

Tower Defense

Alexis Laouar, Rémi Oudin, Kévin Le Run

March 4, 2016

Plan

- 1 Fonctionnalités
 - Tours
 - Ennemis
- 2 Architecture
 - MVC
 - Design Patterns
- 3 Le SpawnScheduler
 - La création de vagues
 - La gestion d'une vague

Tours

- 4 types de tours
- Tours qui tirent en zone
- Achat et vente de tours

Tours

```
trait TowerType
{
    val tower_graphic =
        ImageIO.read(
            new File(getClass().
                getResource("/towers/base_tower.png").getPath()))
    val size           = 1
    val damage         = 5
    val range          = 5
    val aoe_radius     = 0
    val throw_speed    = 10.0
    val throw_cooldown = 1.0
    val buy_cost       = 50
    val sell_cost      = 25
}
```

Ennemis

- 5 types d'ennemis
- Dont 1 boss et 1 easter egg

Ennemis

```
trait BunnyType
{
    val bunny_graphic =
        ImageIO.read(
            new File(getClass().
                getResource("/mobs/bunny_alt1.png").getPath())
    val initial_hp      = 10.0
    val shield          = 1.0
    val speed           = 2.0
    val reward          = 10
    val damage          = 1
}
```

Modèle Vue Contrôleur

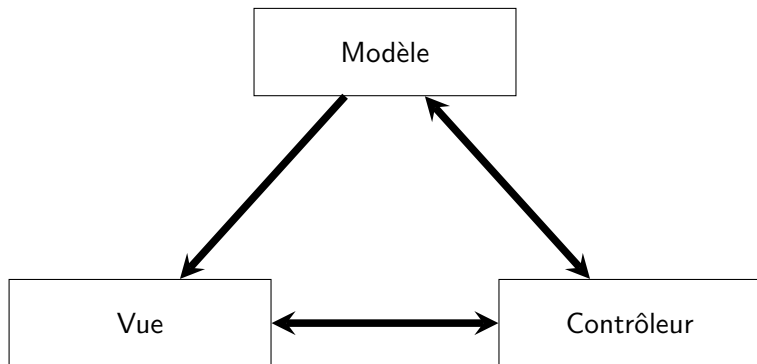


Figure: Modèle-Vue-Contrôleur

Type Object

Au lieu d'avoir une classe par type d'ennemi : une classe ennemi et une classe type d'ennemi

Création des vagues

- Les vagues sont stockées dans un fichier csv
- Chaque vague à son fichier, qu'on lit pour stocker dans un buffer
- Un programme Caml génère des vagues automatiquement selon des facteurs de difficulté.
- Un easter egg : Avec une probabilité 1/1000, c'est un lapin doré super rapide mais qui rapporte beaucoup d'argent qui apparaît

Gestion d'une vague

- Le SpawnScheduler calcule le temps déroulé depuis le début de la vague
- Si le temps d'apparition d'un ennemi est inférieur au temps déroulé depuis le début de la vague, on le clone depuis le SpawnScheduler pour le rajouter dans les ennemis vivants.