



## Ecos - Simulation d'écosystème

Bastien BONVARLET    Brandon CHAMPENOIS    Joris  
MASSON

Université de Caen Normandie

Présentation générale

Répartition des tâches

Explication du projet

Démonstration

Conclusion

# Ecos

## Généralités

- ▶ Une carte sur laquelle évoluent les entités la peuplant(possibilité d'en créer avec Tiled)
- ▶ Divers type d'entités:
  - ▶ Humains
  - ▶ Orcs
  - ▶ Loups
  - ▶ Ours
  - ▶ Lapins
- ▶ Une chaîne alimentaire
- ▶ Un système de reproduction
- ▶ Un système de combat

# Pourquoi ce choix de projet?

## Raisons

- ▶ Ça avait l'air sympa
- ▶ Sujet assez libre
- ▶ Le sujet le plus inspirant pour nous

# Qui a fait quoi?

## Bastien

- ▶ Les différentes cartes
- ▶ Toute la base du projet
  - ▶ Les classes
  - ▶ Interface graphique
- ▶ Tentative de gestion des animations
- ▶ Notre incroyable et magnifique logo

## Brandon

- ▶ Les différents sprites
- ▶ Le menu de lancement
- ▶ Le rapport LaTeX

# Qui a fait quoi?

## Joris

- ▶ La programmation de certains aspects du projet:
  - ▶ L'algorithme A\*
  - ▶ Système de combat
  - ▶ Système de reproduction
- ▶ Création des graphiques
- ▶ Ce magnifique diaporama en beamer

# Général

- ▶ Les différentes entités vivent leur vie
  - ▶ Déplacements aléatoires
  - ▶ Elles s'attaquent entre-elles
  - ▶ Elles peuvent se reproduire
  - ▶ Elles peuvent mourir
    - ▶ Si leur vie atteint 0
    - ▶ Si elles ont atteint leur âge limite
    - ▶ Si elles sortent de la matrice(bug de naissance)
- ▶ Le temps passe
  - ▶ Il passe à un rythme de 60 jours par seconde, un jour par frame
  - ▶ 365 jours dans une année
  - ▶ Chaque entité a un intervalle de temps de vie

# Déroulement du programme

## 1. Initialisation

- 1.1 La carte est créée et affichée
- 1.2 Les cases contenant des collisions sont récupérées et stockées pour plus tard (quand on crée la carte, on peut définir des filtres, il y en a un pour les collisions)
- 1.3 On en déduit les cases n'ayant pas de collisions
- 1.4 On crée un nombre fixe d'entités de manière aléatoire
  - ▶ Type
  - ▶ Genre
  - ▶ Position

## 2. Les entités font leurs vies, et le monde suit son cours



## Le système de classes

- ▶ Une classe centrale: Game
- ▶ Une classe mère représentant toutes les entités vivantes: LivingEntity
- ▶ Une classe mère pour les objets

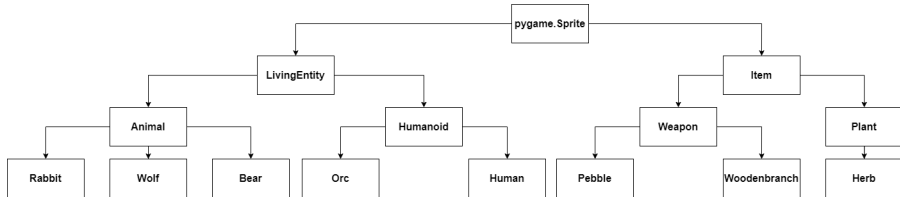


Figure: Diagramme de classes

On lance la simulation, on y reviendra plus tard

# Les déplacements

- ▶ Gérés par l'algorithme A\*
- ▶ Destination choisie au hasard
- ▶ Une fois la destination atteinte, une autre est choisie au hasard
- ▶ Une seule exécution d'A\* par frame par entité

# La gestion des collisions

- ▶ Chaque entité est enregistrée dans un groupe de sprite
  - ▶ Chaque groupe de sprite est enregistré dans un dictionnaire regroupant tous les groupes d'entités
  - ▶ Le dictionnaire possède comme clé les noms des types d'entités
- ▶ On ne prend que les groupes d'entités nécessaires lors des vérification de collisions

## Le système de combat

- ▶ Les entités ne peuvent pas se battre avant que le monde ait atteint l'âge de 3 ans
- ▶ Une attaque survient lorsque deux entités rentrent en contact, et peuvent s'attaquer
- ▶ Un délai d'attaque est présent(150 frames)
- ▶ Une attaque a une probabilité de  $1/3$  d'être initiée par une entité

## Le système de reproduction

- ▶ Le monde doit avoir plus d'un an
- ▶ Les entités ne peuvent pas se reproduire avant d'avoir atteint un âge minimal spécifique à chaque type d'entité
  - ▶ Humains/Orcs: 16
  - ▶ Loups: 3
  - ▶ Lapins: 2
  - ▶ Ours: 10
- ▶ Ce sont les femelles qui initient la reproduction lorsqu'elle rentre en collision avec une entité du même type, et ayant un genre différent

## Le système d'arme

- ▶ Deux types d'armes
- ▶ Sont à des endroits fixes
- ▶ Seuls les humanoïdes peuvent s'en servir
- ▶ Il y a un "temps de recharge" de un an

C'est le moment de revenir sur la simulation



## Ce qu'on aurait voulu faire

- ▶ Plus d'entités
- ▶ Des animations
- ▶ Réparer le lapin
- ▶ Améliorer le système d'arme

# Conclusion