SpaceDefendList

Dans plusieurs méthodes, ***w*** désigne la largeur, ***h*** la hauteur et ***g*** le Graphics du DrawPanel.
Les méthodes qui ne sont pas expliquées sont en générale des accesseurs ou des manipulateurs.

SpaceObject

x,y	position (centre) de l'objet
dx,dy	pas horizontal et vertical de l'objet
size	diamètre
SpaceObject(w)	un nouvel objet est créé tout en haut avec ***size***=40 et position ***x*** aléatoire. -2<***dx***<2, 0<***dy***<3
SpaceObjetc(so)	un nouvel objet est créé avec les mêmes attributs que ***so***, sauf pour ***dx*** qui est inversé.
isTouched(shotX)	dire si un tir à la postion ***shotX*** a touché le SpaceObject
isVisible(w)	dire si l'objet est visible à l'écran (donc pas encore sorti d'un côté)
hasCrashed(h)	dire si l'objet a touché le "sol" en bas
doStep()	l'objet avance selon ***dx*** et ***dy***.
draw(g)	l'objet est dessiné

Shuttle

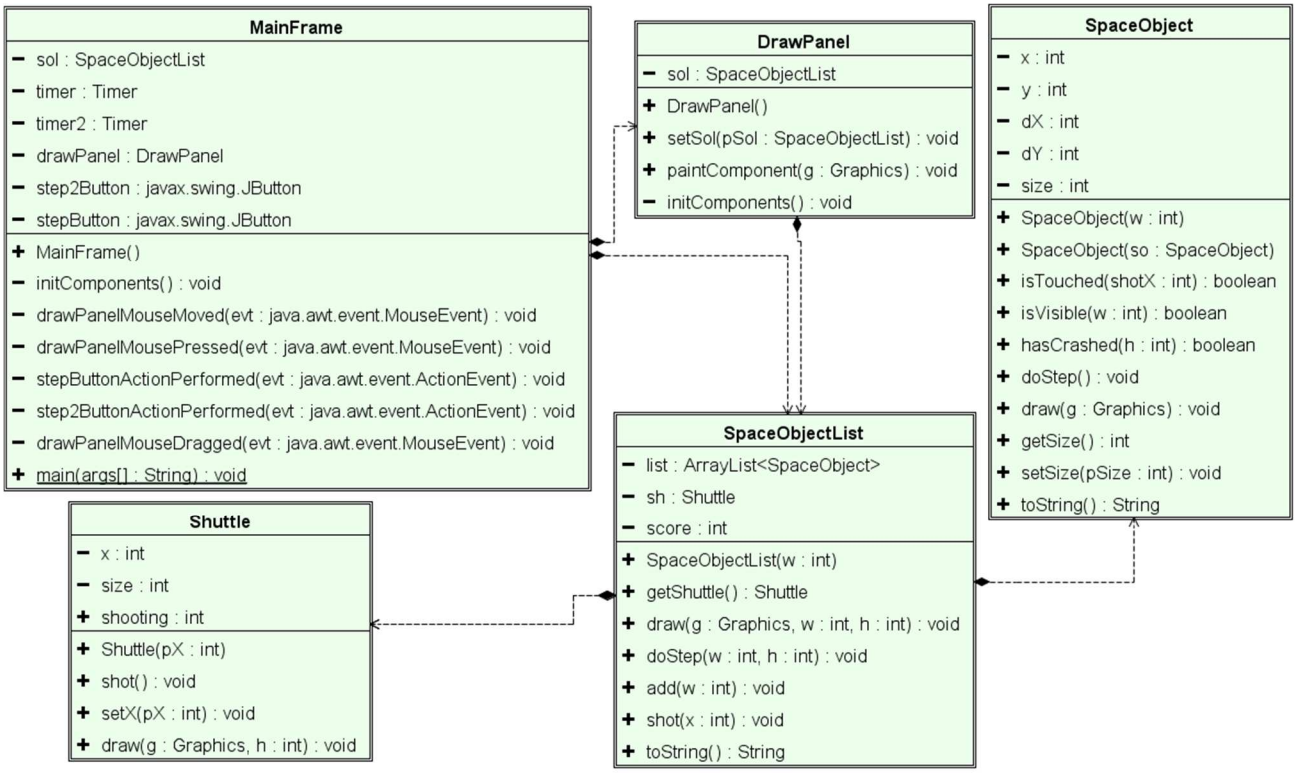
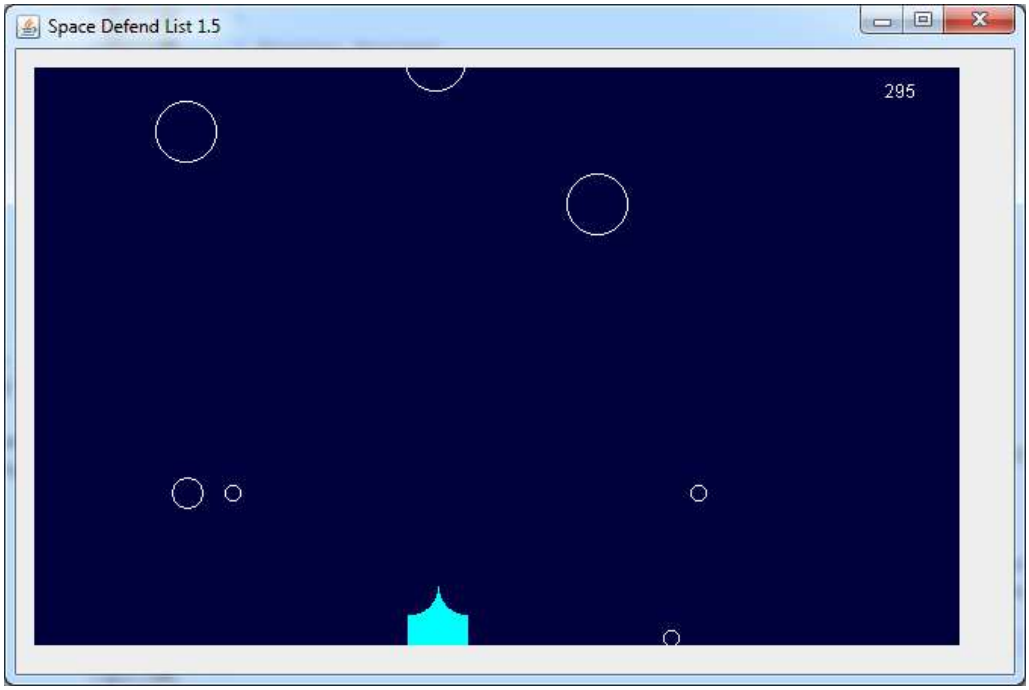
x	position horizontale de l'objet (initialisé avec -1)
size	largeur (=40)
shooting	>0 si le ***shuttle*** est en train de tirer (ligne rouge à dessiner), 0 au début
Shuttle(pX)	constructeur avec valeur pour ***x*** (moitié de la largeur de ***drawPanel***)
shot()	donner à ***shooting*** la valeur 3
draw(g,h)	Le ***shuttle*** est dessiné tout en bas. Il est composé d'un rectangle qui est superposé de deux cercles noirs pour cacher les coins supérieurs. Si ***shooting*** >0 alors le tir (une ligne rouge) est dessinée et la valeur de ***shooting*** est décrémentée.

SpaceObjectList

SpaceObjectList(w)	le ***shuttle*** est créé à la position w/2, ***score*** est initialisé avec 0. Au début la liste est vide.
draw(g,w,h)	On dessine le ***shuttle***, les objets de la liste et aussi le score en haut à droite.
doStep(w,h)	Lance la méthode ***doStep()*** de chaque objet de la liste. Si l'objet n'est plus visible il est enlevé de la liste. Si l'objet s'écrase en bas il est aussi enlevé de la liste et une valeur négative est ajoutée au score.
add(w)	Créer un nouvel **SpaceObjet** et l'ajouter à la liste.
shot(x)	Le ***shuttle*** tire à la position ***x***. Exécuter la méthode ***shot()*** du ***shuttle***. Le score est décrémenté (cela coûte donc un point de tirer). Vérifier pour chaque **SpaceObjet** s'il est touché par le tir. Si c'est le cas, le score est augmenté de 5. Si la taille du **SpaceObject** est 10 alors il est enlevé. Sinon, sa taille est divisée par deux. Puis une copie de ce **SpaceObject** est ajoutée à la liste, avec les mêmes attributs, sauf pour ***dx***.

DrawPanel

sol	La même SpaceObjectList que celle du MainFrame
paintComponent(g)	efface le DrawPanel et dessine la SpaceObjectList si elle n'est pas null

MainFrame

sol	objet de la classe **SpaceObjectList**
timer	Chrono démarré au début avec un delay de 20 ms, sert aux déplacements des **SpaceObjects**. Lié au bouton **stepButton**.
timer2	2e chrono démarré au début avec un delay de 1000 ms, sert à ajouter à la liste un **SpaceObject** au hasard. Lié au bouton **stepButton2**.
drawPanelMouseMove	le ***shuttle*** prend la position x de la souris
drawPanelMouseDragged	Idem
drawPanelMousePressed	le ***shuttle*** tire à la position x de la souris