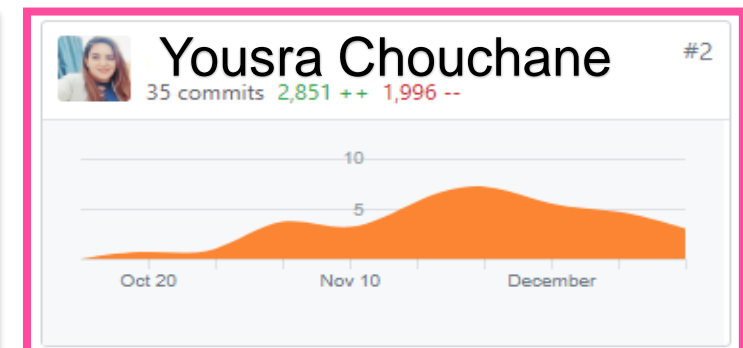
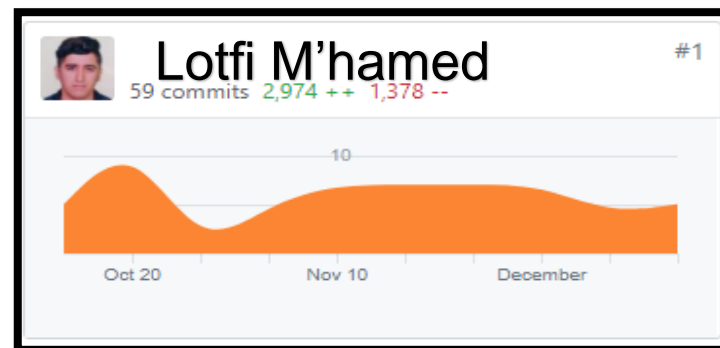
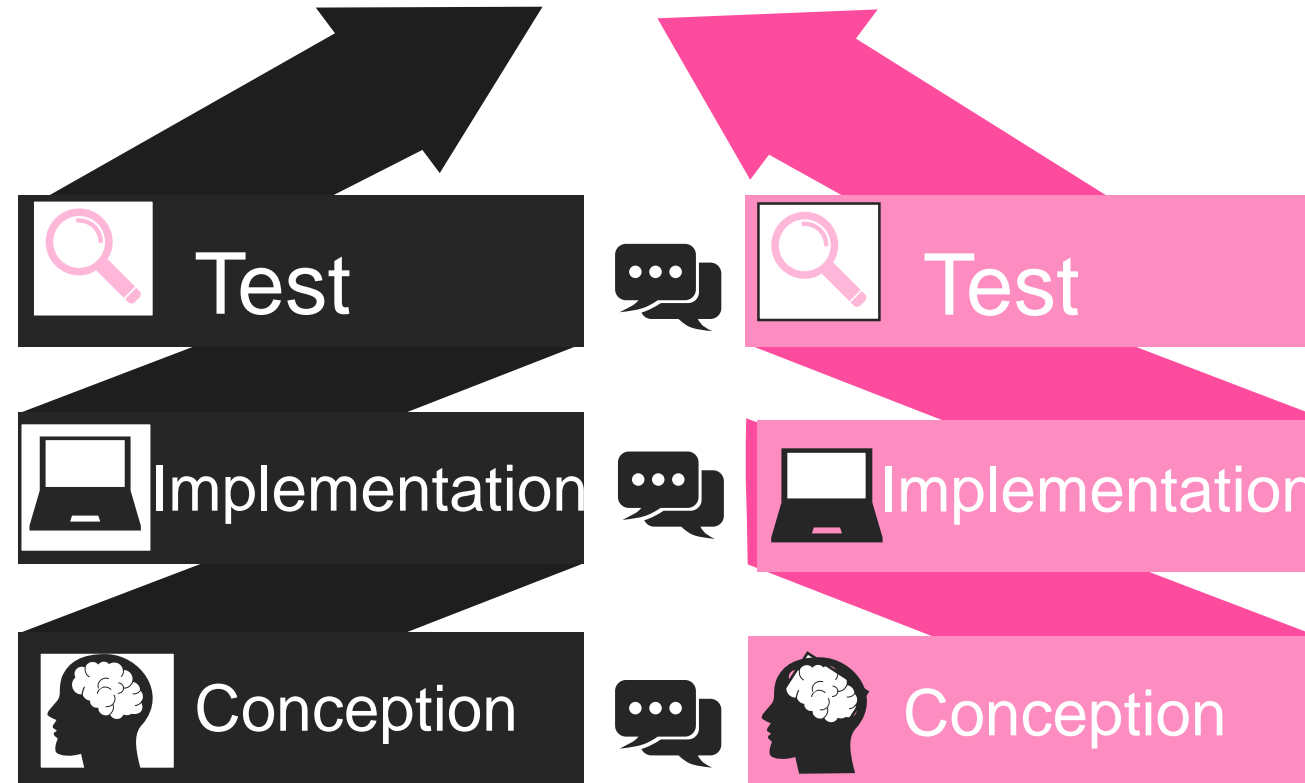


L'âge du Pierre (Stone Age)

2019/2020
L 3 INFO
GRP: SAE



Oct 13, 2019



Dec 15, 2019

LES ZONES

- ❖ Forêt
- ❖ Glissière
- ❖ Carrière
- ❖ Rivière
- ❖ Chasse
- ❖ Champ

LES CARTES

- ❖ Cartes De Bâtiments avec ressources imposées (20 cartes)

LES PHASES

- ❖ Placement
- ❖ Action
- ❖ Nutrition

LES ROBOTS

- ❖ RandomBot (gagne environs 2 parties parmi 500)
- ❖ HunterBot (gagne environs 0 parties parmi 500)
- ❖ SmartBot (gagne environs 498 parties parmi 500)

fonctionnalités manquantes

- Hutte
- Fabriquant d'outils
- Cartes bâtiments avec ressources aux choix du joueur
- Cartes civilisation
- Changement des règles du jeu selon nombre de joueurs
- faire des tests plus spécifique à chaque classe.



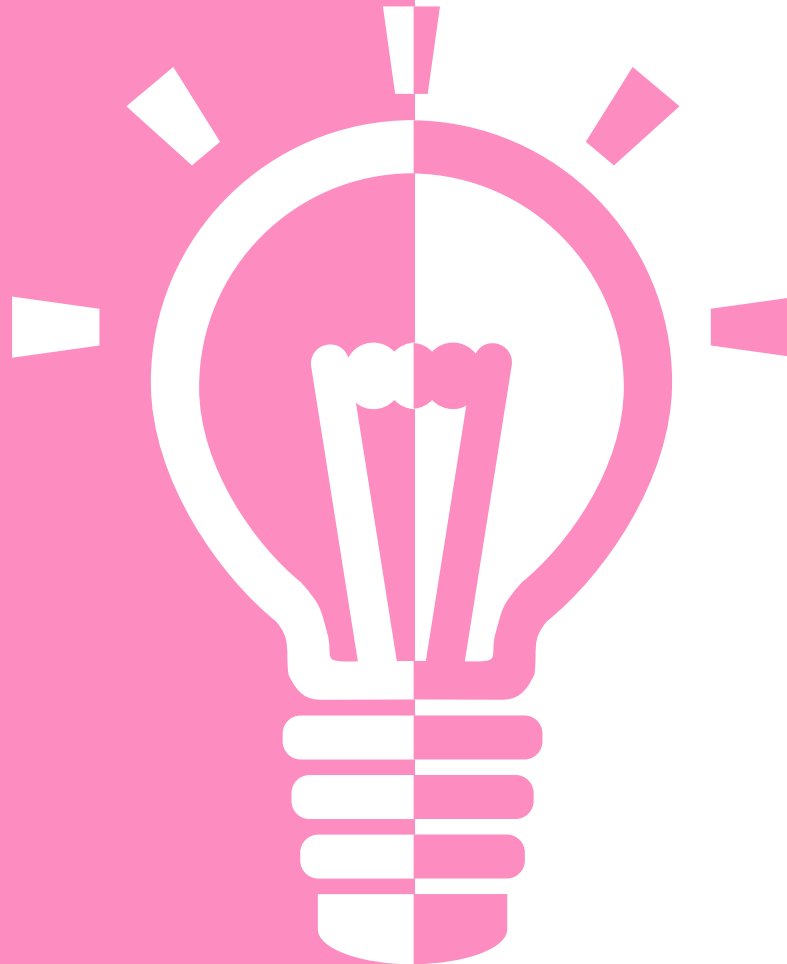
Choix de conceptions et organisation du code

Comment sont conçus les éléments du jeu ?

Création d'énumération pour la plupart des éléments du jeu (zones ,ressources)

Pourquoi ce choix ?

Structure simple à comprendre



- ☐ Paradigme objet
- ☐ Création de classes pour modéliser les éléments du jeu
- ☐ Interaction entre les classes via les paramètres passés dans les méthodes et/ou constructeurs
- ☐ Séparer en différents packages
- ☐ utilisation de l'héritage et le polymorphisme afin de minimiser et optimiser le code

Organisation Des Tests

- Comment sont testés les éléments du jeu ?
 - Test d'un élément avec un scénario nominal
 - Test avec d'autres scénario alternatifs

• Comment est testé le robot ?

- Chaque robot est spécialisé
- Les choix intelligents des joueurs ont été testé pour voir s'il respecte la stratégie mise en place tout en respectant les règles du jeu.
- On teste les choix du joueur Random pour voir s'il respecte les règles du jeu dans ses choix.
- Robot : HunterBot : Resultat attendu
SmartBot : Resultat attendu

Points Forts et Points Faibles

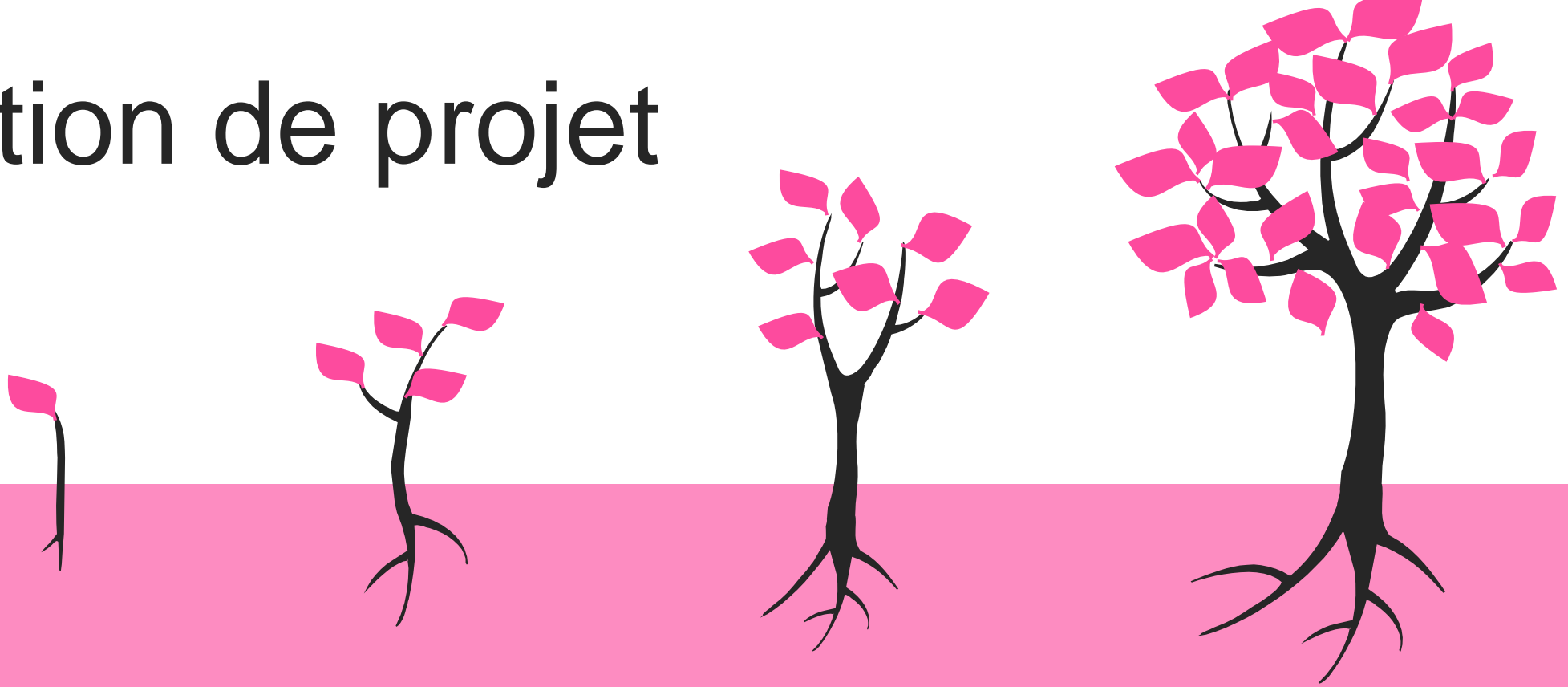
- Point Forts :

- Conception du moteur du jeu , du Area et du robot
- La conception des règles dans le jeu
- Les actions possibles des robots

- Point Faibles et/ou amélioration:

- La conception des Robots(il faut séparer le robot de son inventaire)
- La conception du moteur de jeu
 - Les tests en général, manque de scénario alternatifs et des tests specifique pour chaque classe
- code est un peu compliqué

Gestion de projet



- Ce projet a été réalisée sur 7 itérations suivant les méthodes agiles de gestion des projets .
- chaque itération est exécutable et testée
- Utilisation du GIT (gestion de version)
- Utilisation de JUnit pour les tests unitaires