Tower défense avec une IA deep :

Développement IA

Développement Tower défense

Le principe serait de créer d'abord un jeu vidéo de type "Tower Defense" ``où l’objectif est de défendre une zone contre des vagues successives d’ennemis se déplaçant suivant un itinéraire ou non, en construisant et en améliorant progressivement des tours défensives`` ([Wikipédia](https://fr.wikipedia.org/wiki/Tower_defense)).

Puis, dans un second temps, nous programmerons une Intelligence Artificielle basée sur le "Deep Learning", l'IA apprendra d'elle-même à partir de données plutôt qu'à partir de règles (comme il est plus courant). Le but de cette IA étant de remplacer le joueur pour se battre contre les vagues d'ennemies.

Gestion projet/git … : Cyril

Intergrateur : Victor

Développement IA : Rayan, Cyril

Développement TD : Alexandre, Guillaume, Calvin

Langage :

IA => python (stockage à voir avec techno) : langage de bas niveau, facile à implémenter et utiliser dans d’autres solution et d’autres langage. Communauté très grande donc framework facile à trouver avec doc. Compétences en python partiel gain de temps de formation

Pas de bdd pour le jeu static

UML ??

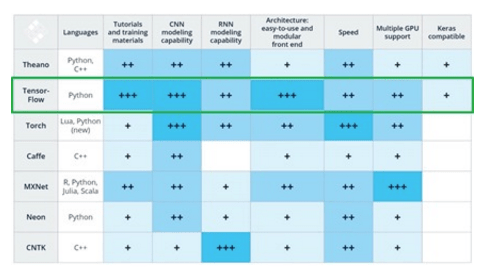
TD : unity : la base pour le jeu vidéo gain de temps, compétences en C# déjà partiel (temps de formation plus court)

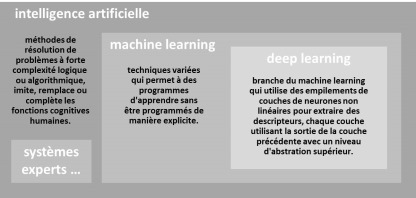
(((((Mobile en dépend de l’avancement ???? ))))

Plan de test : test unitaire ?? pyTest

Boucle d’intégration avec contrôle ??

PyTorch

TensorFlow  
[](https://www.frenchweb.fr/peut-on-benchmarker-lintelligence-artificielle/292336)

[](https://www.oezratty.net/wordpress/2017/peut-on-benchmarker-ia/)

Règles des TD :

1 monstre qui arrive à la ligne d’arrivé -1 (variable) vie et mort du monstre  
Nombre de vagues infini

Vague avec des pauses

Meilleure façon d’évaluer les IA : systèmes de scores = nombre de monstres tué

Taille prédéfinie : (~15x15 cases)

* Vagues : Monstres caractéristiques (nb, vitesse, PV) / vulnérabilité aux types de tourelles
* Nombres de vies
* Tourelles (Vitesse t’attaque, élément, dégâts, portée)
* Emplacement
* Ressources
* Taille de map
* Chemin prédéfini -> choix de chemin -> chemin libre
* Passe en mode continue

Variables à vérifier :

1. Les tourelles
   1. Chemin prédéfini, choisir la meilleure tourelle à un emplacement
2. L’emplacement des tourelles
   1. Plusieurs emplacements, 1 seule tourelle à placer
   2. Zone d’emplacements, 1 tourelles
   3. Emplacement libre
3. Gestion ressources
   1. Chemin prédéfini, plusieurs emplacements plusieurs tourelles
   2. Chemin prédéfini, plusieurs emplacements plusieurs tourelles, achats entre les vagues, gestion budget (qualité/prix, quantité)
4. Gestion chemin
   1. Plusieurs chemins possibles, avec tourelle similaire
   2. Plusieurs chemins, tourelles différentes

Améliorations futures possibles

Ajout de monstres et de tourelles, limiter ressources entre chemin et l’or les tourelles

Gestion temps réel et gestion avec des pauses entre vagues

Mode vague en continue (pas de pause)

Règle du score : Plus tué plus de score