



Projet Jeu de l'Oie

Objectifs

Vous connaissez grâce aux projets précédents les concepts fondamentaux de l'algorithmique et de la programmation :

- Les variables (leur portée et leur type)
- Les conditions
- Les boucles
- Les fonctions

Ce projet est là pour vous les faire manipuler plus encore ! Face à la longueur des codes produits vous serez également amenés (si vous ne l'avez pas déjà fait) à manipuler un débogueur.

Modalités

- Durée du projet : 5 jours.
- Ce projet sera réalisé en îlots

Description du projet

- Le but de ce projet est de coder un “**Jeu de l'oie**” multi joueur.
- Afin de vous faciliter la tâche, le projet a été découpé en itérations successives.
- Ce jeu de réflexion très avancé sera implémenté en suivant les règles officielles de la FIJO¹ énoncées ci-dessous.

Règles du jeu de l'oie

Matériel

- Chaque joueur joue chacun son tour en lançant les 2 dés. Le joueur avance son pion suivant le nombre obtenu.
- Plateau de jeu

Condition de victoire :

- Pour gagner une partie, il faut être le premier à arriver sur la case 63.
- Obligation d'arriver pile sur la case 63. Si le score au dés est supérieur au nombre de case le séparant de la victoire, il devra reculer d'autant de

1 Fédération Internationale du Jeu de l'Oie

cases supplémentaires. [**Exemple** : Le joueur est en case 60. Il lance les dés et obtient 10. Le joueur devra retourner à la case 56 ($60 + 3 - 7 = 56$)]

Règles spéciales au commencement du jeu :

- Si l'un des joueurs fait **9** par **6 et 3**, il doit avancer son pion immédiatement au nombre **26**.
- S'il fait **9** par **4 et 5**, il ira au nombre **53**.
- Si un joueur fait **6**, il doit se rendre sur la case **12**.

Cases spéciales sur le plateau :

- Des **oies** sont placées en cases **9, 18, 27, 36, 45 et 54**. Le joueur qui y tombe avance de nouveau du nombre de points réalisé.
- Un **hôtel** en case **19**. Un joueur qui y tombe passe ses 2 prochains tours.
- Un **puits** en case **3**. Si un joueur y tombe il devra attendre qu'un autre joueur arrive au même numéro. L'autre joueur tombera dans le puits et devra attendre d'être secouru.
- Un **labyrinthe** en case **42**. Le joueur se perd et devra retourner en case **30**.
- Une **prison** en case **52**. Si un joueur y tombe il devra attendre qu'un autre joueur vienne le libérer.
- Une **tête de mort** en case **58**. Le joueur qui y tombe recommence la partie depuis le début

Interaction entre joueurs

- Celui qui est rejoint par un autre joueur sur la même case devra se rendre sur la case où l'autre joueur se situait avant de jouer.

Informations et consignes générales

Les différentes itérations vont suivre un schéma répétitif. Les modalités et consignes seront les mêmes pour toutes les itérations et ne seront donc pas répétées. Seul le cahier des charges varie d'une itération à l'autre.

Modalités

- Travail en autonomie puis en groupe
- Production individuelle

Consignes

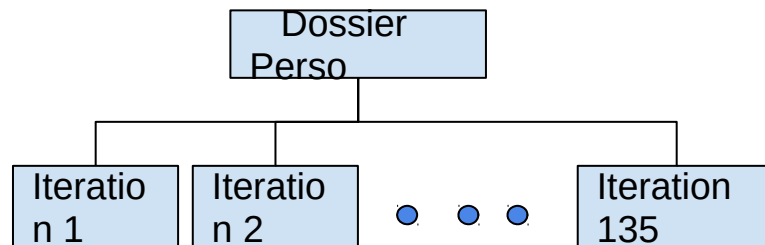
1. Travail individuel : Vous dessinerez les organigrammes de chaque algorithme AVANT de le coder.
2. Vous confronterez vos idées au sein de l'îlot pour aboutir à un organigramme

commun que vous collerez à votre tableau .

3. Vous programmerez cet algorithme sur Processing.

Livrables

- Organigrammes de programmation (logigrammes)
- Code Processing (.pde) sur votre dossier personnel.
- Ces livrables seront stockés dans vos dossiers personnels itération par itération.



Ressources

1. <https://processing.org/reference/>
2. https://fr.wikipedia.org/wiki/Organigramme_de_programmation
3. <https://troumad.developpez.com/C/algorigrammes/>

Itération 1

Objectifs de l'activité

- Manipuler les types "**int**" en Java
- Manipuler les "**conditions**" en Java
- Manipuler les boucles "**while**" en Java
- Formaliser un logigramme simple

Cahier des charges

Vous devez fournir un algorithme simulant le comportement suivant :

- Un joueur lance deux dés différents.
- Il avance d'autant de cases que nécessaire.
- Le joueur doit tomber exactement sur la case 63 pour terminer la partie.
- Tout sera fait dans la console (pas de partie graphique).

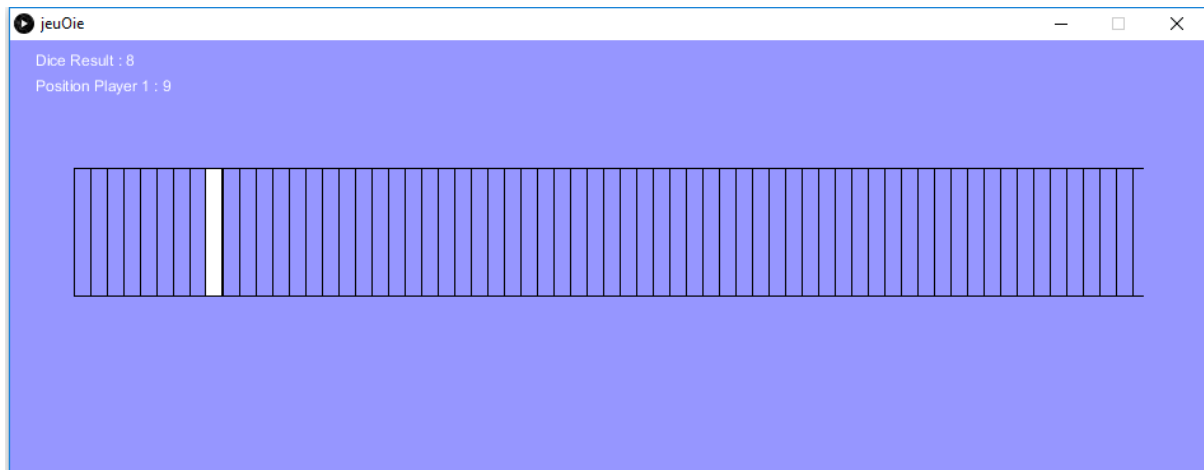
Itération 2

Objectifs de l'activité

- Manipuler les boucles “**for**” en Java
- Se familiariser avec les représentations graphiques en 2D (x et y).

Cahier des charges

- On commence la partie graphique du jeu.
- Créer un plateau composé de **63 cases**. Inutile de se compliquer la tâche : des cases organisées le long d'une ligne feront l'affaire (voir dessin ci dessous). Toutes les cases sont **identiques**.



Ressources :

- https://processing.org/reference/rect_.html
- https://processing.org/reference/rectMode_.html
- https://processing.org/reference/background_.html

Cadeau : citations de sagesse

« N'importe quel idiot peut écrire du code qu'un ordinateur peut comprendre. Les bons programmeurs écrivent du code que les humains peuvent comprendre. »

- Martin Fowler

« Je ne suis pas un excellent programmeur. Je suis juste un bon programmeur avec d'excellentes habitudes. »

- Kent Beck

« Codez toujours comme si le gars qui finira par maintenir votre code était un psychopathe

violent qui sait où vous vivez. »
- John Woods (?)