Soutenance du projet 3: Aidez MacGyver à s'échapper!

Elève: Vincent Cailly
Mentor du projet: Stéphane Nédélec
Mentor de soutenance: Alexandre Iwanesko
Code source: https://github.com/Totobriac/McGyver

Lors de la réalisation de ce projet, je suis passé par une phase de réflexion, et penser en Programmation Orientée Objet.
J'ai identifié deux classes:

- la classe 'Map', qui regroupe tout ce qui touche au plateau de jeu
- la classe 'Player', qui est en charge du joueur et des mécaniques de jeu

Création du niveau:

Tout d'abord j'ai crée un niveau sous forme de fichier .txt, les '0' représentants les couloirs, les 'X' représentants les murs.
Il m'a donc ensuite suffit de charger le fichier .txt et ensuite effectuer une itération dessus et créer quatre listes. Une avec les murs, une avec les couloirs, une avec le garde, une avec le hero.
Créer une liste avec le garde à l'avantage de pouvoir si besoin est, rajouter plus de gardes.
J'ai aussi fait le choix de laisser le héro et le garde à une position fixe et ai donc aussi du mettre leurs positions dans la liste des couloir afin de ne pas créer un bug où le héro pare exemple ne pourrais plus passer par sa position de départ.

Positionnement des objets:

Au début de la partie un algorithme détermine la position des trois objets.
Ayant les listes des emplacements où le positionnement est impossible (murs, garde, hero) je n'ai donc plus qu'à peupler cette liste en choisissant au hasard 3 positions se trouvant dans la liste des couloirs mais pas dans celle du garde et du hero.
Je suis tombé face à ma première petite complication, quelques fois les positions choisies étaient les même. J'ai du donc réécrire mon algorithme avec une boucle "while" et mettre en position impossible les positions déjà présentes dans ma liste d'objets.

Gestion des déplacements:

Cela fut la partie la plus compliquée pour moi, même si sa logique est simplissime.
Chaque position étant une liste [x,y], il suffisait de vérifier si par exemple [x+1,y] (avancer une case à droite) était ou non dans la liste des couloirs.

Malheureusement mon algorithme, en essayant de vérifier si le mouvement était permis, effectuait le mouvement.
A ce stade le jeu était uniquement en mode console, et me renvoyait la position du héro ainsi qu'un booléen qui m'indiquait si le mouvement était possible où non.
Donc le jeu m'indiquait que le mouvement était impossible mais le héro y allait quand même!
Je suis donc passé par une valeure intermédiaire mais le résultat était le même.
Après beaucoup de tâtonnements, j'y suis arrivé en utilisant un slice [:] sur ma liste [x:y] ce qui m'a permis de copier la valeur de la position du héro et non la position du hero.
Ensuite je soumettais à mon algorithme cette valeur qui me renvoyait si oui ou non le mouvement était possible.

Récupération des objets:

J'ai créé une liste vide qui se peuplait chaque fois que la position de héro se trouvait dans la liste des objets. J'ai du être vigilant et ajouter une condition pour que la liste de rajoute pas plusieurs fois le même objet si on passe des multiples fois sur la même position incluse dans la liste des objets.

Conditions de victoire:

Les objets à ramasser étant 3, si on se présente sur une position incluse dans la list de gardes et que la longueur de notre liste objets est inférieure à 3, nous mourrons. Si elle est égale à trois, nous gagnons.

Implantation de l'interface graphique:

A ce stade le jeu était 100% jouable en mode console, le joueur se déplacant à l'aide d'input.
Pour passer en mode graphique j'ai initialisé à l'aide de Pygame une fenêtre de 450*450 pixels. 30px étant la largeur souhaitée par chaque case; 15 étant le nombre de case par côtés de mon niveau.

Je n'avais jamais utilisé Pygame auparavant, mais j'ai rapidement saisis sa logique. Chaque case est affichée une par une et lors d'un mouvement, tout est rafraîchi et l'élément mouvant change de position.

Quand un objet disparaît, comme quand MacGyver en ramasse un, il est simplement enlevé de la liste des objets à apparaître lors du prochain rafraichissement.

Le seul soucis lors de la partie de l'implantation de l'interface graphique fut la mise en place des objets, dans un premier temps les trois objets étaient les mêmes.
Une modification de mon algorithme m'a permis de pouvoir afficher chaque objet avec son propre icône.