

# Rapport du projet MapMaker

KHADHRI Lina

FRANCHOMME Eliott

MEKKI Nesma

*TP12*

## Sommaire :

### I. Guide utilisateur

Règles du jeu, exécution du programme, option de sauvegarde

### II. État d'avancement du projet

Fonctionnalités réalisées, problèmes rencontrés, améliorations.

### III. Description des fonctions importantes

Pour la gestion des tuiles, du solveur et du défilement infini.

### IV. Organisation, répartition des tâches et ressentis

Outils collaboratifs, pourcentage des tâches.



## I. Guide utilisateur :

Le programme proposé est une implémentation d'une interface graphique de création de carte du monde et un algorithme de complétion automatique de carte en langage Python.

### Lancement du programme :

1. Assurez-vous d'avoir Python installé sur votre machine=μ.
2. Exécutez le fichier principal (mapmaker.py) en écrivant sur le terminal de votre machine :

```
python mapmaker.py.
```

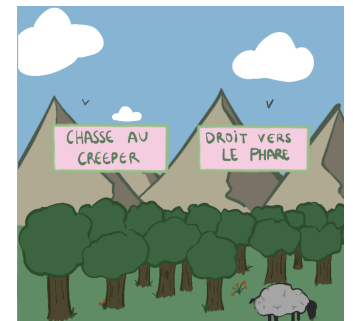
L'ensemble des fichiers nécessaires au bon fonctionnement du programme sont fournis dans le fichier `programme_mapmaker.zip`.

**Interface du menu principal :** L'interface lorsque l'utilisateur lance le programme constitue le menu principal du jeu **MapSaga™** et donne accès aux deux fonctionnalités du projet :

- MapMaker : Permet de générer manuellement ou automatiquement une carte composée de tuiles. L'utilisateur peut placer les tuiles à la main en sélectionnant les tuiles compatibles, ou bien lancer un solveur pour remplir automatiquement la carte selon les contraintes de raccordement.
- MapGames : Permet de jouer à des mini-jeux thématiques intégrés à la carte. Deux idées principales ont été développées :
  - *DROIT VERS LE PHARE* Un mini-jeu de course de bateaux, dans lequel l'utilisateur doit cliquer rapidement pour faire avancer son bateau.
  - Un jeu de chasse au creeper, où l'utilisateur doit retrouver un creeper dissimulé dans une carte générée.
- Quitter : Ferme proprement l'application

### Commandes à la souris et au clavier :

- Clic gauche : pour placer des tuiles sur la carte, et pour choisir les modes de jeux.
- Clavier : Touches fléchées pour déplacer la carte dans toutes les directions.
- Scroll : Pour choisir les tuiles dans la conception manuelle de la carte.





## II. État d'avancement du projet :

### Fonctionnalités réalisées :

- Génération interactive d'une carte à base de tuiles, avec comptabilité automatique entre les biomes
- Solveur (complétion automatique) de carte par backtracking respectant les tuiles déjà placées.
- Défilement infini de la carte dans toutes les directions, avec génération dynamique des nouvelles zones en utilisant le solveur
- Assistant de conception
- Ajout de décors adaptés au biomes (mer, plaine)

### Problèmes rencontrés :

- Ralentissements lors du solveur et du défilement rapide, notamment avec de grandes cartes
- Problèmes de superposition de décors (résolu en ajustant le système de placement)

### Améliorations réalisées :

- Ajout de nouveaux types de tuiles et de décors
- Optimisation du solveur pour améliorer les performances
- Modes de jeux, la *Chasse au Creeper* et la *Course de Bateaux*

## III. Description des fonctions importantes :

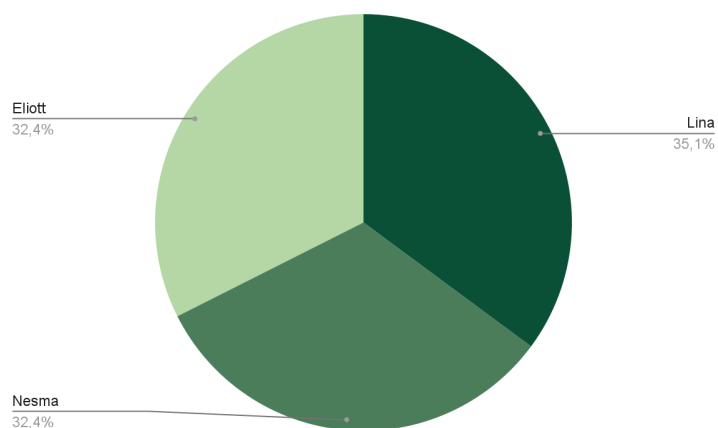
- **tuiles\_possibles(grille, i, j)** : Elle détermine toutes les tuiles valides qu'on peut placer à une position (i, j) selon les tuiles voisines (haut, bas, gauche, droite). Elle a pour particularité de prendre en compte les raccords entre biomes et permet de vérifier la continuité et d'empêcher les blocages des rivières.
- **solveur2(grille, dico, coos)** : Le solveur remplit automatiquement la grille de façon valide en utilisant un algorithme de backtracking, en choisissant grâce à la fonction `case_plus_contrainte(grille)` la case la plus difficile à rem . En cas d'échec répété, la fonction `effacer_autour()` nettoie les cases autour pour débloquer la situation.
- **decor(...)** : Elle ajoute des décors visuels sur les tuiles (bateaux, châteaux, etc...) en sélectionnant aléatoirement un décor approprié selon le type de tuile.
- **scroll(direction)** : C'est la fonction centrale du défilement infini, qui est appelée lorsqu'on appuie sur les touches fléchées du clavier et qui permet de déplacer la carte dans la direction choisie et par conséquent d'appeler le solveur pour générer dynamiquement de nouvelles tuiles aux extrémités, tout en gardant en mémoire les parties précédemment visitées.



## IV. Organisation du travail :

**Communication et synchronisation du groupe :** Pour coordonner les tâches et partager les ressources, nous avons utilisé GitHub et Discord pour la communication quotidienne. On a pu donc suivre les avancées de chacun et résoudre les bugs ou incompréhensions rapidement.

### Répartition des tâches :



**Lina :** Conception graphique, aide à l'implémentation du défilement infini, création des mini-jeux MapGames.

**Elliott :** Gestion des décors et des rivières, implémentation de l'assistant de conception et implémentation du défilement infini.

**Nesma :** Cheffe de projet, implémentation du solveur et aide à l'implémentation du défilement infini, rédaction du rapport.

## Conclusion :

Ce projet nous a beaucoup inspirés, notamment par la liberté artistique qu'il permettait. Bien que l'aspect graphique ne soit pas exigé dans l'énoncé, nous avons pris plaisir à enrichir visuellement le jeu avec une interface personnalisée, toutes les tuiles et les décors dessinés à la main, et un menu interactif. Cette dimension artistique, portée en grande partie par Lina, donne une identité forte à notre projet.

La répartition des tâches a également été un point très positif de notre groupe; en effet chacun a pu s'approprier une partie du projet (que ce soit le graphisme, la logique du jeu, le solveur, l'affichage etc...) et nous a donc permis d'avancer de manière fluide, efficace et surtout collaborative. La cohésion du groupe a été un vrai moteur tout au long du développement.

S'il fallait relever un point négatif, ce serait peut-être le manque de temps. Nous avons encore de nombreuses idées en tête (ajout d'autres modes de jeux, possibilité de confection d'avatars implémentés ensuite au décor, animations plus poussées, création de nouveaux biomes, etc...) mais nous avons dû prioriser l'essentiel.

Ce projet nous a permis de consolider nos compétences en programmation, en gestion de projet, mais aussi de redécouvrir la richesse du travail en équipe sur un jeu à la fois technique et créatif.