

Rapport projet chasse au trésor

Année 2022-2023

Unité Programmation Objet Avancée

**UNIVERSIT^é DE
FRANCHE-COMT^é**

Sommaire

Rapport projet chasse au trésor.....	1
I – Présentation :.....	3
1 – Résumé :.....	3
2 – Choix :.....	3
3 – Extensions :.....	3
II – Conception :.....	4
III – Développement :.....	7
1 – Partage :.....	7
2 – Implantation :.....	7
3 – Image :.....	8
IV – Bilan :.....	9

I – Présentation :

1 – Résumé :

Le sujet de Programmation Objet Avancée de cette année consiste en un chasse au trésor sur un monde où des joueurs mobiles s'y déplacent afin de récupérer le trésor avec l'aide d'autres joueurs et/ou d'objets mis à sa disposition sur le monde.

2 – Choix :

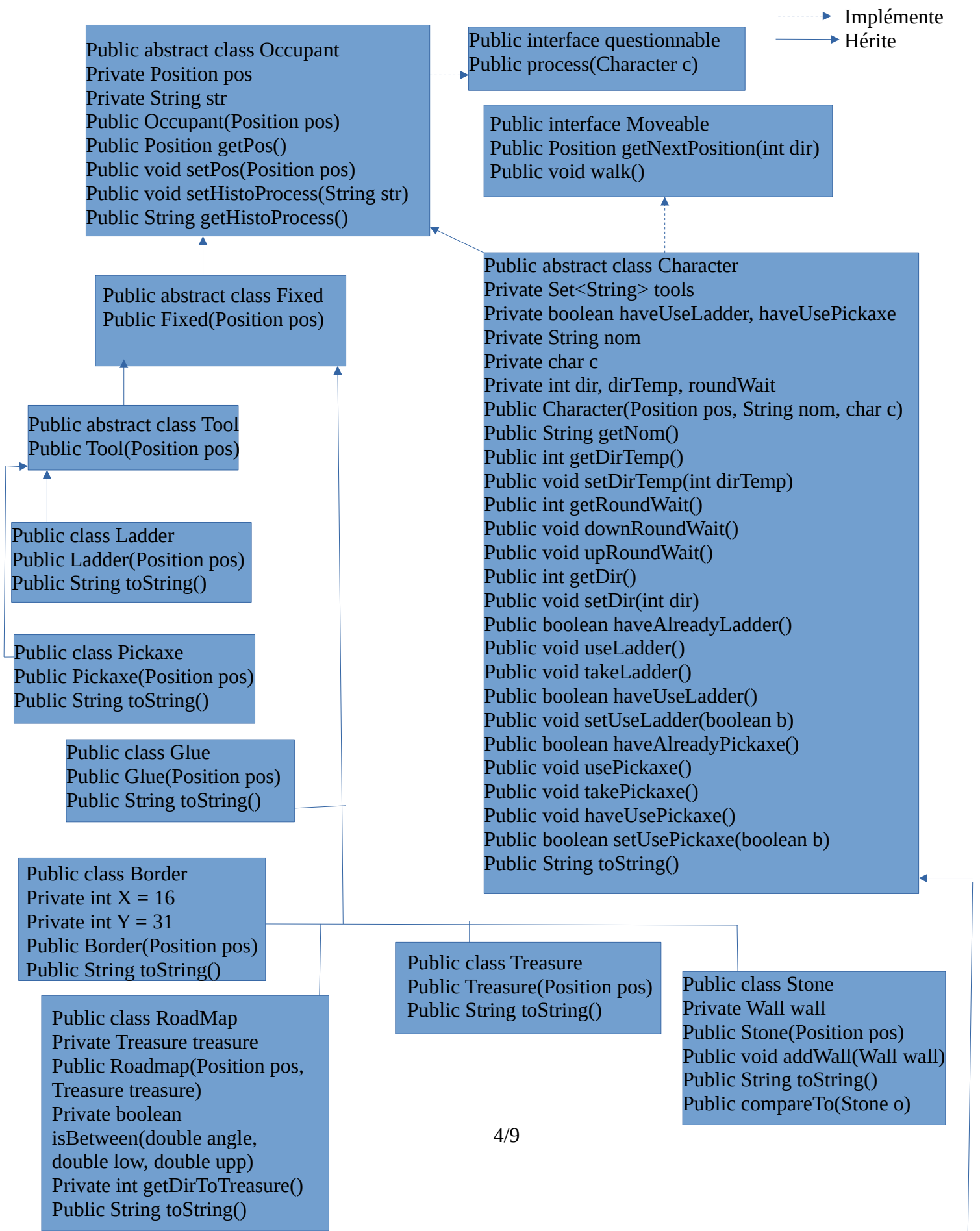
Nos choix par rapport au sujet sont rester les mêmes pour la simplicité afin de développer le jeu sans contrainte supplémentaires.

3 – Extensions :

L'extension dans notre jeu qui a été implantée est celle de la pioche, qui obtient comme avantage les inconvénients de l'échelle et inversement.

En effet, si un joueur rencontre un mur vertical alors qu'il a une direction horizontale la pioche sera plus efficace qu'une échelle. A l'inverse si le joueur a une direction horizontale et rencontre un mur horizontal, il va se retrouver bloqué de nouveau contre un mur lorsqu'il aura utilisé sa pioche.

II – Conception :



```

Public class Wise
Private Treasure treasure
Public Wise(Position pos, Treasure treasure)
Private boolean isBetween(double angle,
double low, double upp)
Private int getDirToTreasure()

```

```

Public class Cheater
Private Treasure treasure
Public Cheater(Position pos, Treasure
treasure)
Private boolean isBetween(double angle,
double low, double upp)
Private int getDirToTreasure()

```

```

Public class Hunter
Private boolean haveTreasure
Public Hunter(Position pos)
Public boolean haveTreasure()
Public void gotTreasure()

```

```

Public class Game
Private int round
Private List<Occupant> occupants
Private Grille grille
Private int qte_hunter, qte_wise, qte_cheater
Private ArrayList<String> prHunter = new ArrayList<String>()
Private ArrayList<String> prWise = new ArrayList<String>()
Private ArrayList<String> prCheater = new ArrayList<String>()
Private int qte_roadmap, qte_pickaxe, qte_ladder, qte_wall, qte_glue
Private int X = 16
Private int Y = 31
Private ArrayList<String> histo
Public Game(int qte_hunter, int qte_wise, int qte_cheater, int qte_roadmap, int qte_pickaxe, int qte_ladder, int
qte_wall, int qte_glue)
Public Game()
Public Game(ArrayList<Integer> stats)
Public ArrayList<Integer> getStats()
Private void tabString(ArrayList<String> pr, String src)
Public List<Occupant> getOccupants()
Public Grille getGrille()
Public void initialisation()
Private Hunter createHunter(char c)
Private Wise createWise(Treasure treasure, char c)
Private Cheater createCheater(Treasure treasure, char c)
Private Position getFreeRandomWallPosition()
Private Position getFreeRandomPosition()
Private Position getRandomPosition()
Private boolean posIsNextToStone(Position pos)
Private boolean posIsFree(Position pos)
Public boolean treasureIsFind()
Public int getRound()
Public ArrayList<String> getHistory()
Public void playARound()
Public void results()

```

```

Public class Grille
Private int X = 16
Private int Y = 31
Private Map<Position, List<Occupant>> grille
Public Grille(List<Occupant> occupants)
Public void afficher()
Public Occupant getOccupant(Position pos, int i)
Public void removeOccupant(Position pos, int i)
Public void addOccupant(Position pos, Occupant o)
Public int getSizeListOccupant(Position pos)

```

```

Public class Controleur
Private Game game
Private Window window
Private ArrayList<Integer> stats
Public Controleur(Game game, Window window)
Public void update(Grille grille)
Public void actionPerformed(ActionEvent e)

```

```

Public class Wall
Private LinkedList<Stone> stones
Private int orientation
Public Wall(int orientation)
Public void addStone(Stone stone)
Public LinkedList<Stone> getStones()
Public int getOrientation()

```

```

Public class Window
Private JScrollPane sp
Private JPanel histo
Private JButton buttonNewGame, buttonPlayManual,
buttonNextRound, buttonPlayAuto
Private JLabel[][] imageLabel
Public Window(int X, int Y)
Public Window()
Public void addListener(ActionListener listener)
Public void updateMap(ImageIcon image, int i, int j)
Public void enabledNewGame(boolean activer)
Public void enabledPlayManual(boolean activer)
Public void enabledPlayAuto(boolean activer)
Public void enabledNextRound(boolean activer)
Public void updateHistory(String message)
Public void cleanHistory()

```

III – Développement :

1 – Partage :

Ayant deux autres projets en même temps que celui de Programmation Objet Avancée, nous avons répartis le temps du projet en fonction des capacités sur les langages de chacun. Nous avons aussi utilisés l’outil GitHub pour partager notre travail en temps réel ainsi que d’avoir des messages qui explique ce qui a été fait sur les programmes.

2 – Implantation :

Nous avons implantés toutes les fonctionnalités énoncer dans le sujet en plus de la pioche citer plus haut. Voici la liste de ce qui a été implanté :

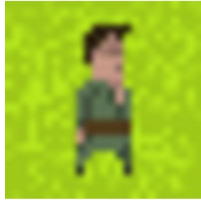
- Mur
- Carte
- Colle
- Trésor
- Bord
- Échelle
- Pioche
- Chasseur
- Sage
- Tricheur

3 – Image :

Pour la version graphique du sujet nous avons créés des images de nous-mêmes de 25*25 pour remplir les cases de l'interface graphique. En voici quelques-uns :



*Illustration 1
: Tricheur*



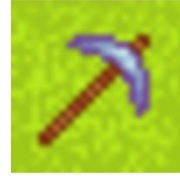
*Illustration 2 :
Chasseur*



*Illustration
3 : Sage*



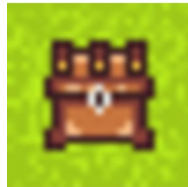
*Illustration 4
: Échelle*



*Illustration 6
: Pioche*



*Illustration
7 : Carte*



*Illustration
8 : Trésor*

IV – Bilan :

Pour conclure ce rapport, nous avons réussi à implémenter toutes les fonctionnalités que nous souhaitons faire et nous n'avons pas rencontrés de grandes difficultés.

Des améliorations pourraient être effectuées. En effet, pour la version graphique on pourrait faire qu'on puisse choisir la taille de la carte ainsi que le nombre d'éléments dessus. Une autre amélioration qui pourrait être faite est celle du rapport mur car lorsqu'un Character utilise une pioche en plein milieu d'un mur et qu'il va se retrouver dirigé aléatoirement par un autre Character, nouvelle direction qui le dirige vers le mur mais n'ayant pas de pioche ou d'échelle il se retrouve bloqué presque indéfiniment avant si un Character venait à nouveau le rediriger.