur de la zone de jeu.
- `public void toucheGauche()`, `public void toucheDroite()` et `public void toucheEspace()` ne renvoyant rien et qui sont appelées automatiquement lorsqu'on appuie sur la flèche gauche, droit ou touche espace du clavier.
- `public EnsembleChaines getChaines()` renvoyant les chaines à afficher.
- `public void jouerUnTour()`qui est appelée à fréquence constante et que vous remplirez pour faire évoluer le jeu.

1.1 Mise en place

1. Écrivez le contenu de la classe `GestionJeu` pour obtenir une zone de jeu de largeur 100 et de hauteur 60, affichant la chaine "@@" en position X=0, Y= 30.
2. Modifiez votre classe pour que lorsque l'utilisateur appuie sur la touche espace, le texte "Appui sur la touche espace" soit affiché dans le terminal.
3. Modifiez votre classe pour que cette chaine soit déplacée vers la droite à chaque fois que l'utilisateur appuie sur la flèche droite. Pour cela, votre classe devra contenir un attribut privé "positionX" contenant la position en X de la chaine de caractères.

2 Vaisseau et projectiles

2.1 Vaisseau

Écrivez une classe `Vaisseau` qui :

1. — contient un attribut `posX` de type double.
   — contient une méthode `déplace` prenant en entrée un double (dx) et ajoutant à posX cette valeur.
   — contient une méthode `getEnsembleChaines()` renvoyant l'ensemble de chaines permettant d'afficher en position (posX, 0)
2. Faites afficher ce vaisseau. Pour cela, ajoutez un attribut de type Vaisseau au gestionnaire de jeu.
3. Faites en sorte que lorsqu'on appuie sur les flèches du clavier, le vaisseau se déplace.
4. Ajoutez une méthode `positionCanon` qui renvoie la position en x du canon (on utilisera cette méthode plus tard pour savoir d'où faire partir le projectile).

2.2 Projectile(s)

1. Écrivez une classe `Projectile` qui :
   — contient deux attributs `positionX` et `positionY` de type double.
   — a un constructeur prenant en argument deux entiers : la position en X et la position en Y.
   — contient une méthode `getEnsembleChaines()` renvoyant l'ensemble de chaines(ici, cet ensemble ne contiendra que la chaine positionnée (positionX, positionY,"^").
   — contient une méthode void `evolue()` qui fait incrémenter la position en y de 0.2 .

2. Ajoutez à votre classe `GestionJeu` un attribut de type projectile. Ajoutez le code nécessaire pour que ce projectile s'affiche et évolue à chaque tour.
3. Ajoutez un attribut de type Score à votre gestionnaire de jeu.
4. Modifiez votre code pour que le score soit incrémenté à chaque tour de jeu. Pour cela, ajoutez une méthode `ajoute(int valeur)` à votre classe Score, et ajoutez un appel à `ajoute(1)` à chaque tour de jeu.

2.3 Aliens

— Écrivez une classe `Alien` représentant un Alien (les *super vilains* qu'il faut *dégommer* dans Space Invader). Reprenez par exemple un dessin de la page `http://textart4u.blogspot.fr/2014/04/space-invaders-copy-paste-ascii-text-art.html`.

— Modifiez la classe `GestionJeu` pour qu'elle contienne un `ArrayList` d'Aliens initialisé comme sur le dessin du début du sujet.

— Modifiez la classe Alien pour qu'elle contienne une méthode `evolue` qui : incrémente de 0.1 la position en X 100 tours de suite, puis décrémente la position en Y de 1 puis décrémente la position en X pendant 100 tours de suite, puis décrémente de la position en Y et ainsi de suite.

Comment faire pour que les Aliens soient "animés" ? Implémentez cette fonctionnalité.

3 Collisions

On va maintenant gérer le fait que lorsqu'un balle touche un alien, ce dernier disparaît. Pour cela, on va d'abord devoir être capable de détecter qu'une balle touche un alien, c'est-à-dire que la position (x,y) correspond à la position d'un des caractères représentant l'alien.

On va décomposer par étapes :

3.1 Appartenance

— Créer une méthode `contient` dans la classe `EnsembleChaines` prenant en argument deux entiers x et y et renvoyant true si le point à ces coordonnées appartient à une des chaines et faux sinon.

— Créer une méthode contient dans la classe `Alien` prenant en argument deux entiers x et y et renvoyant true si le point à ces coordonnées touche l'Alien.

— Vous pouvez maintenant modifier votre classe `GestionJeu` pour que lorsqu'un alien est "touché" par une balle, le texte "TOUCHE" s'affiche dans le terminal. **Remarque** : vous allez devoir imbriquer deux boucles : Pour chaque Balle b faire : Pour chaque Alien a faire : `tester a.contient(b.positionX,b.positionY)` .

3.2 Suppression des objets

— Maintenant, au lieu d'afficher "TOUCHE", vous allez remplir deux ArrayList : l'un contenant les balles qui ont touché des aliens, l'autre contenant les aliens qui ont été touchés par une balle. Affichez ces deux ArayList à chaque tour.

— Plutot que d'afficher ces deux ArrayList, on va appeler la méthode remove de l'ArrayList de balles pour supprimer chaque balle ayant touché un alien. De même avec l'ArrayList d'aliens pour supprimer tous les aliens qui ont été touchés.

Remarque La bibliothèque graphique utilise javafx. Il faut le préciser au moment de la compilation et de l'exécution

— pour compiler, il faut ajouter les options suivantes :
`javac -module-path /usr/share/openjfx/lib/ -add-modules javafx.controls *.java`

— `java -module-path /usr/share/openjfx/lib/ -add-modules javafx.controls Executable`