

Projet NFP121 : Game of Drones

Objectif

Le but de cet exercice est de vous permettre d'utiliser différents design patterns en **Kotlin**. Vous allez devoir les combiner dans un projet de type « compétition » dans la plateforme Codingame.

Lien vers le projet :

<https://www.codingame.com/ide/puzzle/game-of-drones>

Règles du jeu

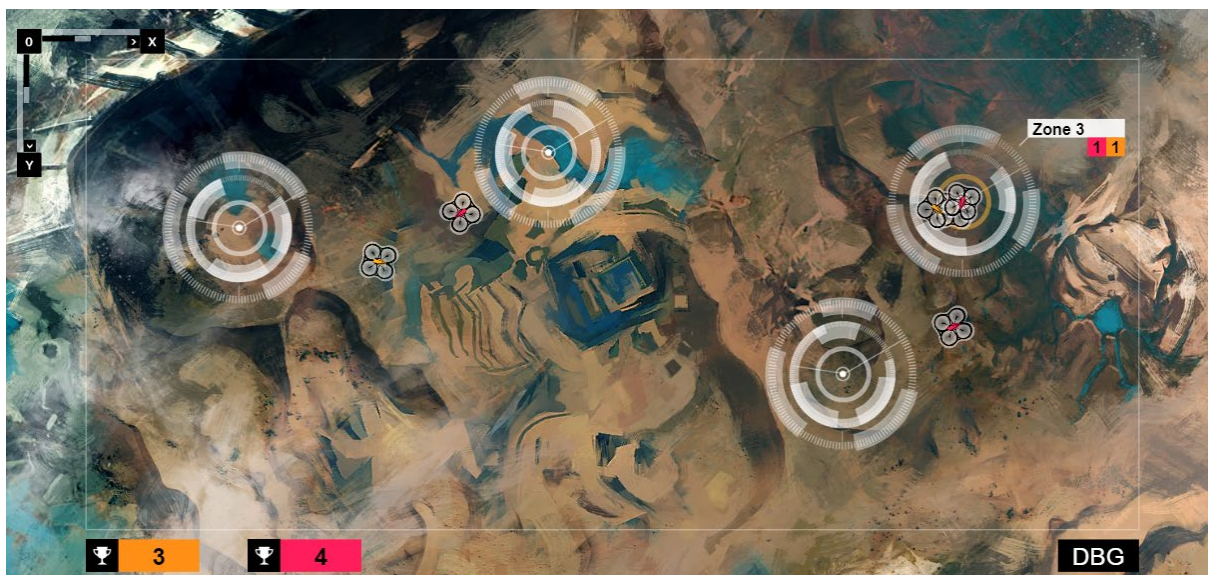
Ce puzzle est multijoueur, ce qui implique que vous êtes en compétition contre d'autres joueurs (humains ou ordinateurs).

Le but de ce puzzle est de dominer des zones en utilisant des drones. Une zone est dominée quand le nombre de vos drones est supérieur au nombre de drones du joueur adverse.

Chaque zone dominée vous rapporte des points à chaque tour. Au bout de 200 tours de jeu, le joueur avec le plus de points remporte la partie.

A chaque tour de jeu vous devez indiquer à vos drones la position à atteindre.

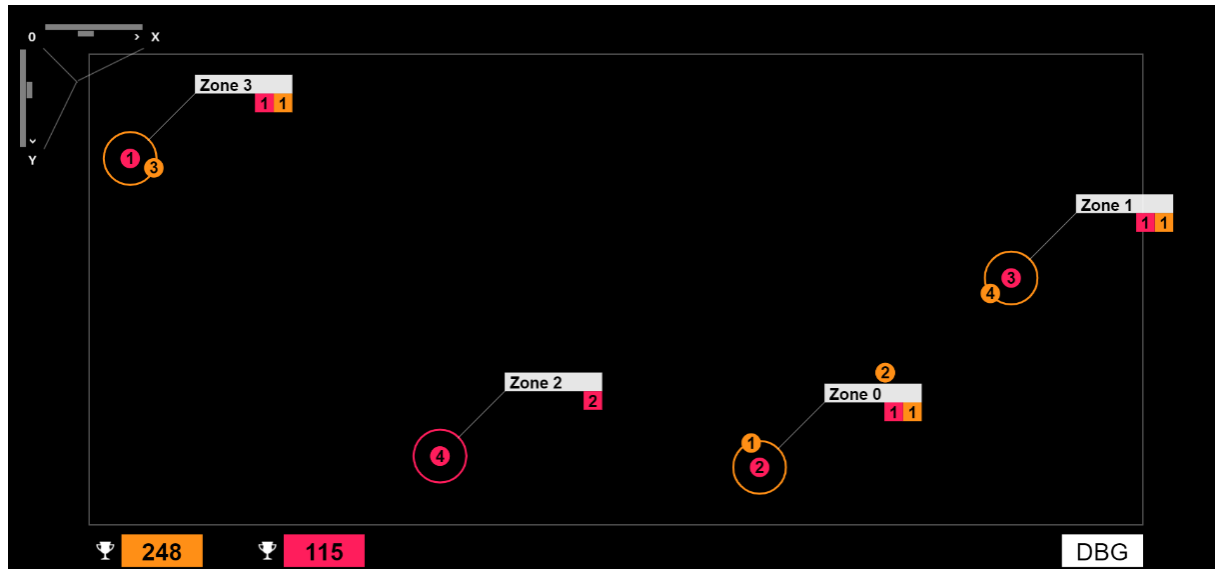
L'interface graphique du jeu est représentée par l'image ci-dessous. Vos drones sont représentés en orange, les drones de l'adversaire en rouge. Les zones à capturer sont représentées par les cercles. Les zones se colorent en fonction du joueur qui la contrôle, un rappel du nombre de drones de chaque joueur dans la zone est indiqué par des chiffres.



Le terrain est une grille rectangulaire. Il mesure 4000 unités en largeur (X) et 1800 en hauteur (Y). Le point 0,0 est en haut à gauche, le point 4000,1800 est en bas à droite. Un drone parcourt au maximum 100 unités de distance au maximum

Ainsi, si vous demandez à un drone qui est en 0,0 d'atteindre 4000,0 il va avoir besoin de 40 tours de jeu. Tenez-en compte lors de vos déplacements.

Vous avez la possibilité d'afficher le mode debug de la visualisation afin d'avoir des informations épurées



Règles du projet

Ce projet doit être réalisé en **Kotlin** en utilisant obligatoirement **au minimum** les classes suivantes :

- **Point** qui représente un point de la grille avec ses coordonnées X,Y
- **Drone** qui représente un drone avec sa position actuelle, son id, ...
- **Zone** qui représente une zone capturable avec le joueur qui l'a actuellement capturée, sa position,...
- **GameEngine** qui gère les tours de jeu dans leur globalité
- **Player** qui représente un joueur avec son ID et avec des références vers ses drones

Ainsi qu'**au minimum** les design patterns suivants :

- **Factory** afin de pouvoir créer les drones
- **Singleton** pour que la classe GameEngine ne soit instanciée qu'une seule fois
- **State** afin de définir l'état des drones (en déplacement, en cours de capture, en protection de zone, ...)
- **Listener** : les drones seront listeners des événements de capture des zones. Lorsqu'une zone sera capturée par un joueur, l'évènement de capture sera transmis aux drones afin qu'ils en soient informés.

Modalités logistiques

Ce projet sera réalisé individuellement. Vous pouvez vous aider de toutes les informations que vous le souhaitez. N'hésitez pas à vous appuyer sur les cours, les exemples de code envoyés ainsi que toute documentation que vous trouverez sur le net.

Certaines séances des mois d'avril et de mai seront entièrement dédiées à la réalisation de ce projet. Bien entendu, un travail en dehors des séances prévues à cet effet devra être fourni par vos soins, au risque de ne pas atteindre l'objectif et recevoir une note très faible.

Le rendu attendu est le code source complet de votre programme (il peut être décomposé en plusieurs fichiers) ainsi qu'un document Word de quelques pages indiquant le fonctionnement de votre algorithme de décision, ainsi que du schéma UML de votre programme. Il doit être envoyé aux emails thomas.martinet@utbm.fr ainsi que alexandre.lombard@utbm.fr avec comme sujet « NFP121 projet 2021 – VOTRE NOM ».

La date de rendu est fixée au 12 juin 2021. Tout rendu arrivé après cette date sera considéré comme nul et recevra la note de 0/20.

Une présentation de votre rendu devra être effectuée pendant la séance NFP121 du 13/06/21. Cette présentation devra s'appuyer sur un support de présentation type Powerpoint. Elle devra durer 15 min suivie de 15 min de questions.

Une note de projet sera calculée en fonction :

- Du classement que votre algorithme aura été capable d'atteindre sur la plateforme Codingame.
 - *Bonus : si votre algorithme est capable de gagner contre le joueur « hankerspace », vous obtiendrez automatiquement la note maximale.*
- De la qualité de votre code
- Du respect des consignes
- De votre présentation

Conseils

Le jeu va se décomposer en plusieurs phases :

1. Initialisation : la plateforme va vous envoyer des informations que vous devrez récupérer et stocker. Notamment le nombre de joueurs, l'identifiant de votre joueur, le nombre de drones pour chaque joueur, le nombre de zones sur la carte ainsi que leurs positions.
 - a. Vous devrez au cours de cette phase instancier les classes Player et Zone avec les informations fournies.
2. Tour de jeu : la plateforme va vous envoyer des informations pour le tour de jeu en cours.
 - a. Pour chaque zone, l'identifiant du joueur qui domine la zone (ou -1 si elle n'est pas dominée)
 - b. Pour chaque drone, ses coordonnées actuelles.
 - c. Vous allez donc devoir utiliser ces informations pour mettre à jour la position des drones dans vos instances de la classe Drone, ainsi que prendre des décisions afin de donner une destination souhaitée pour chacun de vos drones.

N'hésitez pas à vous envoyer des messages de debug dans l'output en utilisant `System.err.println`.

Il est important dans un premier temps de vous concentrer sur la structure de votre programme et les interactions entre classes. La phase d'algorithmique et de calcul de positionnement pourra venir dans un second temps.

Vous utiliserez les formules de calcul de distance entre deux points sur le plan cartésien (https://fr.wikipedia.org/wiki/Distance_entre_deux_points_sur_le_plan_cart%C3%A9sien)

Mettez à l'épreuve votre algorithme progressivement. D'abord contre un seul ordinateur, puis plusieurs. Vous pouvez définir vos propres options de jeu pour spécifier le nombre de drones et le nombre de zones à utiliser pour cette partie.

N'hésitez pas à travailler sur l'IDE de votre choix pour le plus gros des développements, copiez/collez votre code dans la plateforme codingame pour effectuer les tests. Certains liens google donnent des indications pour connecter IntelliJ à Codingame.