

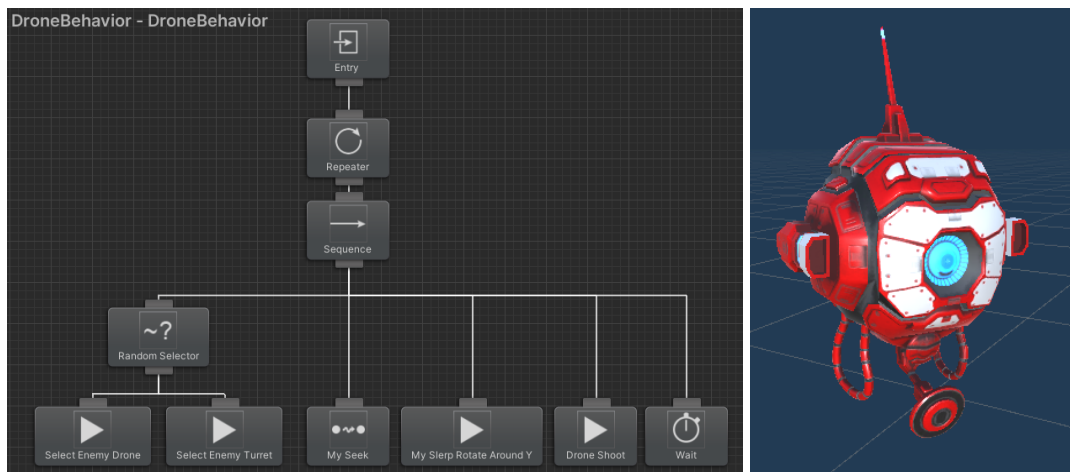
EPF 2022 - 2023

Syllabus du module “Real Time Decision Making in Video Game”

par David Bilemdjian

Objectifs

- Découverte de certaines techniques d'IA utilisées dans le jeu vidéo pour la prise de décision des PNJ, en particulier les arbres de comportement (Behavior Trees)
- Elaboration d'arbres de comportement pour des unités combattantes dans un mini-jeu de genre “RTS Wargame”



Modalités

- Présentation magistrale à base de slides
- Projet TP de développement en groupes de 4 étudiants sous Unity3D/C#

Pré-requis

- Une première expérience de développement sous Unity3D/C# est préconisée pour la réalisation du projet

Logiciels à Installer

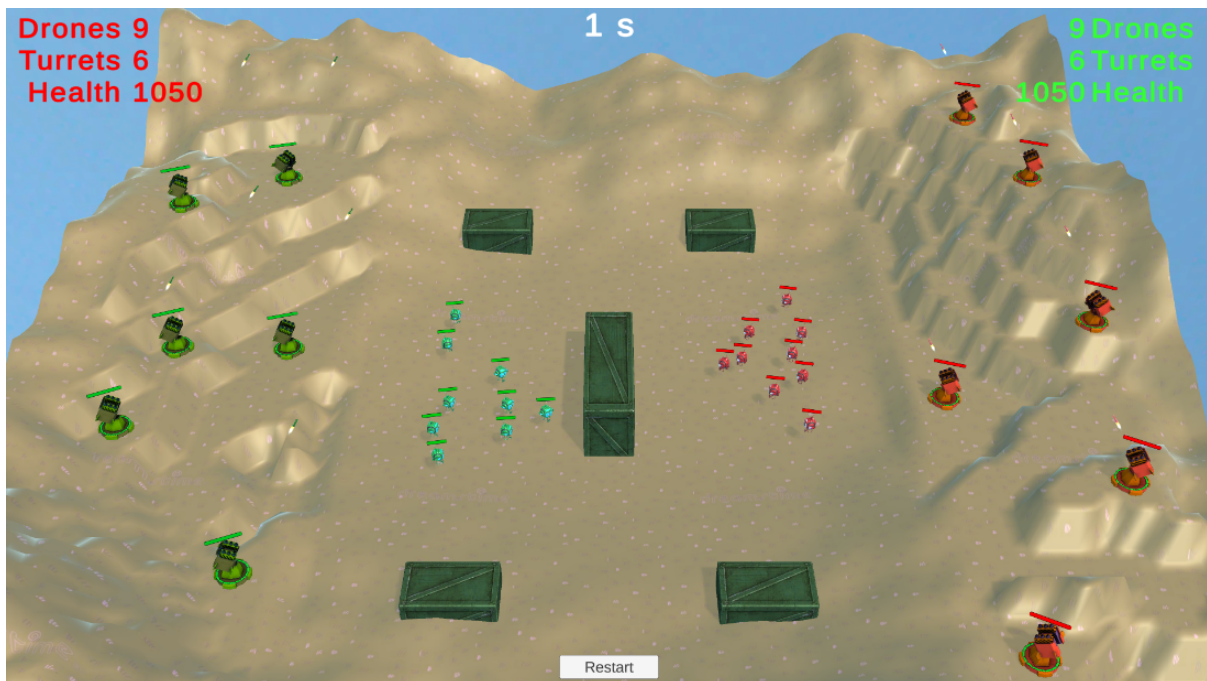
- Unity3D version LTS 2021.3.15f1 (...ou supérieure). Tous les membres d'une même équipe devront utiliser la même version
- Un IDE de développement C# compatible avec Unity3D: Visual Studio, Visual Studio Code, JetBrains Rider
- Un client Git: SourceTree, GitKraken, GitHub Desktop

Calendrier

- Le 02/01/2023 (4h)
 - PM:
 - Présentation du module, des objectifs et des modalités (0,5h)
 - Introduction à l'IA de prise de décision dans les jeux vidéos (2,5h)
 - Présentation du Projet "bac à sable" de base fourni (1h)
- Le 03/01/2023 (4h)
 - PM:
 - En équipe: préparation d'un repository Git pour le projet (0,5h)
 - Présentation des fonctionnalités Unity3D utiles au projet (0,5h)
 - Présentation de Behavior Designer (1h)
 - Exploration de Behavior Designer et du projet de base en autonomie puis, brainstorming en équipe sur les stratégies / tactiques à mettre en place et les arbres de comportement correspondants (2h)
- Le 16/01/2023
 - AM & PM:
 - Accompagnement de production (8h)
- Le 19/01/2023
 - AM:
 - Accompagnement de production (2h)
 - Ou bien Tournoi inter-équipes (si techniquement OK) (2h)

Le projet

Un projet de jeu embryonnaire de genre "RTS War Game" développé sous Unity3D est fourni.



Deux armées (ROUGE et VERTE) s'affrontent sur un terrain de 60m x 60m constitué d'une plaine, de terrasses surplombantes et d'obstacles divers.

Chaque armée occupe un côté de la map. Deux types d'unités sont proposées:

- des drones mobiles capables de se rapprocher d'une cible, de viser puis tirer des missiles à la trajectoire "rectiligne"
- des tourelles d'artillerie (mortiers) statiques et orientables, capables de viser et de tirer des grenades à la trajectoire parabolique (comportement balistique)



Un missile tiré par un drone n'inflige des dégâts qu'à l'unité adverse avec laquelle il collisionne.

Une grenade tirée par un mortier, lorsqu'elle retombe sur le sol, inflige des dégâts de zone aux unités adverses se trouvant dans son rayon d'action.

Chaque type d'unité possède son IA propre, définie par un arbre de comportement.

Les arbres de comportement présents dans le projet sont pour le moment très basiques:

- Boucle d'IA du drone:
 - demande d'une cible (drone ou tourelle au hasard) à son chef d'état-major (*ArmyManager*)
 - rapprochement de la cible
 - orientation vers la cible
 - tir de deux missiles
- Boucle d'IA de la tourelle d'artillerie:
 - demande d'une cible (priorité aux tourelles ennemies) à son chef d'état-major (*ArmyManager*)
 - changement d'orientation pour viser
 - tir d'une grenade

Les unités possèdent des caractéristiques listées en *Annexe A*, qui ne doivent pas être modifiées.

Par défaut, l'armée verte est initialement plus forte que l'armée rouge.

L'objectif est de développer en équipe une IA de l'armée **ROUGE** plus performante que celle de l'armée **VERTE**

Le développement consiste essentiellement en:

- le développement d'un arbre de comportement pour le drone (*DroneRedBehavior*), et éventuellement de nouvelles tâches (actions, conditions, décorateurs, etc)
- le développement d'un arbre de comportement pour la tourelle d'artillerie (*TurretRedBehavior*), et éventuellement de nouvelles tâches (actions, conditions, décorateurs, etc)
- le développement d'un chef d'état-major (*ArmyManagerRed*)
 - le chef d'état-major est capable de tout voir :
 - position et trajectoire de ses propres unités, ainsi que des unités ennemies (drones, tourelles, missiles et grenades)
 - état de santé de toutes les unités
 - les obstacles
 - les particularités du terrain
 - par contre il n'est pas capable de connaître les intentions de l'ennemi

Le comportement de l'armée VERTE ne doit pas être modifié.

Evaluation

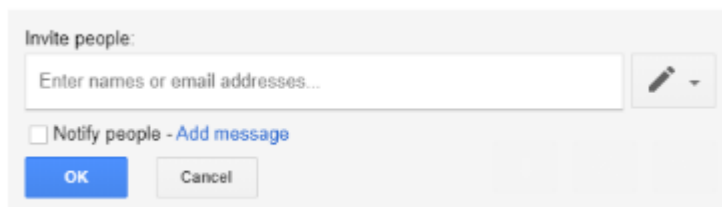
L'évaluation porte uniquement sur le projet et prend en compte:

- la réflexion stratégique et tactique
- la complexité des arbres de comportement
- la qualité de développement des scripts C#
- l'efficacité de l'armée. Voir l'Annexe B pour les critères statistiques

Modalités de livraison

La date limite de livraison est fixée au dimanche 22 janvier 2023 à 23h59.

Vous devez livrer une archive compressée, en me les partageant depuis votre compte Google Drive (ou autre) sur l'adresse dbilemdjian@gmail.com. Par contre, vous décochez « Envoyer une notification » lorsque vous partagez votre fichier, et vous m'envoyez le lien de partage « Link to share » de l'archive (je dis bien l'archive, pas le répertoire) par retour du mail d'envoi ce document.



Votre mail de livraison ne doit donc contenir, au-delà des formules de politesse d'usage, que le lien Drive pointant vers l'archive.

Description du contenu de l'archive:

- Un fichier ReleaseNotes exposant:
 - les noms des membres de l'équipe
 - votre réflexion stratégique et tactique
 - ce qui a été développé:
 - arbres de comportement
 - scripts C#
 - vos remarques particulières: limitations, bugs connus, voies d'amélioration
 - la répartition de la production: qui a fait quoi ?
- Un fichier CSV (utilisez la tabulation "\t" comme séparateur des données) listant les résultats statistiques, une ligne par bataille. Un fichier template vous a été fourni. Voir *Annexe B*.
- une vidéo (HD ou Full HD) compressée d'un combat épique
- le projet *Unity3D*, dont vous aurez pris soin de retirer le répertoire Library



Le nom de cette archive doit adopter la nomenclature suivante:

EPF_2022_2023_DecisionAI_UnityProject_<Nom1Nom2Nom3Nom4>.zip

Références

Documentation de Behavior Designer (Opsive)

[Overview - Opsive](#)

Série de vidéos tutorielles sur Behavior Designer

<https://www.youtube.com/watch?v=PuLuwzgYB4g&index=1&list=PLCPL5IYZm1NLVVxJx3-CFTLuHvLbCgmzk>

Les Navmesh dans Unity3D

<https://docs.unity3d.com/Manual/Navigation.html>

AI for Game Developers (Glenn Seemann, David M Bourg)

<https://www.oreilly.com/library/view/ai-for-game/0596005555/>

AI for Games (Ian Millington)

<https://www.eyrolles.com/Informatique/Livre/ai-for-games-third-edition-9780367670566/>

Annexe A

Armée ROUGE

Nombre de drones en début de bataille: 13

Nombre de tourelles d'artillerie en début de bataille: 6

Armée VERTE

Nombre de drones en début de bataille: 15

Nombre de tourelles d'artillerie en début de bataille: 7

Tableau des caractéristiques par type d'unité.

Caractéristiques du drone

Action "My Seek"	
Speed	10 m/s
Angular Speed	180 °/s
Arrive Distance	6 m
Acceleration (component NavMeshAgent)	8 m/s ²
Action "My Slerp Rotate Around Y"	
Slerp Coef	4
Arrival Angle	5°
Action "Wait' (en fin de chaque boucle d'action)	
Random Wait	true
Random Wait Min	1
Random Wait Max	2
Component "Health"	
Start Health	50

Caractéristiques de la tourelle d'artillerie

Action "Select Enemy" (drone & turret)	
Min Radius	5 m
Max Radius	50 m
Action "Turret Seek Target"	
Rotation Speed (component Turret)	90 °/s

Rocket Travel Duration	4 s
Action “Wait” (en fin de chaque boucle d’action)	
Random Wait	true
Random Wait Min	1
Random Wait Max	3
Component “Health”	
Start Health	100

Caractéristiques du missile tiré par le drone

Component “Missile”	
Life duration	3 s
Acceleration	40 m/s ²
Translation Max Speed	20 m/s
Damage Points	5

Caractéristiques de la grenade tirée par la tourelle d’artillerie

Component “Rocket”	
Max Life duration	10 s
Damage Radius	5 m
Damage Points	10

Les caractéristiques géométriques des unités n’ont pas été listées, mais elles ne doivent pas être modifiées, en particulier les dimensions des différents *Colliders*.

Annexe B

Les critères statistiques sont, par ordre d'importance, et **pour chaque bataille**:

- victoire ou défaite (devrait être victoire pour chaque bataille contre l'IA par défaut), (1 pour victoire, 0 pour défaite)
 - Il se peut qu'une situation de "deadlock" (impasse) survienne, par exemple lorsqu'il reste uniquement des tourelles d'artillerie dont le rayon d'action n'est pas suffisant pour se détruire mutuellement: dans ce cas la simulation n'est pas comptabilisée et doit être relancée.
- nombre de drones en vie à la fin de la bataille,
- nombre de tourelles en vie à la fin de la bataille,
- total cumulé des points de vie restants
- durée de la bataille (en s),

20 batailles doivent être simulées contre l'IA par défaut (dummy AI)

Exemple de tableau statistique des résultats après simulation de 20 batailles

INDEX_BATTLE	IS_VICTORY	DURATION	ALIVE_DRONES	ALIVE_TURRETS	REMAINING_CUMULATED_HEALTH_POINTS
0	1	65	10	1	200
1	0	45	0	0	0
2	1	80	5	2	225
3	1	77	4	1	125
4	1	85	12	0	550
5	0	67	0	0	0
6	1	90	10	0	231
7	1	72	8	1	185
8	1	83	0	2	50
9	1	96	5	2	200
10	1	68	4	2	107
11	1	123	9	1	352
12	1	56	3	3	104
13	1	78	1	0	10
14	1	98	5	1	50
15	1	86	8	2	28
16	1	77	7	1	125
17	1	69	6	0	39
18	1	70	3	2	78
19	1	76	7	2	257
20	1	56	8	1	365

En fin de bataille une chaîne de caractères des statistiques de l'armée ROUGE est automatiquement écrite dans le "clipboard" de l'ordinateur. Il vous suffit alors de la coller (Ctrl+V) dans un fichier ou dans une spreadsheet.