

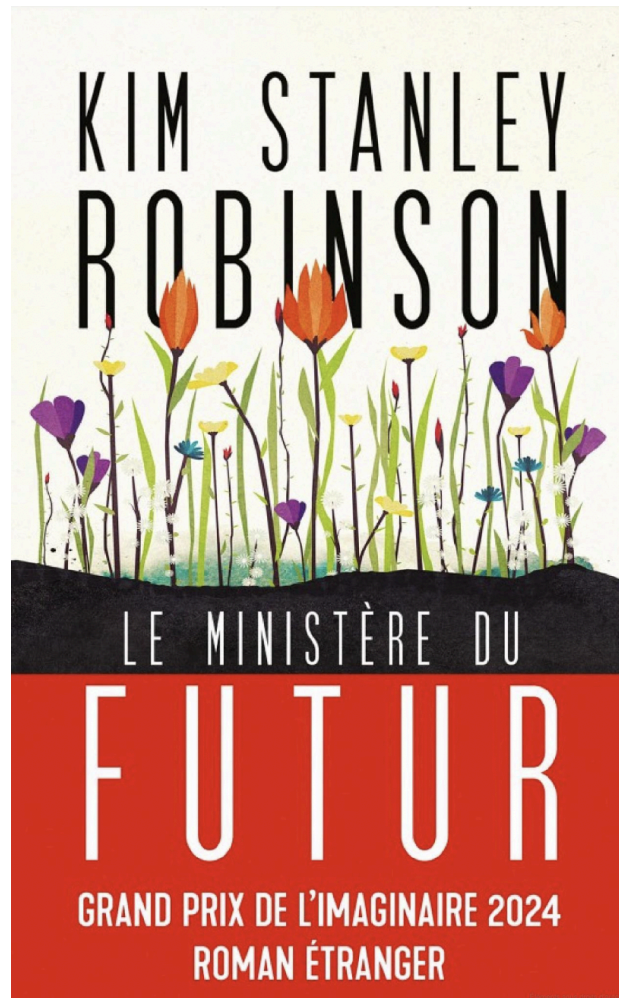
Timothée Crouzet / Merwan Bekkouche

GitHub : <https://github.com/merwan-bekkouche/TBA>

Vidéo : <https://github.com/merwan-bekkouche/TBA/blob/main/DemoTBA.mov>

Le ministère du futur

Notre jeu s'inspire du début du roman de science-fiction de Kim Stanley Robinson, quand quelque part en Inde la température atteint 35°C avec un taux d'humidité de 100%. Il est alors impossible de survivre dehors plus de six heures (wet-bulb temperature).



Guide utilisateur

Une fois le jeu lancé et votre nom saisi, vous vous retrouvez dans votre appartement à 6h du matin. Vous savez qu'il faut survivre jusqu'à 18h, après l'air ne sera plus respirable et la mort s'en suivra.

En jaune, la signe status indique l'heure, la température, le taux d'humidité qui restera à 100%, vos points de vie (HP), l'eau que vous transportez (quand en négatif vous commencez à avoir très soif), la charge que vous transportez et vos points d'expériences.

Le curseur vous invite à saisir une commande.

Commandes

La flèche vers le haut et le bas permettent de remonter dans l'historique des dix dernières commandes effectuées.

Déplacements

Pour vous déplacer, saisissez le numéro de l'action choisie. À chacun de vos déplacements, le temps avance de 20 minutes, la température s'élève, vous consommez de l'eau.

Help

Affiche la liste des commandes disponibles.

Quit

Quitter le jeu.

Back

Retourner dans le lieu précédent.

History

Obtenir l'historique du parcours effectué.

Look

Regarder autour de vous.

Take <nom>

Prendre un objet révélé par Look (take all pour tout prendre). Impossible de prendre plus d'objet que votre charge autorisée. Les objets avec un poids nul ne peuvent être emportés.

Drop <nom>

Déposer un objet.

Check

Afficher son statut et vérifier l'inventaire.

Talk

Parler à un NPC (il n'y a jamais plus d'un NPC dans un lieu)

Use <nom>

Utiliser un objet. Par exemple, un kit médical pour gagner des points de vie (on peut dépasser son score initial).

Wait

Attendre 2 heures sans bouger.

Fight

Continuer le combat. C'est toujours les NPC qui initient un combat. Vous pouvez fuir la zone de combat. Seuls certains NPC se déplacent de lieu en lieu.

Conditions de victoire

Pour gagner, il faut survivre jusqu'à 18h, ce qui est impossible en extérieur ou dans un lieu non climatisé. Pour sauver toute la ville, il faut rétablir le courant. Il est possible de gagner avec 500, 1000, 1500 ou 2000 XP (score maximal, si vous sauvez la ville).

Il existe a trois façon de gagner :

1. Faire équipe avec le collègue humanitaire et ramener le moteur dans l'appartement, ce qui limite la température et permet d'attendre 18h. Petite victoire à 500 XP.
2. Toujours en équipe se rendre chez Monk (level 2), rejoindre le QG de gardes, où il y a une clim et y rester. Victoire tout aussi égoïste que la première, cette fois à 1 000 XP.
3. Emprunter le tunnel pour gagner le level 3 et rétablir l'électricité dans la ville. Victoire à 2000 XP.

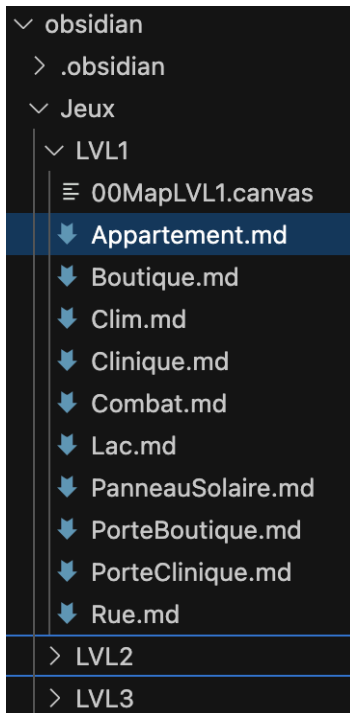
Guide développeur

Obsidian

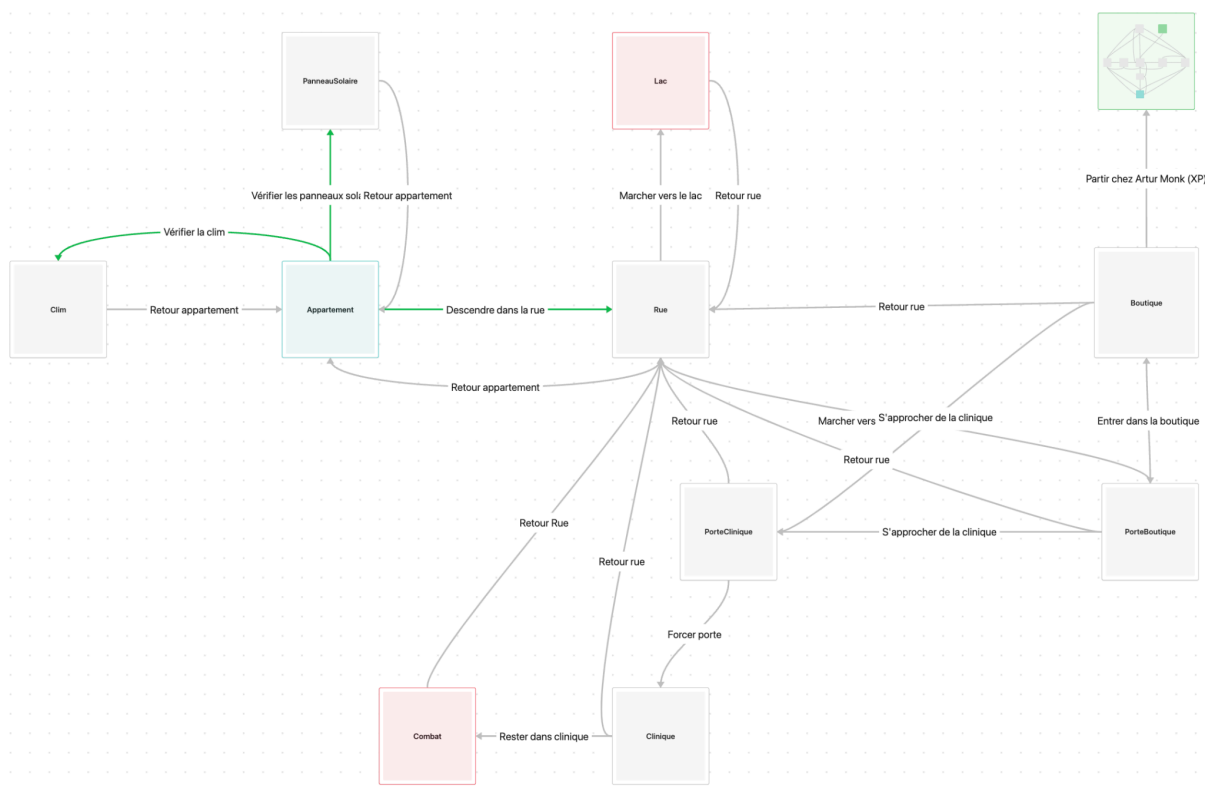
Obsidian est un logiciel libre de prise de notes, mais aussi d'écriture, de plus en plus utilisé par les écrivains et scénaristes. Il travaille avec des fichiers textes au format Markdown. Il dispose d'un plugin Canvas qui permet de créer des MinMap en interconnectant les divers fichiers Markdown (les nodes), tout en nommant les liaisons entre eux (edges).

Nous avons créé trois levels interconnectés avec chacun une dizaine de rooms (donc autant de fichiers Markdown). Cette approche nous a permis de travailler l'histoire indépendamment du code et de la faire évoluer avec souplesse et rapidité. Il est ainsi possible de développer l'histoire quasi à l'infini sans s'occuper du code, ou presque.

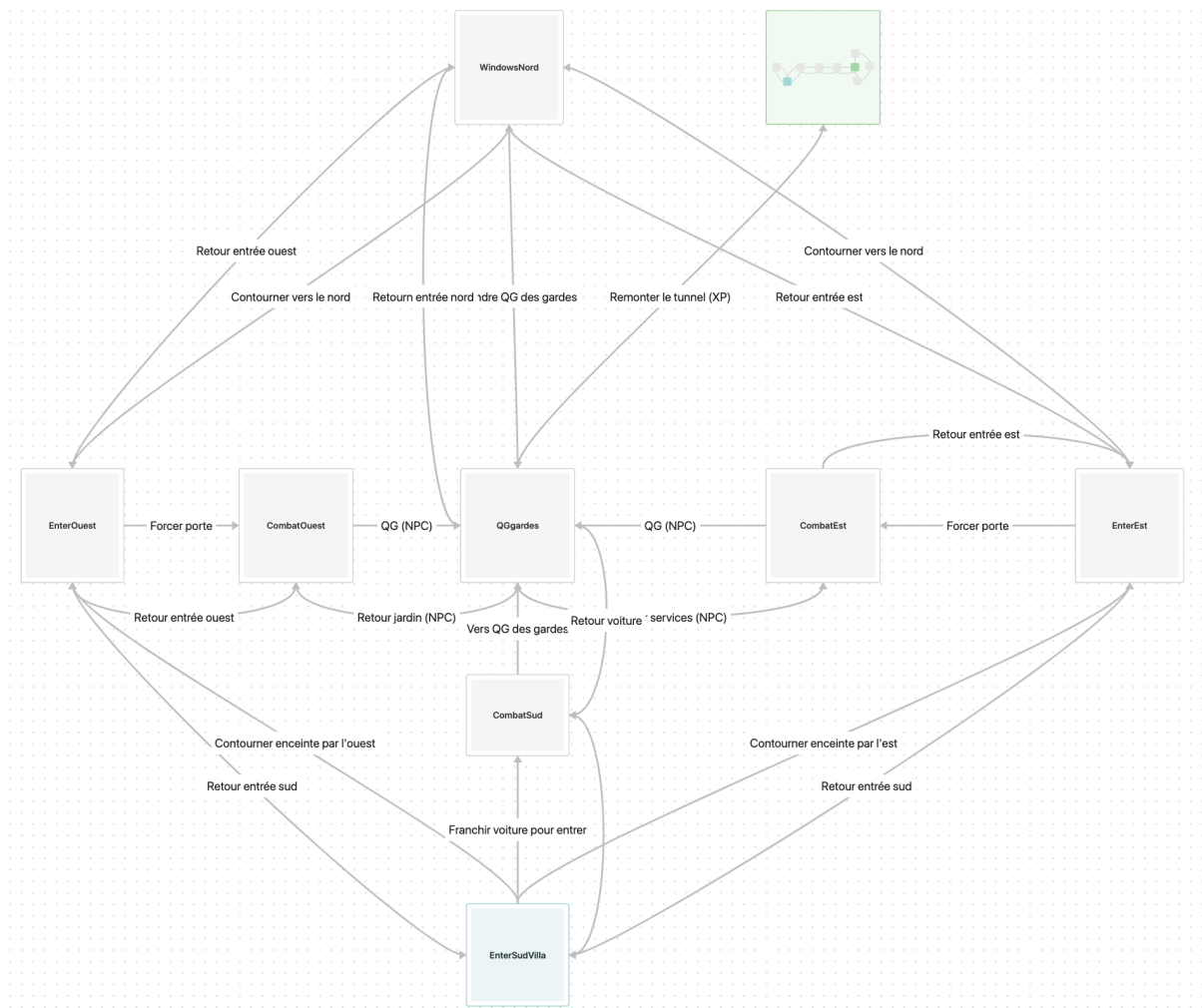
Les fichiers



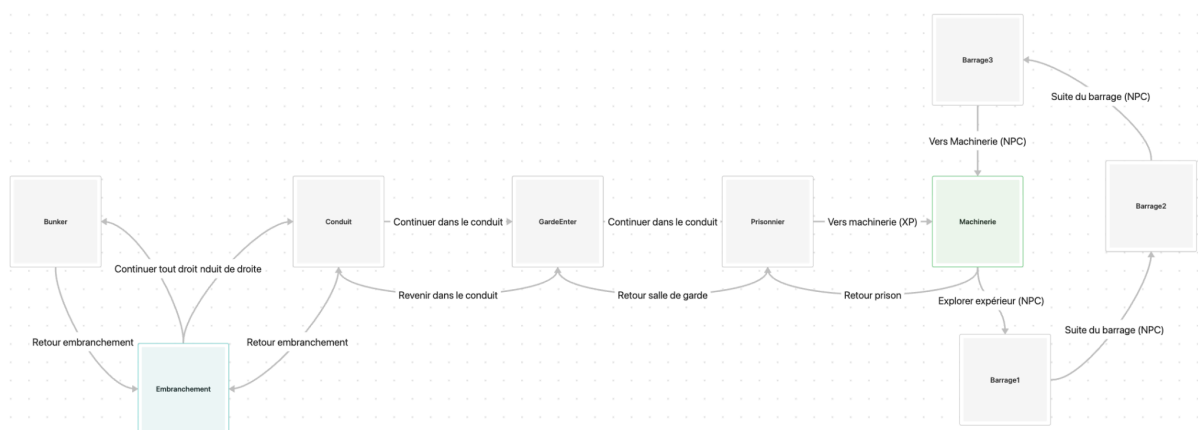
Level 1



Level 2



Level 3



JSON

Les trois Canvas interconnectés sont des fichiers json. Le dictionnaire possède une clé nodes contenant le tableau des nodes (les fichiers Markdown/room) et une clé edges décrivant leurs liaisons. Un node peut pointer vers un autre canvas. Dans ce cas, la room de départ dans ce canvas est définie par le fichier Markdown qui dispose du tag #start.

```
{
  "nodes": [
    {"id": "81295fba74d5c4cd", "type": "file", "file": "Jeux/LVL1/Rue.md", "x": 480, "y": -260, "width": 400, "height": 400},
    {"id": "39335f55dcc311bb", "type": "file", "file": "Jeux/LVL1/Appartement.md", "x": -880, "y": -260, "width": 400, "height": 400, "color": "5"},
    {"id": "e3bacb4035c7dd0c", "type": "file", "file": "Jeux/LVL1/PanneauSolaire.md", "x": -880, "y": -1200, "width": 400, "height": 400},
    {"id": "f900d7174e8cd387", "type": "file", "file": "Jeux/LVL1/Lac.md", "x": 480, "y": -1200, "width": 400, "height": 400, "color": "1"},
    {"id": "8118d543b8424087", "type": "file", "file": "Jeux/LVL1/Clim.md", "x": -2000, "y": -260, "width": 400, "height": 400},
    {"id": "42dbd7812fc9f349", "type": "file", "file": "Jeux/LVL1/Clinique.md", "x": 480, "y": 1495, "width": 400, "height": 400},
    {"id": "fb59ae0af358e4c5", "type": "file", "file": "Jeux/LVL1/PorteClinique.md", "x": 760, "y": 655, "width": 400, "height": 400},
    {"id": "e8f41f369aab9826", "type": "file", "file": "Jeux/LVL1/PorteBoutique.md", "x": 2496, "y": 655, "width": 400, "height": 400},
    {"id": "18d38073ef610559", "type": "file", "file": "Jeux/LVL1/Combat.md", "x": -480, "y": 1495, "width": 400, "height": 400, "color": "1"},
    {"id": "67f1a5ecd53a4ed5", "type": "file", "file": "Jeux/LVL2/00MapLVL2.canvas", "x": 2480, "y": -1280, "width": 400, "height": 400, "color": "4"},
    {"id": "957b150856efc173", "type": "file", "file": "Jeux/LVL1/Boutique.md", "x": 2465, "y": -315, "width": 431, "height": 455}
  ],
  "edges": [
    {"id": "038559f2ae4ada84", "fromNode": "39335f55dcc311bb", "fromSide": "top", "toNode": "e3bacb4035c7dd0c", "toSide": "bottom", "color": "4", "label": "Vér"},
    {"id": "905a82e3fd71ee84", "fromNode": "39335f55dcc311bb", "fromSide": "right", "toNode": "81295fba74d5c4cd", "toSide": "left", "color": "4", "label": "Des"},
    {"id": "7d397f820c77d1f6", "fromNode": "39335f55dcc311bb", "fromSide": "top", "toNode": "8118d543b8424087", "toSide": "top", "color": "4", "label": "Vérifi"},
    {"id": "d67e5db6dbd3236", "fromNode": "81295fba74d5c4cd", "fromSide": "bottom", "toNode": "e8f41f369aab9826", "toSide": "top", "label": "Marcher vers la"},
    {"id": "b6176b78e3a66c36", "fromNode": "81295fba74d5c4cd", "fromSide": "top", "toNode": "f900d7174e8cd387", "toSide": "bottom", "label": "Marcher vers le"},
    {"id": "1e61cb8dba07d49e", "fromNode": "e8f41f369aab9826", "fromSide": "top", "toNode": "957b150856efc173", "toSide": "bottom", "label": "Entrer dans la"},
    {"id": "9150c8ef5a41d26f", "fromNode": "e8f41f369aab9826", "fromSide": "left", "toNode": "fb59ae0af358e4c5", "toSide": "right", "label": "S'approcher de"},
    {"id": "b3f0998be855045a", "fromNode": "fb59ae0af358e4c5", "fromSide": "bottom", "toNode": "42dbd7812fc9f349", "toSide": "top", "label": "Forcer porte"},
  ]
}
```

Appartement

La canicule se prolonge depuis dix jours, causant de plus en plus de victimes. Beaucoup de gens ont fui la ville, d'autres se réfugient dans les appartements climatisés. Les prévisions annoncent 35°C à partir de midi avec un taux d'humidité de 100 %, soit un seuil critique à partir duquel le corps humain ne peut plus se refroidir par transpiration. Dans ces conditions, la mort survient en quelques heures. Il faut tenir jusqu'à 18 heures pour passer le pic de chaleur.

Vous êtes un humanitaire, envoyé pour aider la population locale. La veille, vous avez refusé de quitter la ville, décidé à trouver une solution. Vous vous réveillez difficilement. Votre climatiseur ne fonctionne plus. Aucun bruit ne vous parvient, pas même celui du réfrigérateur. Votre appartement n'est plus alimenté en énergie, rien n'arrive depuis le secteur ni depuis vos panneaux solaires. Plus d'eau au robinet.

Vous pensez à tous ceux qui mourront si vous ne trouvez pas une solution pour rétablir l'électricité.

Pour commencer, serez-vous capable de sauver votre peau ? Dans 12 heures il sera trop tard.

- Eau | 3 litres d'eau | 3
- Mobile | Téléphone mobile | 0.2 | La batterie est quasi à 100%, mais aucun signal. Pas de Wifi, pas de réseau cellulaire. Impossible de contacter les autres humanitaires. Vous ne pouvez qu'envoyer un SOS d'urgence via satellite. Mais vous savez que personne ne pourra vous secourir.
- Céréales | 5 barres de céréales | 0.3 | Vous récupérez 25HP
- Kit | Kit de premier soin | 0.5 | Vous récupérez 50HP

#start

Il s'agit d'un simple fichier texte, qui peut être édité dans Obsidian comme dans VSC. Le titre de niveau 1 (#) donne le nom au lieu, affiché en rouge dans le jeu. Après le titre, on trouve le texte de présentation.

Nous utilisons les listes pour décrire les objets/NPC présents dans un lieu. Les pipes séparent le nom de l'objet, sa description, son poids, son usage éventuel.

La rue

La rue est quasi déserte, avec des voitures abandonnées au milieu, certaines avec le capot ouvert.

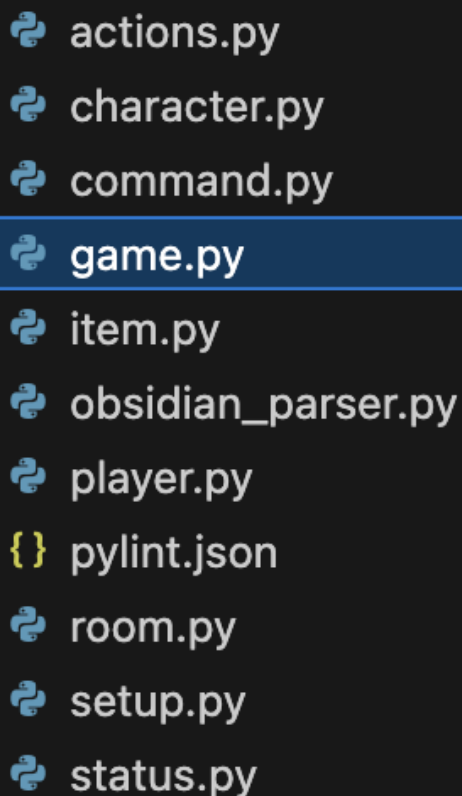
- Voiture | Une voiture électrique, capot ouvert. Batterie HS. | 0
- NPC | Mourant | Un homme de dos, allongé sur le côté gauche, dans la position PLS (position latérale de sécurité). Il respire difficilement. | 30 | 1 | MainsNues | Le mourant gémit. | Le mourant se redresse et tente de vous dépouiller

On décrit les NPC comme les objets, mais avec plus de paramètres. Après la description, on trouve les points de vie, puis un flag à 1 quand le NPC est statique, à 0 quand il peut se déplacer de lieu en lieu, suit son armement éventuel, puis ce qu'il est susceptible de dire avant de se mettre à combattre.

Edges

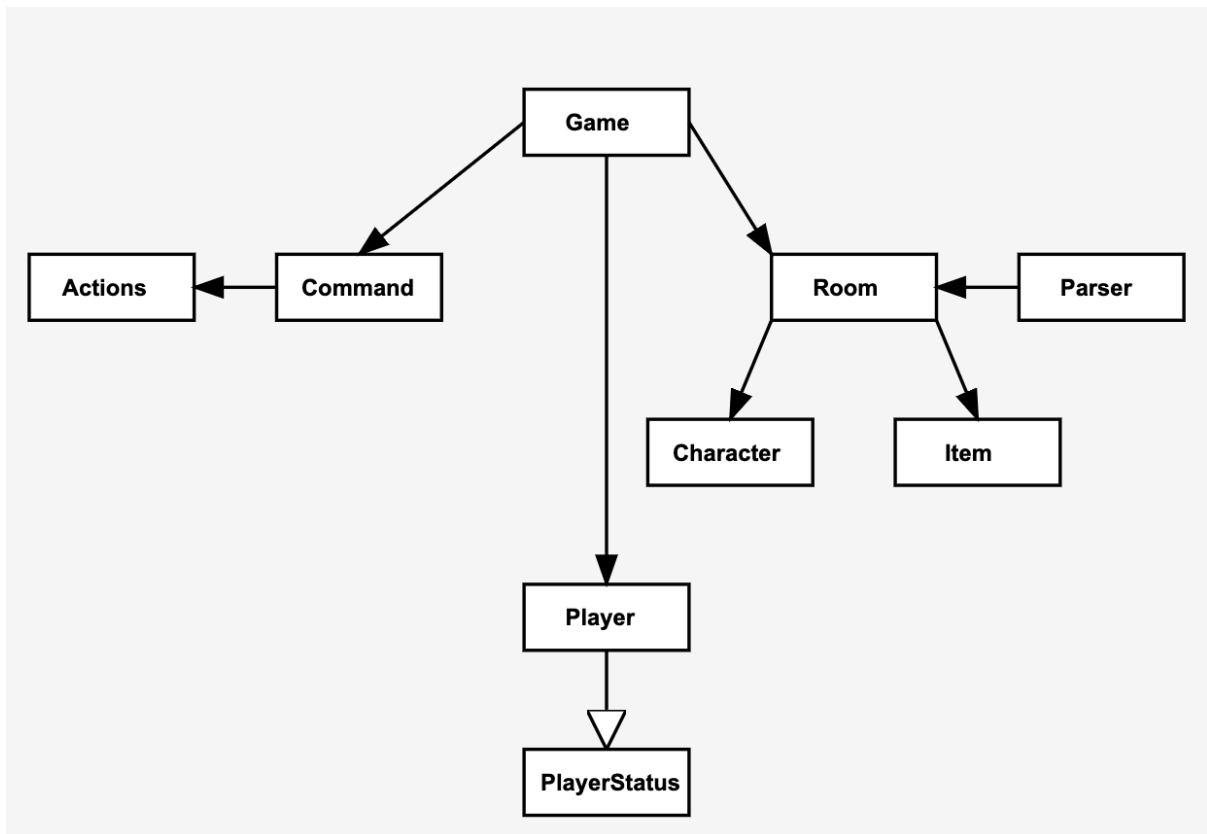
Les liens entre les boîtes du MindMap possèdent un label qui décrit l'action possible. Quand le texte comprend (NPC), un NPC mobile peut emprunter aléatoirement ce chemin. Les NPC sont déplacés aléatoirement chaque fois que le joueur se déplace. Si le label comprend (XP) le joueur récupère 500 XP, quand il passe par là pour la première fois.

Classes Python



```
actions.py
character.py
command.py
game.py
item.py
obsidian_parser.py
player.py
pylint.json
room.py
setup.py
status.py
```


- game.py : Game (point d'entrée du code, appelle les autres classes)
- command.py : Command (définir liaison entre Game et Actions)
- actions.py : Actions (actions possibles dans le jeu)
- room.py : Room (description des différents lieux du jeu)
- character.py : Character (dans les rooms il peut y avoir des NPC)
- item.py : Item (dans les rooms/ou l'inventaire du joueur)
- player.py : Player (définit le joueur, sa localisation, son inventaire, son historique...)
- status.py : PlayerStatus (extension de Player pour suivre ses HP et autres paramètres, comme expérience, HP, soif...).
- obsidian_parser.py : Parser (traduit le scénario écrit sous Obsidian en rooms, chacune avec des objets, des NPC, un titre, une description, diverses voies de sortie possibles...). Permet de séparer strictement le code des données qu'il utilise. A permis de construire une histoire un peu plus complexe que si les rooms/NPC/objets étaient décrits en dur dans le code.
- setup.py : juste une constante pour activer ou nom de mode DEBUG.



Perspectives de développement

- Des NPC beaucoup plus proactifs. On a créé la possibilité de dialoguer avant de combattre ou la possibilité qu'un NPC fasse équipe avec le joueur et double ses points de vie et sa charge possible. On aurait pu créer des NPC qui révèlent des

rooms cachés (par exemple au level 2 le capitaine des gardes aurait pu révéler la porte du tunnel à la fin du dialogue).

- Des objets plus interactifs. Quand une clim et le moteur se retrouvent dans une même pièce, la température ne peut pas dépasser les 30°. Par exemple, si la clim de la clinique est transportée dans l'appartement et déposée à côté de la clim existante, il suffit d'attendre pour gagner la partie. On aurait pu créer bien d'autres interactions entre les objets. Par exemple, entre des clés et des serrures.
- Une représentation graphique du parcours effectué.
- Un historique interactif.
- Une sauvegarde pour reprendre la partie.