**PROJET : Tower Defense**

*Conventions sur le code :*

* Ecriture en Français
* Nom de classes en majuscule
* Underscore si nom composé
* Variable lisible
* Docstring
* If a != b
* (a, b)

*Les classes :*

Class Carte

- Attributs :

longueur, largeur, grille = « 0 : Chemin – 1 : Décor – 2 : Construction de tour», carte\_couts

- Méthodes :

getitem(renvoie grille[i,j])

setitem(set grille[i,j])

Class Projectile

- Attributs :

position(tableau 2 éléments), type, degat, vitesse (les pixels pacourus par le projectile par unité de temps)

Class Personnage \*

- Attributs :

vie, vitesse, position, dégâts,

- Méthodes :

Ne pas oublier la méthode pour ajouter le score à l’utilisateur

Class Armee

- Attributs :

nombre\_soldats, case\_objectif

Class Tour \*

- Attributs :

projectile, vie, portee, cout\_construction, cout\_entretien, cout\_amelioration, id\_ tour (0 : tour basique … 10 : Tour très développée), amelioration (0 : Normale, 10 : Très améliorée)

- Méthodes :

Attaque le soldat le plus proche de l’une des bases prioritairement et qui est à portée de la tour

Class Base

- Attributs :

vie, cout\_entretien, cout\_amelioration

Class Joueur

- Attributs :

score, argent, bases (liste des points de vie des bases que l’utilisateur défend)

Class Jeu

Attributs :

carte : du type Carte

joueur : du type Joueur

armee : du type Armée

tours : liste de « Tour »

bases : liste de « Base »

Class Affichage\_fenetre

- Attributs :

liste\_images (plusieurs listes d’images des classes)

- Methode :

affiche(objet)

Class Main :

ouvrir application...

**Remarques :**

**Load file XML for inputs (paramètres du jeu) !**

**On introduit un modèle aléatoire avec randn() pour voir quelle base attaquer, si deux bases possibles.**

\* : Sous classes à créer.