Projet tutoré - Étude préalable

S22 : Jeu d’aventure textuel

Lucas GERARD - Laura TRIVINO - Loïc LEPRIEUR - Louis ZWAVIAK
S3B

Sommaire :

1. Présentation du projet

Le sujet “jeu d’aventures textuelles” n’est pas qu’un jeu mais également un éditeur de jeu. Dans l’interpréteur (qui interprète le script), le joueur interagit avec le jeu grâce a une zone de texte dans laquelle il décrit à l’aide du clavier des actions qu’il souhaite exécuter. Ces instructions seront sous la forme de phrases simples impératives : par exemple “Prendre bâton”.

Nous développerons donc le projet avec : Un **éditeur de jeu** (qui créera un script, qui servira à jouer) et un **interpréteur** (qui exécutera un niveau généré par un script édité dans l'éditeur) dans lequel sera inclus un analyseur syntaxique.

L’interpréteur consiste à charger un niveau créé a partir de l’éditeur. Chaque jeu permet au joueur d’évoluer dans le niveau grâce a une zone de saisie de texte dans laquelle il entre les actions qu’il souhaite exécuter (instructions) à l’aide du clavier. Ses instructions sont analysées par l’analyseur syntaxique inclus dans l’interpréteur de jeu, l’analyseur va permettre de définir la conséquence de l’instruction donnée par le joueur.

Concernant l’analyseur, si l’instruction donnée par le joueur est invalide, un message d’erreur du type “Hey j’ai pas compris, peux tu reformuler ?”, sera envoyé sur l’interface de jeu pour avertir le joueur et il pourra taper de nouvelles instructions. Si l’instruction est correcte et que les conditions pour pouvoir exécuter les instructions sont remplies, l’interpréteur va effectuer la/les conséquence(s) correspondante(s) qui vont consister à des changements sur l’environnement ou/et sur le personnage joué.

L'éditeur de jeu sera une interface simpliste et intuitive dans laquelle l’utilisateur (ici appelé concepteur) crée un niveau. Créer un niveau consiste à : définir l’environnement (La carte), les salles (se trouvant dans la carte), les objets et leurs placements (dans quelle salle les objets se situent et à quel endroit les objet sont positionnés dans la salle) et les actions (composées de conditions et de conséquences qui agissent sur l’environnement et/ou un objet et/ou sur l’inventaire du joueur). Nous verrons par la suite comment le concepteur peut éditer un jeu dans l’éditeur à l’aide d’une interface intuitive et simpliste (le but étant de permettre à un non professionnel en développement informatique de pouvoir créer son propre jeu).

En résumé, on peut lister les définitions des éléments structurant l'application dans le lexique ci-dessous.

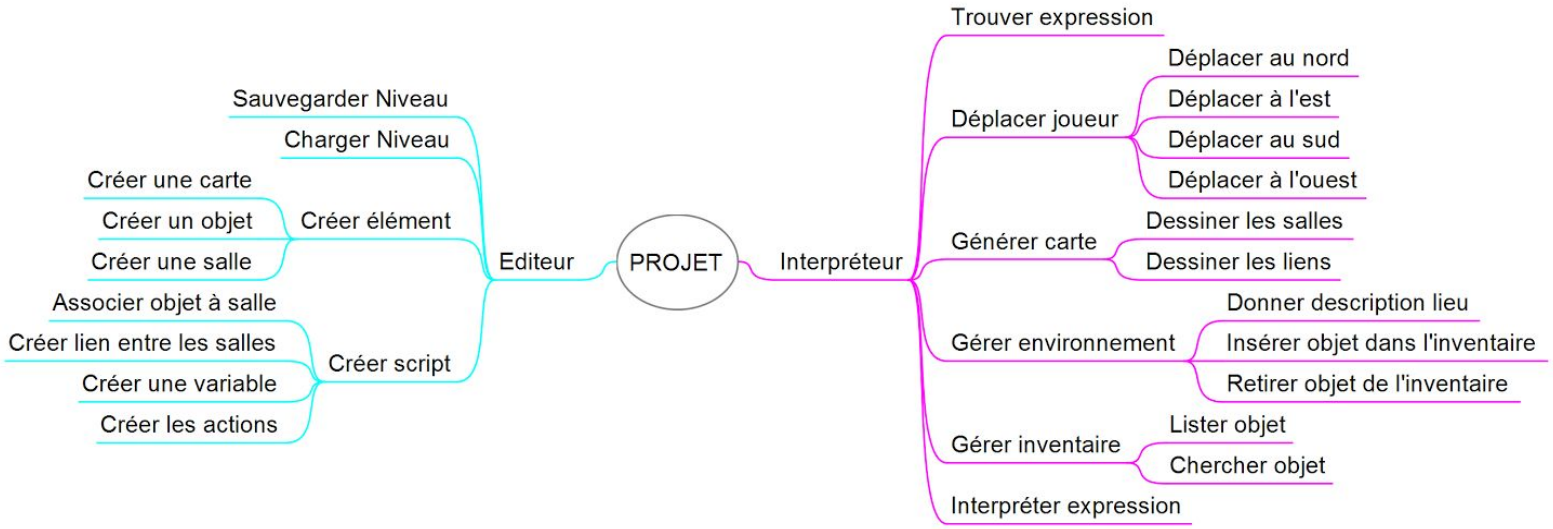
Lexique :

- **Éditeur de jeu** : L'éditeur de jeu permet de construire un niveau qui contient une carte. Cette carte contiendra des salles.
- **Jeu** : Fichier généré par l'éditeur qui contient l'ensemble des données. Le joueur devra charger ce fichier dans l'interpréteur pour pouvoir y jouer ou dans l'éditeur de jeu pour le modifier.
- **Script** : Texte structuré construit par l'éditeur de jeu permettant de créer un jeu. Le script contient toutes les données nécessaires pour faire fonctionner le jeu.
- **Action**: Partie du script comprenant les actions que l'interpréteur va effectuer en fonction de ce que le joueur fait.
- **Niveau** : Le niveau correspond au jeu créé par l'éditeur de niveau.
- **Carte** : Ensemble des salles et des liens entre salles d'un niveau. On affichera la structure du niveau au joueur.
- **Lien** : Jointure entre deux salles de la carte. Lorsque deux salles sont reliées par un lien, alors le déplacement du joueur par le jeu est possible, sinon aucun déplacement n'est effectué.
- **Salle** : Lieu constitué d'un nom et d'une image. La salle affichée sera l'emplacement courant du joueur. Le joueur parcourt la carte de salle en salle.
- **Objet** : Un objet est créé dans l'éditeur. Il peut être placé dans une salle, être récupéré par le joueur et être utilisé par le joueur.
- **Joueur** : Personne physique interagissant avec un niveau. Le joueur peut rentrer une phrase dans un formulaire présent dans l'interpréteur. Les verbes et les compléments de la phrase rentrée seront analysés selon les règles de la salle. Une conséquence sera exécutée si une des règles de la salle courante correspond à la phrase.

Ainsi, chaque jeu créé par l'éditeur est composé d'un niveau, lui même composé de salles et de liens. Une salle comprend une liste d'objets et des actions. Les actions doivent définir les conséquences appliquées par le jeu selon l'interpréteur. Les conditions de déclenchement d'une règle sont composées d'un ensemble de verbes, où chaque verbe est associé à un ensemble de complément.

2. Liste des fonctionnalités du système et présentation de l'éditeur

Le tableau ci dessous regroupe l'ensemble des fonctionnalités et leurs sous-fonctionnalités du projet ainsi que leur description brève. Ces fonctionnalités seront à faire pour que le projet soit opérationnel. Il est accompagné d'un schéma présentant toutes les fonctionnalités.

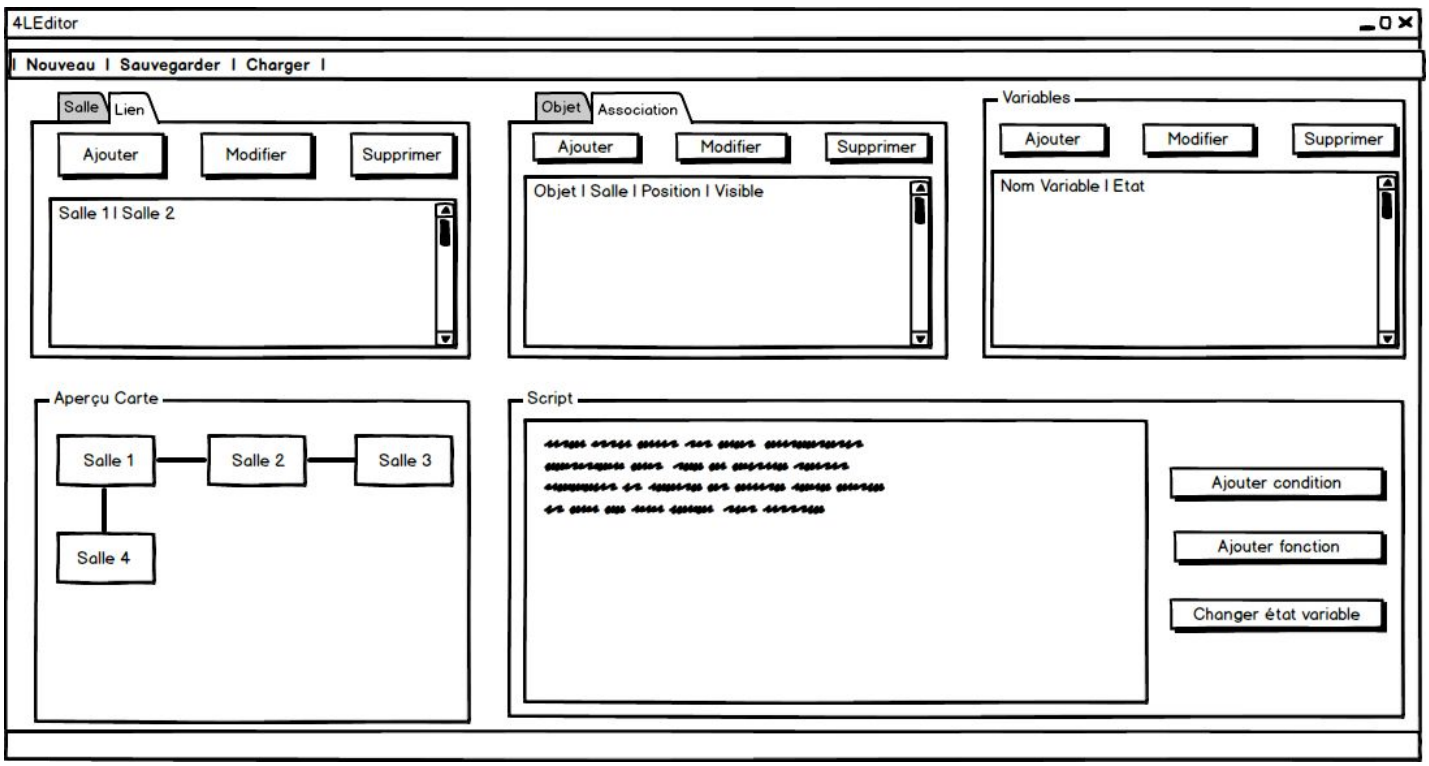


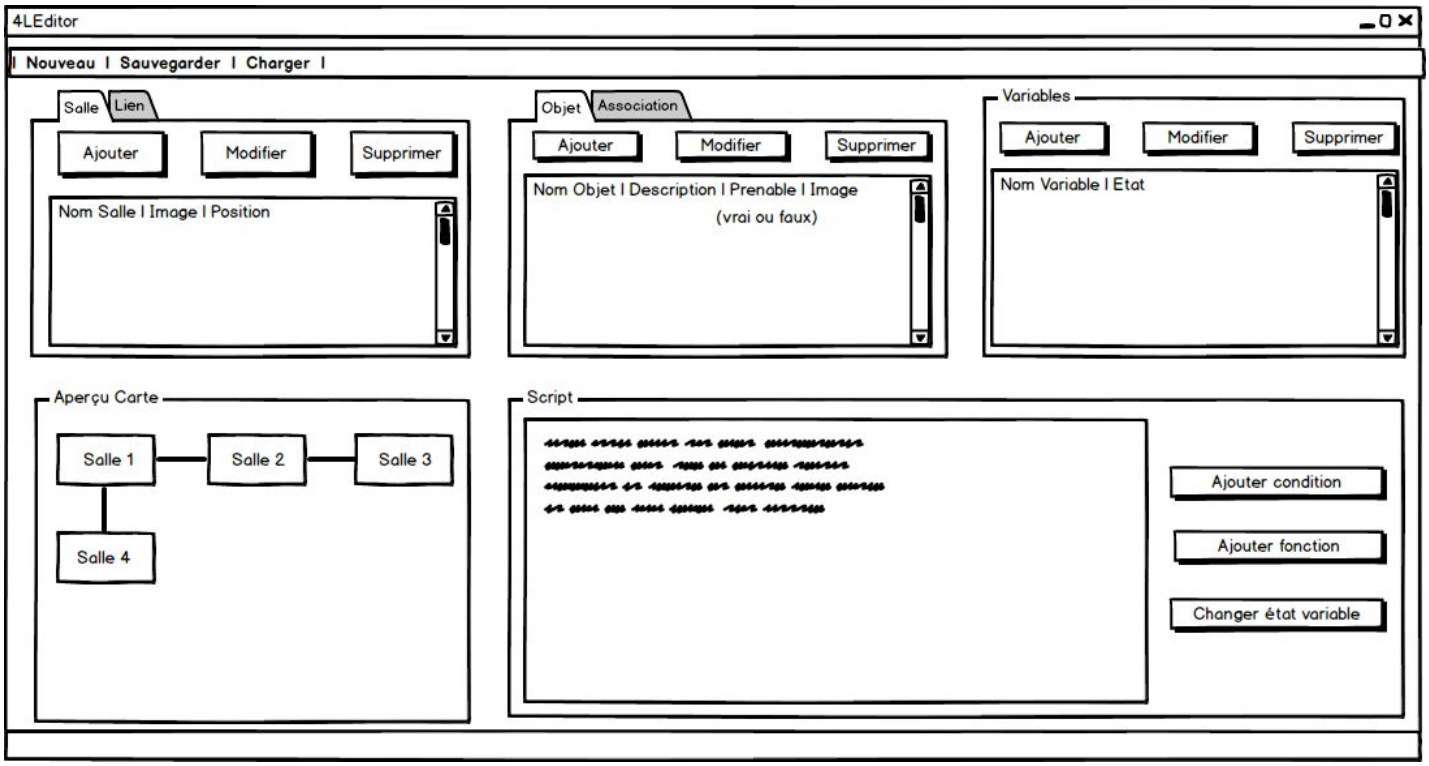
Fonctionnalité	Sous-Fonctionnalité	Description de la fonctionnalité
Interpréteur		
Trouver expression	Pas de fonctionnalité secondaire	L'interpréteur va lire la phrase écrite par le joueur, l'analyseur va ensuite découper la phrase et regarder la signification de chaque mot. L'analyseur va ensuite rechercher un verbe en premier, puis un complément et s'arrêtera.
Interpréter expression	Pas de fonctionnalité secondaire	Si des verbes et des compléments ont été trouvés alors l'interpréteur fera appel à l'action correspondante au couple verbe/complément
Déplacer joueur	Déplacer au nord	Le joueur se déplacera sur la carte du haut / bas / droite / gauche en fonction de la commande entrée. Il pourra se déplacer si une carte est
	Déplacer au sud	

	Déplacer à l'est	existante et si le passage est possible. L'interpréteur placera le joueur dans le lieu demandé.
	Déplacer à l'ouest	
Générer carte	Dessiner les salles	Les salles appartiennent à une carte, l'interpréteur de jeu affichera dans la carte les différentes salles. (cf : schéma de l'interpréteur)
	Dessiner les liens entres les salles	Les liens servent à relier les salles entre elles. L'interpréteur affichera dans la carte les liens entre les salles. L'interpréteur ne permettra au joueur de se déplacer d'une salle à une autre, que si elles sont reliées par un lien (cf : schéma de la structure général de l'interpréteur)
Gérer environnement	Donner description lieu	L'interpréteur donne le nom/description de la salle quand le joueur rentre dans celle-ci.
	Insérer objet dans l'inventaire	L'interpréteur placera l'objet dans l'inventaire du joueur quand le joueur aura indiqué de ramasser/prendre l'objet. L'objet disparaîtra également de la salle.
	Retirer objet de l'inventaire	L'interpréteur retirera l'objet de l'inventaire du joueur quand le joueur aura indiqué qu'il voudra l'utiliser et que l'action est possible.
Gérer inventaire	Lister les objets	L'interpréteur affichera graphiquement tous les objets que le joueur possède dans son inventaire.
	Chercher un objet	Si le joueur possède plusieurs objets, il pourra taper les premières lettres du nom de l'objet dans un champs pour le rechercher, l'interpréteur affichera les objets contenant les lettres.
Éditeur de niveau		
Sauvegarder niveau	Pas de fonctionnalité secondaire	L'éditeur de niveau pourra sauvegarder le niveau que le concepteur est en train de créer afin de pouvoir le modifier par la suite. L'extension des fichiers de sauvegarde de niveau sera .4l (Ex : save_niveau.4l)
Charger niveau	Pas de fonctionnalité secondaire	L'utilisateur pourra charger un fichier de sauvegarde (.4l) afin de pouvoir reprendre son travail. L'éditeur de niveau se chargera de lire le fichier et de remettre la carte, les composant et le script comme ils étaient.
Créer élément	Créer une carte	L'éditeur de niveau créera automatiquement une carte à la création d'un nouveau niveau.
	Créer un objet	L'éditeur de niveau utilisera un nom, une description et une image donnés par le concepteur pour créer un objet. L'objet sera traduit dans le script en texte

		(Exemple Feuille:Feuille de papier:true:images/objets/feuille01.png)
	Créer une salle	L'éditeur de niveau utilisera un nom, une image pour créer une salle, elle sera traduite dans le script en texte(Exemple : Feuille:Feuille de papier:true:images/objets/feuille01.png)
Créer script	Associer objet à salle	L'éditeur de niveau se chargera de créer une relation entre un objet et une salle a partir d'un objet, d'une salle, d'une position et si l'objet est visible ou non, cette relation sera traduite dans le script par du texte (Exemple : Feuille:Rue Sombre:210:110:true)
	Créer lien entre les salles	L'éditeur de niveau se chargera de créer un lien entre deux salles, cette relation sera traduite dans le script par du texte (Ex : Rue Sombre:Entrée de l'immeuble)
	Créer une variable	L'éditeur de niveau se chargera de créer une variable, cette variable sera traduite dans le script par du texte (Exemple : passageEntreeImmeubleVersSalonImmeuble = false)
	Créer les actions	L'éditeur de niveau se chargera de créer à partir de conditions, les actions réalisables par le joueur, cela sera traduit dans le script par du texte.

Les fonctionnalités présentent dans l'éditeur concernant les éléments et le script peuvent non seulement être créées mais également modifiées ou supprimées. Nous n'avons pas voulu préciser cela dans le tableau pour éviter de l'alourdir et de le rendre moins lisible.
Voici une maquette de l'éditeur de niveau comprenant toutes les actions qu'il sera possible de faire :





Chaque bouton “ajouter” ou “modifier” ouvrira une nouvelle fenêtre avec une interface intuitive pour créer son objet/salle etc. Le bouton “ajouter condition” permettra d’utiliser des variables, des objets, des opérateurs ou des “variables système” tels que “position” qui permet de connaître le nom de la salle où se situe le joueur. Des fonctions seront intégrées par défaut telles que :

- CACHER (Nom de l’objet, Salle où se situe l’objet) : Elle permettra de cacher un objet.
- MONTRER (Nom de l’objet, Salle où se situe l’objet) : Elle permettra de montrer un objet.
- MESSAGE (Texte) : Affiche un message au joueur.
- CONTIENT(MOTS) : Renvoie Vrai ou Faux si l’action de l’utilisateur contient les mots passés en paramètre.
- BREAK : Indique que l’on veut arrêter de regarder les actions suivantes.

Voici un prototype de script que notre éditeur va générer:

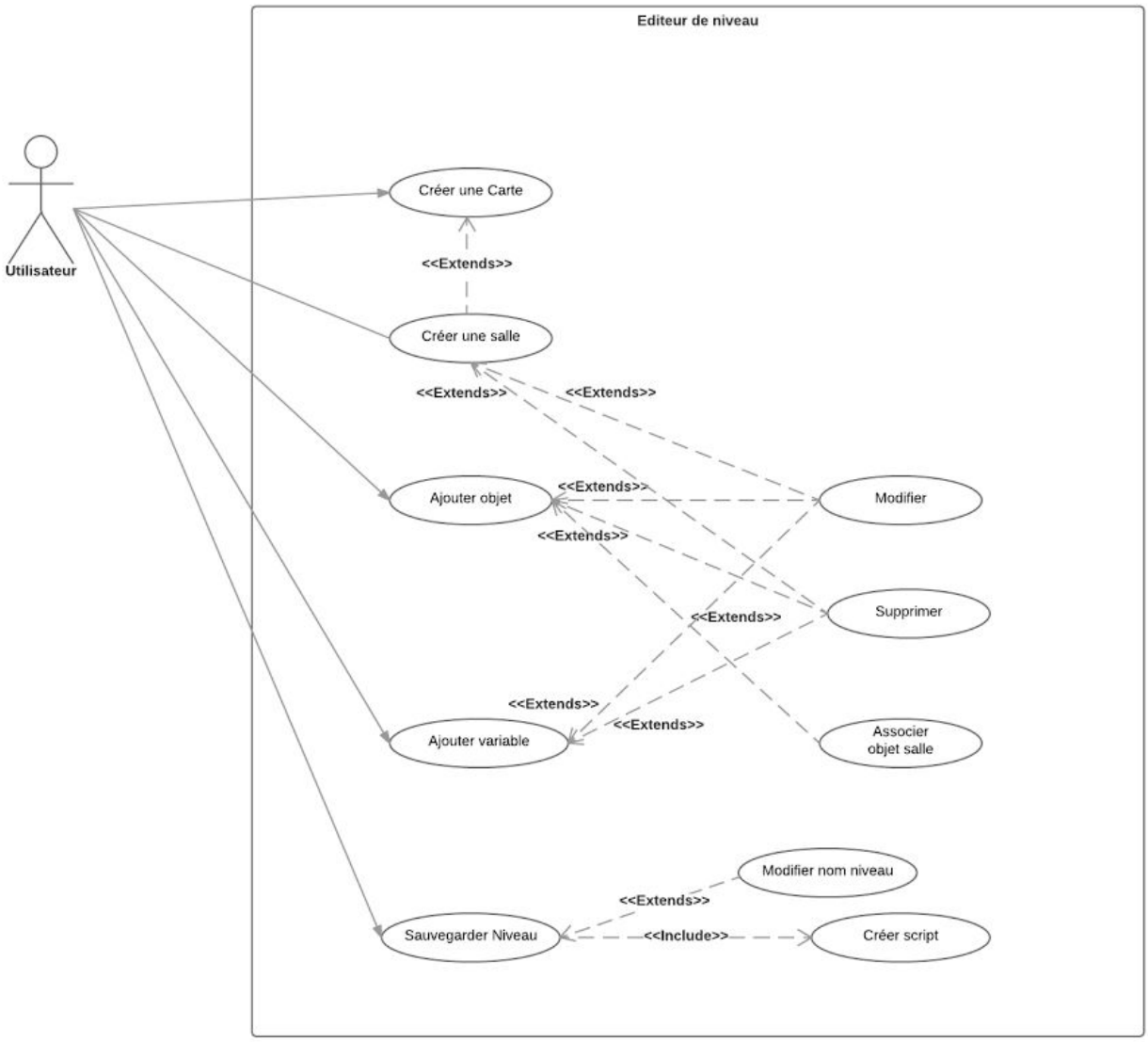
```
// // = Commentaire sur une ligne
// ## NOM ## = Catégorie

##VARIABLE##
passageEntreeImmeubleVersSalonImmeuble=false

##OBJET##
//Nom:Description:Prenable:Chemin de l'image
Feuille:Feuille de papier:true:images/objets/feuille01.png
Diamant:Diamant très rare:true:images/objets/diamant01.png
Porte Fermer:Cette porte est fermé:false:images/objets/porte_fermer01.png
```

```
Porte Ouverte:Cette porte est ouverte:false:images/objets/porte_ouverte01.Png
##SALLE##
//Nom:Chemin de l'image
Rue Sombre:images/salles/rue_sombre01.png
Entrée de l'immeuble:images/salles/entree_immeuble01.png
Salon de l'immeuble:images/salles/salon_immeuble01.png

##RELATION_OBJET##
//Nom Objet:Nom Salle:Positionx:Positiony:Visible
Feuille:Rue Sombre:210:110:true
Diamant:Entrée de l'immeuble:300:300:true
Porte Fermer:Entrée de l'immeuble:500:500:true
Porte Ouverte:Entrée de l'immeuble:500:500:false

##RELATION_SALLE##
//Nom salle 1:Nom salle 2
Rue Sombre:Entrée de l'immeuble
Entrée de l'immeuble:Salon de l'immeuble

##ACTION##
//position est une variable qui est la salle actuel du joueur

//la fonction CONTIENT retourne vrai si l'action contient les mots présent dans le tableau
//la fonction MESSAGE affiche un message
//la fonction CACHER cache un objet dans une salle donnée
//la fonction MONTRER montre un objet dans une salle donnée
//la fonction BREAK permet de sortir de toutes les conditions et de passer à l'étape suivante

SI (position == "Entrée de l'immeuble") ALORS{
  SI (CONTIENT("aller","nord")) ALORS{
        SI (passageEntreeImmeubleVersSalonImmeuble==false) ALORS{
              MESSAGE("La porte est fermer par un mécanisme bizarre...")
              BREAK
        }
  }
  SI (CONTIENT("placer","diamant","socle")) ALORS{
        CACHER("Porte Fermer","Entrée de l'immeuble")
        MONTRER("Porte Ouverte","Entrée de l'immeuble")
        passageEntreeImmeubleVersSalonImmeuble=true
        MESSAGE("La porte est ouverte !")
        BREAK
  }
}
```

3. Cas d’utilisation – scénarios - conditions de validation

Notre première itération étant sur l’éditeur et l’analyseur syntaxique (partie de l’interpréteur) nous nous pencherons plus particulièrement sur les fonctionnalités associées. L’éditeur de jeu doit comprendre des fonctionnalités, telles que la création d’objets, la création de salles, le chargement et l’édition de scripts existant.

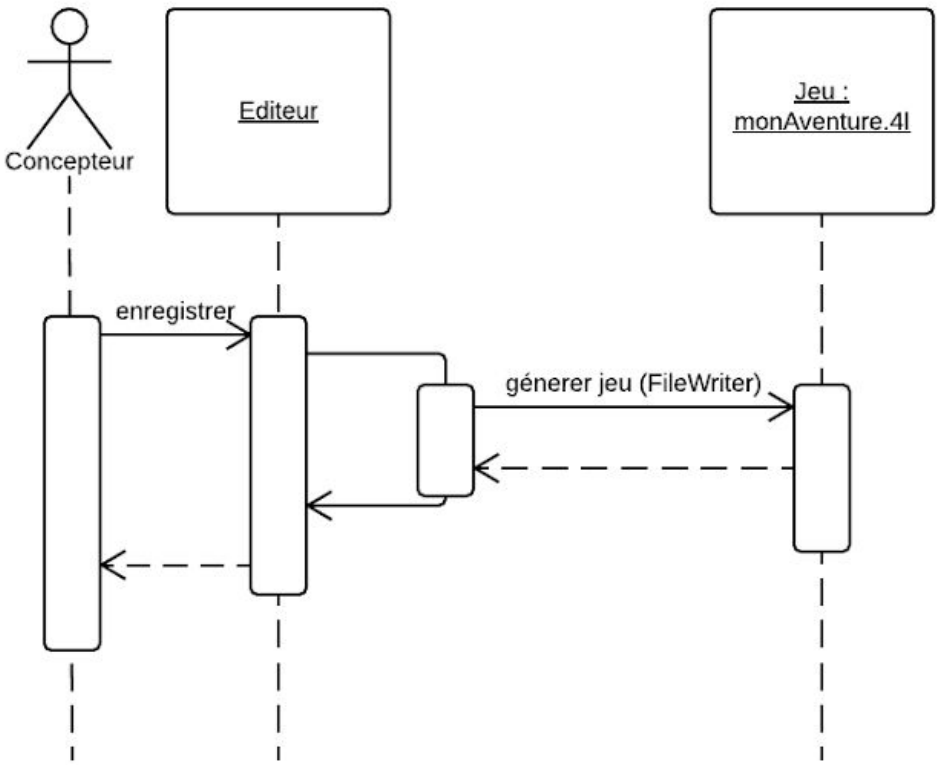
Le concepteur va pouvoir créer une carte, cela va créer automatiquement créer une salle et le concepteur va pouvoir également en créer contenu dans la carte. Les salles peuvent être modifiées ou supprimées. Le concepteur peut créer des objets, les modifier et les supprimer. Ils peuvent être reliés à une salle à la demande du concepteur. Le concepteur peut ajouter, modifier, supprimer des variables. Quand le concepteur le décide il peut sauvegarder son niveau (et en modifier le nom) , ce qui créera un script.



● Scénario 1 de l’éditeur - sauvegarder un jeu :

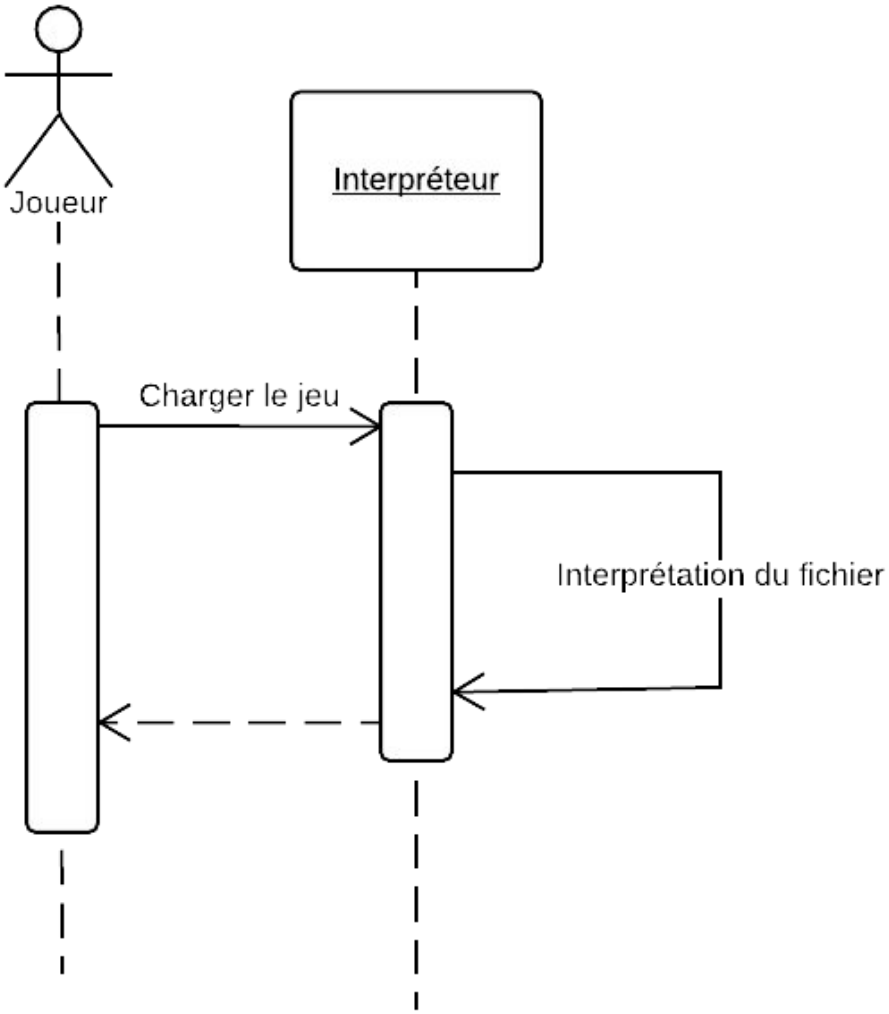
La sauvegarde d'un jeu depuis l'éditeur consiste en une écriture dans un fichier des données du jeu (cf. ébauche de script précédente).

Le concepteur va informer l'éditeur (via l'interface) qu'il veut enregistrer le script. L'éditeur va générer le fichier .4l, soit le jeu changeable par l'interpréteur (le fichier recensera les actions déclenchables, les objets associées aux salles, les caractéristiques du niveau [placement des salles et des liens]...).



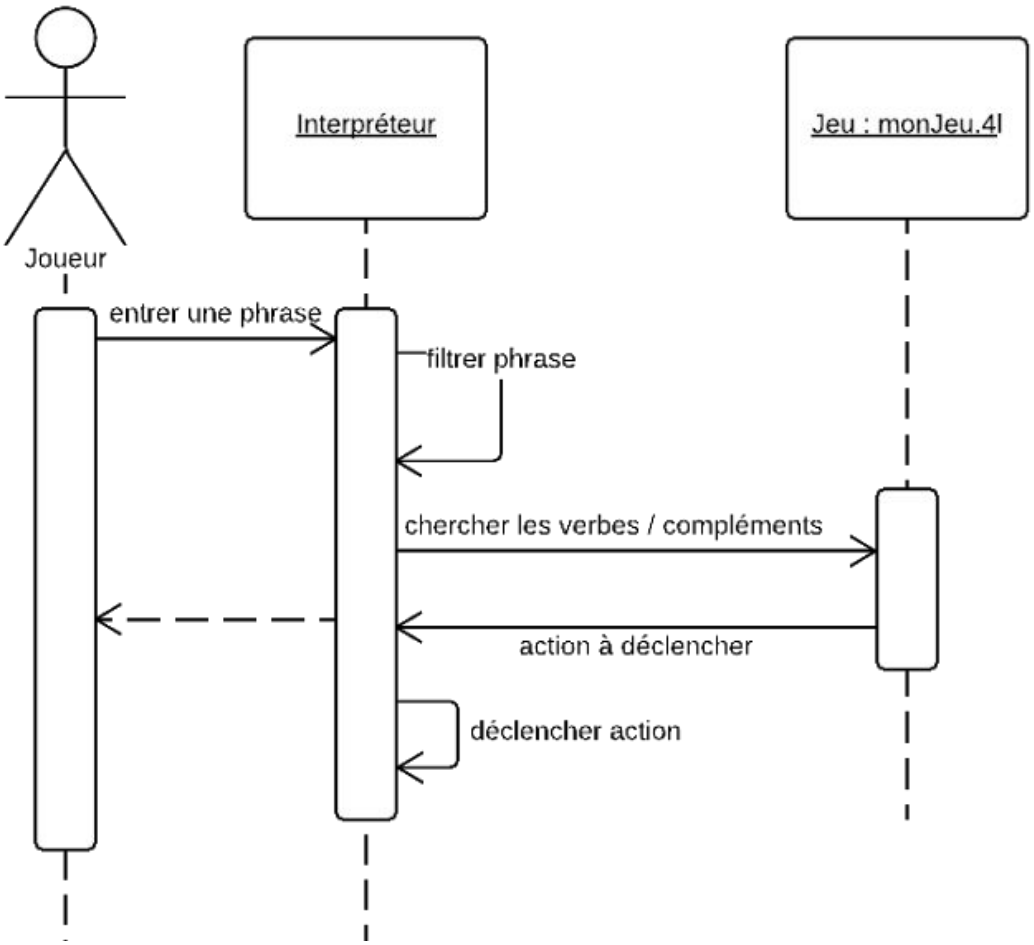
- Scénario 2 de l’interpréteur - charger jeu :

Le concepteur a sauvegardé son jeu et souhaite le tester dans l'interpréteur. L'interpréteur va récupérer le chemin du fichier jeu et le charger par une lecture de caractère. Lorsque la lecture du fichier est terminé, le fichier est interprété et le niveau est généré.



● Scénario 3 de l’interpréteur - déclencher les actions :

Le joueur va rentrer au clavier une chaîne de caractères dans le formulaire d'action du jeu. L'interpréteur va parcourir les actions reconnues dans la salle où se trouve le joueur. Le parcourt correspond à une lecture du fichier jeu.4l écrit dans l'éditeur. Si la chaîne rentrée contient un couple verbe/complément(s) existant alors l'interpréteur déclenche l’action correspondante. (variante : si la chaîne n’est pas reconnu un message d’erreur sera affiché)



5. Recensement des risques et répartition des rôles

a) Risques

Le projet peut comporter plusieurs types de risques comme l'impossibilité de réaliser certaines fonctionnalités trop complexes à mettre en place par rapport à la structure de l'application ou qui prennent plus de temps que prévu à l'implémentation. Un autre risque peut être la casse du moteur graphique par une action non prévue créée dans le script ou due à la mauvaise lecture du script (altéré par une action externe au programme, ou interne au programme en cas de mauvaise gestion).

Il y a également un risque de mauvaise compréhension de l'utilisateur vis a vis de l'éditeur. Car même s'il est développé en visant le côté intuitif et la rapidité de la prise en main, il est possible que la création de règles soit un concept flou et difficile à s'approprier pour l'utilisateur. De plus pour le moment, le fichier généré par l'éditeur (cf ébauche de script) n'est pas accessible au plus grand nombre, une connaissance technique de l'interprétation du script est requise.

On citera également la possibilité d'une limite de fonctionnalités implémentables. En effet si notre structure n'est pas suffisamment générale, alors l'application ne pourra pas recevoir davantage de fonctionnalités sans une modification profonde et chronophage du code.

b) Répartition des rôles

Pour cette première itération, nous allons plus particulièrement nous intéresser a l'éditeur de jeu ainsi qu'à l'analyseur syntaxique. De ce fait, nous allons nous diviser en deux sous groupes.

Le premier groupe sera constitué de Louis ZWAWIAK et Lucas GERARD, tout deux travaillerons sur l'éditeur de niveau. Le but est que celui-ci soit terminé à la première itération afin de pouvoir commencer l'interpréteur dans une seconde itération. Il sera donc possible de créer un fichier qui pourra être lu et interprété par le futur interpréteur avec toutes les fonctionnalités telles que la création d'objets, de salles, de variables, d'associations entre un objet et une salle ainsi que la création de liens entre les salles.

Le second groupe sera constitué de Loïc LEPRIEUR et Laura TRIVINO et s'occupera de l'analyseur syntaxique (on considérera cet analyseur comme le prototype de l'interpréteur afin de valider les choix d'implémentation de l'éditeur et tester les fichiers générés par l'éditeur). Le travail à effectuer sera le suivant : récupération et traitement de

la phrase envoyée par le joueur. A partir de cette phrase, ils utiliseront le dictionnaire en JSON pour trouver les actions de bases.