

02/04/2017

IA : Projet Prolog

« Les Chroniques de Caradoc » :
Création d'un jeu vidéo textuel –
ENSC 2A



PROLOG

Grégoire MELIN – Josias LEVI ALVARES

BORDEAUX INP- ECOLE NATIONALE SUPERIEURE DE COGNITIQUE

TABLE DES MATIERES

Introduction	2
Positionnement & objectifs	2
Equipe projet	2
Monde.....	2
Carte.....	2
Personnages et Créatures	2
Spécifications	4
Exigences.....	4
Analyse des besoins utilisateurs	5
Modélisation & Réalisation du projet	5
Architecture du jeu	5
Description prédicats	6
Crafting	6
Combat.....	7
Commerce.....	9
Interface Utilisateur	10
Illustration du système par un cas d'utilisation	10
Démarrer le jeu	10
Test de Validation	10
Conclusion.....	12
Difficultés rencontrées.....	12
Git.....	12
Prolog.....	13
Point d'amélioration	13
Système de combat.....	13
Developper le prédicat construire	13
Permettre l'achat d'objet auprès du marchand.....	14
Apports du projet.....	14

INTRODUCTION

POSITIONNEMENT & OBJECTIFS

L'objectif du projet est la création d'un jeu vidéo textuel dans le langage PROLOG dans le cadre du projet Prolog de 2^{ème} année de l'ENSC.

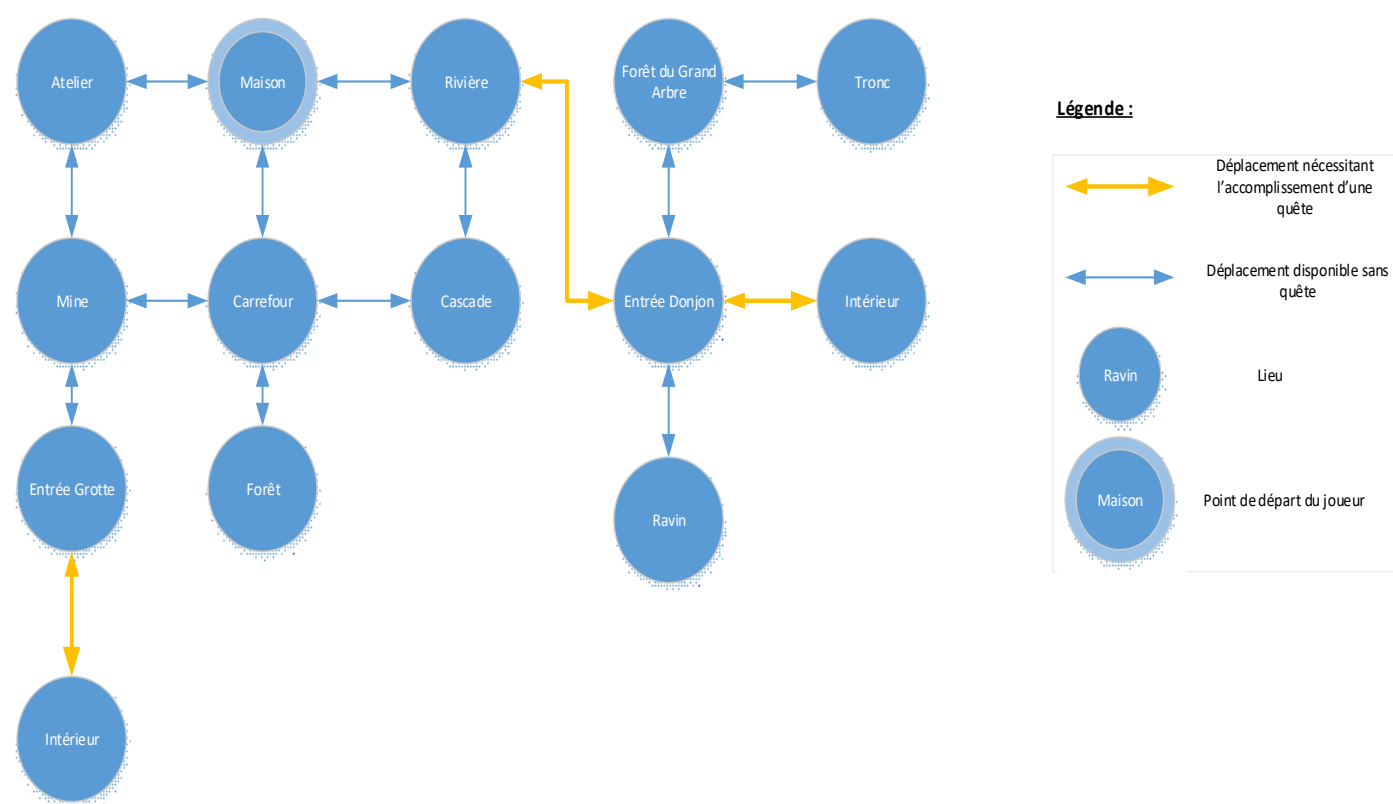
EQUIPE PROJET

L'équipe projet est composée de Grégoire MELIN et Josias LEVI-ALVARES tout deux élèves de deuxième année à Bordeaux INP – Ecole Nationale Supérieure de Cognitique.

MONDE

On s'attachera dans cette partie à décrire quelques aspects du monde dans lequel évolue le joueur. Pour résumer, le jeu peut s'apparenter à un action-RPG dans un univers médiéval-fantastique.

CARTE



PERSONNAGES ET CREATURES

ANIMAUX

Au cours d'une partie, le joueur rencontrera animaux qu'il pourra chasser. Le principal but de chasser des animaux est de récupérer du cuir permettant de crafter des objets ou de les vendre au marchand contre de l'argent. Chaque animal est défini par un prédicat d'arité 4. Le premier attribut correspond au nom de l'animal, le deuxième attribut au nombre de point d'attaque, le troisième au nombre de points de défense et le dernier au nombre de « cuir » récupéré par le joueur quand la bête est tuée. Différents bestiaux sont disponibles dans le jeu. La liste des monstres présents est disponible dans le paragraphe suivant.

En termes de prédicats, on définit un animal tel que :

animal(nom,nb_point_attaque,nb_point_defense,nb_cuir_recuperable).

LE COCATRIX

Description : « La cocatrix, coquatrix, caucatrix ou cocatrice est un animal fabuleux et imaginaire, qui possède selon la légende une tête de coq, des ailes de chauve-souris et un corps de serpent ou de coq »

<https://fr.wikipedia.org/wiki/Cocatrix>

Nom : cocatrix

Point d'attaque : 5

Point de défense : 5

Nombre de « cuir » récupéré : 1

LE NEKKER

Description : « Ces créatures primitives sont le fléau des zones sauvages. Les habitants de la forêt les craignent et les animaux évitent soigneusement leurs nids. Les nekkers, créatures sociales, se réunissent dans ce qui ressemble à des tribus, car ils ne peuvent se défendre qu'en masse. » <http://sorceleur.gamepedia.com/Nekker>

Nom : nekker

Point d'attaque : 15

Point de défense : 10

Nombre de « cuir » récupéré : 2

LE CENTAURE

Description : « Dans la [mythologie grecque](#), un **centaure** (en [grec ancien](#) Κένταυροι / *Kéntauroi*) est une créature mi-homme, mi-cheval, que l'on disait issue soit d'[Ixion](#) et de [Néphélé](#), soit de [Centauros](#) et des juments de [Magnésie](#). » <https://fr.wikipedia.org/wiki/Centaure>

Nom : centaure

Point d'attaque : 20

Point de défense : 30

Nombre de « cuir » récupéré : 3

LE SANGLIER

Description : « Le **sanglier** (*Sus scrofa*) est un [mammifère omnivore](#), [forestier](#), proche du [porc](#). Le [genre](#) *Sus* appartient à la [famille](#) des [suidés](#), dans l'[ordre](#) des [Artiodactyla](#) ou des [cétartiodactyles](#) selon les classifications. » <https://fr.wikipedia.org/wiki/Sanglier>

Nom : sanglier

Point d'attaque : 30

Point de défense : 50

Nombre de « cuir » récupéré : 10

LE TROLL

Description : « Un **troll** est un être de la [mythologie nordique](#), incarnant les forces naturelles ou la magie, caractérisé principalement par son opposition aux hommes et aux dieux. Ce troll est souvent assimilé aux [Jötunns](#), les « Géants » de cette mythologie. » <https://fr.wikipedia.org/wiki/Troll>

Nom : troll

Point d'attaque : 60

Point de défense : 60

Nombre de « cuir » récupéré : 20

PNJ

Les PNJ sont définis par leur nom. Dans le jeu est disponible un marchand disponible. Le mage est un PNJ de quête et le marchand permet d'introduire un commerce dans le jeu. Remarquons que ces personnages n'ont pas d'existence physique réelle dans le jeu étant donné qu'ils sont définis par leurs interactions avec le joueur.

La syntaxe Prolog est donc la suivante :

pnj(mage).

pnj(marchand).

BOSS

Les boss sont définis de manière analogue aux animaux à la différence que leur mort ne permet pas de récupérer du cuir. Les boss permettent de faire avancer le jeu. La mort de l'orc permet de débloquent la porte de la grotte où se trouve l'un des objectifs et le mage permet d'ouvrir une porte. Ils agissent en quelques sortes comme des clés. Il est cependant possible d'interagir avec eux comme avec le reste des PNJ en les attaquant par exemple.

Les deux boss existants sont :

UN ORC

Description : «Les orcs, ou orques, humanoïdes à l'aspect repoussant et aux mœurs barbares, forment un peuple imaginaire appartenant au sous-genre médiéval-fantastique de la fantasy.»

[https://fr.wikipedia.org/wiki/Orque_\(cr%C3%A9ature\)](https://fr.wikipedia.org/wiki/Orque_(cr%C3%A9ature))

Nom : orc

Point d'attaque : 100

Point de défense : 50

UN DRAGON

Description : « Le dragon est une créature légendaire représentée comme une sorte de gigantesque reptile, ailes déployées et pattes armées de griffes. Dans de nombreuses mythologies à travers le monde, on retrouve des créatures reptiliennes possédant des caractéristiques plus ou moins similaires, désignées comme dragons. »

<https://fr.wikipedia.org/wiki/Dragon>

Nom : dragon

Point d'attaque : 300

Point de défense : 100

LE JOUEUR

Le joueur est un PNJ particulier capable de bouger de lieu, de s'armer, d'attaquer. Il est cependant défini en termes de combat par exemple comme les autres entités présentes dans le monde du jeu.

SPECIFICATIONS

Cette partie traitera des exigences choisies lors de la conception du jeu.

Légende :

EIHM_X : Exigences issues des besoins utilisateurs

E_X : Exigences

EXIGENCES

E_1 : Le jeu devra comporter au moins un objet ou une porte fermée.

E_11 : Le jeu comportera un ou des coffres devant s'ouvrir à l'aide d'une clé.

E_12 : Le jeu comportera une porte fermée nécessitant l'accomplissement d'une quête pour l'ouvrir.

E_2 : Le jeu devra comporter au moins un endroit ou un objet caché

E_21 : Le jeu comportera un lieu caché non affiché sur la carte que le joueur devra découvrir au cours de sa quête.

E_3 : Le jeu devra comporter des objets incomplets : Une attention particulière devra être portée à l'action de crafter des objets.

E_31 : Un système de récolte ressources sera mis en place.

E_32 : Le joueur aura la possibilité de combiner des objets et des ressources pour obtenir des objets.

E_4 : Le jeu devra comporter des ressources limitées

E_41 : Les ressources au sein du jeu seront disponibles seulement dans des endroits précis.

E_42 : Les limites en termes de ressources seront modélisées par les performances des outils du joueur (pioche de base, pioche en métal...)

ANALYSE DES BESOINS UTILISATEURS

EIHM_1 : L'affichage devra être clair et standardisé quel que soit le lieu dans lequel le joueur évolue.

EIHM_2 : En plus du mode d'emploi, le jeu devra comporter des indications progressivement pour guider le joueur.

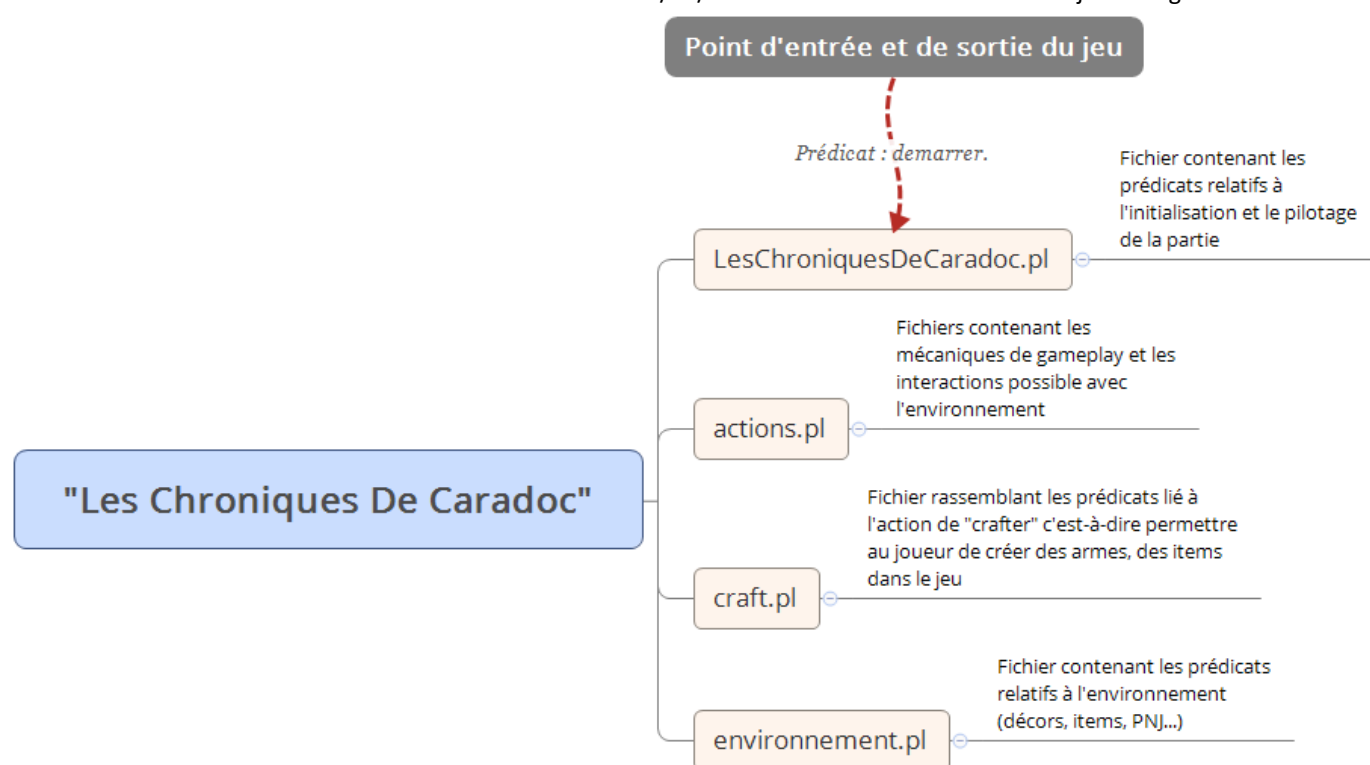
EIHM_3 : Le jeu devra comporter des raccourcis permettant au joueur de ne pas avoir à taper des prédicats trop longs.

EIHM_4 : Le joueur pourra en cas de problème consulter le tutoriel.

MODELISATION & REALISATION DU PROJET

ARCHITECTURE DU JEU

Pour des soucis de clarté du code, il a été décidé de fractionner le code en plusieurs programmes. L'ensemble est piloté par le programme *LesChroniquesDeCaradoc.pl* qui consulte automatiquement les autres fichiers via le prédicat *consult/1*.



DESCRIPTION PREDICATS

Dans cette partie nous nous attacherons à expliquer le fonctionnement de certains prédicats. Il ne s'agit pas de faire une liste exhaustive de tous les prédicats définis pour le jeu mais bien de détailler la démarche sous-jacente à certains prédicats et ainsi d'essayer de justifier leur existence.

CRAFTING

CRAFTING SIMPLE : DES RESSOURCES DONNENT UN OBJET

Le crafting est l'action permettant par la combinaison d'objet de l'inventaire de créer d'autres objets au sein du jeu. Les objets à crafter sont soit des armes, soit des outils, soit des objets de quête. Il nécessite la collecte de ressources et sont définis de la façon suivante, par deux natures de prédicats. Le premier est le prédicat *crafter/4* qui permet au joueur de crafter une première fois les objets. En effet, l'un des principes essentiel du crafting du programme est que le joueur doit essayer différentes combinaisons pour crafter un objet jusqu'à trouver la bonne combinaison typiquement le fait de crafter un arc qui nécessite la combinaison de deux lianes et d'un bois. Ceci donne au final :

```

crafter(arc) :-
    joueur_est(atelier),
    connait(arc),
    crafter(liane, rien, bois, liane).
  
```

Avec la dernière ligne correspondant à :

```

crafter(bois, liane, liane, rien) :-
    crafter(liane, rien, bois, liane), nl, !.
crafter(liane, rien, bois, liane) :-
    verification(liane, 2, montrer_erreur),
    verification(bois, 1, montrer_erreur),
    ajouter(arc, 1),
    assert(connait(arc)),
  
```

```
retirer(liane, 2),
retirer(bois, 1),
ecrire_gras_couleur('Vous avez crafte un ', green), ecrire_gras_couleur('arc', blue),
ecrire_gras_couleur(', pas la peine d'en creer un autre !', green), nl, !.
```

Une fois que le joueur a réussi à crafter une première fois un objet, il peut utiliser un raccourci à savoir le prédicat *crafter/1* qui lui permet de crafter directement l'objet sans avoir à retaper tous les composants nécessaires à la main :

```
crafter(hache, Minerai) :-
    joueur_est(atelier),
    connait(hache),
    crafter(Minerai, Minerai, bois, rien).
```

CRAFTING AUTRE : ENCHANTER

Cette sorte de crafting est permise par l'introduction du prédicat *enchanter/2* qui permet de fusionner deux objets pour en donner un seul. Il est toutefois uniquement utilisé pour la création de « L'épée du Héro » permettant d'accomplir la quête finale.

```
enchanter(epee_du_hero) :-
    sac(epee_du_hero, 1),
    sac(parchemin, N),
    N > 0,
    assert(sac(epee_enchantee, 1)),
    retract(sac(epee_du_hero, _)),
    write('Alors que vous marmonez les mots inscrit sur un parchemin, une flamme bleue jai
llit de l''epee du heros. '), nl,
    nl,
    ecrire_gras_couleur('Vous avez obtenu l'', green), ecrire_gras_couleur('epee enchante
e', blue), ecrire_gras_couleur(' !', green), nl,
    nl,
    write('Sur sa lame est ecrit : '), nl,
    ecrire_gras_couleur('LE CAUCHEMAR DES DRAGONS', red), nl, !.
enchanter(_) :-
    ecrire_gras_couleur('Rien ne se passe.', blue), nl, !.
```

LISTE DES COMBINAISONS POUR LES ARMES

Ci-dessous est disponible la liste des craft possible en termes d'armes et d'outils. Fondamentalement il est nécessaire d'avoir 4 éléments pour en forger un :

Hache	Pioche	Casque	Plastron	Arc	Flèche
Épée	Pioche	Bottes	Pantalon	Arc	Flèche
	Bois		Minerai ou cuir		Liane
	Minerai				

COMBAT

ATTAQUE DE PERSONNAGES POUR ACCOMPLIR DES QUETES

Le système de combat est relativement simple. Chaque entité capable de combattre est caractérisée par des coefficients d'attaque et de défense relative pour joueur par exemple à son équipement. A partir de ce coefficient sont calculés des statistiques de combat grâce au prédicat :

```
statistiquesCombat(Boss, PtsBoss, PtsJoueur):-
    boss(Boss, AB, DB),
    PtsBoss is AB + DB,
    pts_attaque(A),
    pts_defense(D),
    PtsJoueur is A + D.
```

Ainsi lors de l'attaque d'un Boss (entité combattante du jeu qui sera prise pour exemple), le moteur du jeu compare les capacités du joueur aux capacités du boss. Actuellement, les coefficients d'attaque et de défense sont sommés donnant ainsi un seul et unique coefficient par participant. Les deux coefficients, celui du joueur et celui du boss sont comparés. Deux cas sont alors possible comme le montre le prédicat attaquer/1 ci-dessous :

1^{er} Cas :

Dans ce cas-là, le joueur possède un talent permettant une issue favorable au combat. Son coefficient de combat est plus élevé que l'ennemi qui lui fait face.

2^{ème} Cas :

```
attaquer(Boss):-
    vivant(Boss),
    boss(Boss, _, _),
    joueur_est(Ici),
    il_y_a(Boss, Ici),
    statistiquesCombat(Boss, PtsEnnemi, PtsJoueur),
    PtsJoueur >= PtsEnnemi,
    ecrire_gras_couleur('Vous decider d''attaquer ', blue), ecrire_gras_couleur(Boss, red)
, nl,
    ecrire_gras_couleur('Il est vaincu. Gloire au hero !', green), nl,
    retract(vivant(Boss)),
    retract(il_y_a(Boss, _)), !.
attaquer(Boss):-
    vivant(Boss),
    boss(Boss, _, _),
    joueur_est(Ici),
    il_y_a(Boss, Ici),
    statistiquesCombat(Boss, PtsEnnemi, PtsJoueur),
    PtsJoueur < PtsEnnemi,
    ecrire_gras_couleur('Vous decider d''attaquer ', blue), ecrire_gras_couleur(Boss, red)
, nl,
    ecrire_gras_couleur('Il vous a vaincu. ', red), nl,
    mourir, !.
```

Le deuxième cas est complémentaire au premier dans le sens où dans ce cas, le joueur meurt au combat car la chose qui lui fait face à un coefficient de combat plus élevé.

CHASSE POUR RECOLTER DU CUIR

Il est possible pour le joueur de chasser des animaux dans le jeu afin de pouvoir récolter des ressources qu'il pourra vendre au marchand ou encore utiliser pour crafter des objets. Les prédicats relatifs à la chasse sont définis de manière analogue aux prédicats liés au combat et ont pour seule différence l'acquisition de cuir pour le joueur en cas d'issue favorable au combat.

COMMERCE

Il est nécessaire dans certaines constructions d'objets, notamment celle du bateau de posséder de l'argent. C'est la raison d'exister du marchand au carrefour. En effet, il est possible de vendre au marchand des objets, via le prédicat *vendre/2*.

Le marchand se situant au carrefour, il est nécessaire de se trouver au carrefour, d'où le deuxième cas afin d'empêcher la vente dans un autre endroit que le carrefour. La condition suivante est de posséder les objets que l'on veut vendre d'où la ligne suivante. Suite à cela, la transaction a lieu et le joueur se voit retirer le/les objet(s) vendu(s) en échange d'argent. S'en suit un retour visuel pour le joueur.

```
vendre(Objet, Quantitee) :-
    joueur_est(carrefour),
    verification(Objet, Quantitee, montrer_erreur),
    equivalent(Objet, Artefact, Matériau),
    objet(Artefact, Matériau, _, _, _, Prix),
    Prix \= 0,
    retirer(Objet, Quantitee),
    Gain is Quantitee * Prix,
    ajouter(or, Gain),
```

```

    ecrire_gras_couleur('Vous avez vendu ', green), ecrire_gras_couleur(Quantitee, blue),
write(' '), nom_objet(Objet), ecrire_gras_couleur('(s)', blue),
    ecrire_gras_couleur(' pour ', green), ecrire_gras_couleur(Gain, blue), ecrire_gras_couleur(' pieces d\'or', blue), ecrire_gras_couleur('.', green),
    !.
vendre(_, _) :-
    ecrire_gras_couleur('Action impossible !', red), nl,
    (joueur_est(carrefour), nl ; ecrire_gras_couleur('Vous n\'etes pas au carrefour.', red), nl),
    !.

```

INTERFACE UTILISATEUR

Prolog étant limité en termes d’affichage, l’interface utilisateur est assez simple et composé de texte. Les descriptions des évènements et des lieux sont standardisés afin que le joueur ne se perde pas.

```
?- decrire(ravin).
```

RAVIN

Vous pouvez désormais utiliser le voyage rapide pour ce lieu grace a la commande aller(ravin)

Vous : Ou bien je retourne au donjon...

DIRECTIONS POSSIBLES

Aller au donjon : Nord (n.)

ASTUCES

Vous : Ca a l'air haut ici... j'entends un bruit venant du bas, je pourrais aller y jeter un coup d'oeil.
- sepencher pour se pencher.

```
true.
```

Une première partie permet de décrire le lieu dans lequel on est grâce à son label et à la description qu’il en est faite. Suite à cela, une partie relative aux directions que le joueur peut prendre après avoir visité le lieu. Une troisième partie permet d’afficher les astuces de gameplay ou des indications permettant de diriger le joueur vers ses objectifs de quête.

ILLUSTRATION DU SYSTEME PAR UN CAS D’UTILISATION

Dans cette partie sera reportée un des tests de validation fait au cours du projet.

DEMARRER LE JEU

Pour démarrer :

```
?- consult('CHEMIN_DU_DOSSIER//projet-prolog-gm_jla//LesChroniquesDeCaradoc.pl').■
```

```
?- demarrer.■
```

Le jeu se lance alors.

TEST DE VALIDATION

Dans cette partie, on reproduira un des tests de validation fait au cours du développement. Ici, le but du joueur est d’aller récolter un minerai est d’aller attaquer l’orc de la grotte à mains nues.

On démarre donc le jeu :

```
?- demarrer.
```

TUTORIEL

Entrez les commandes avec la syntaxe Prolog standard.

Les commandes disponibles sont :

demarrer.	-- pour commencer une partie.
regarder.	-- pour regarder de nouveau autour de vous.
n: s, e, o, u, d.	-- pour aller dans une direction.
aller(Lieu).	-- pour aller dans un lieu déjà découvert.
sequiper.	-- pour sequiper des meilleurs équipements que vous avez.
miner(Minerai).	-- pour miner un minerai (pierre, cuivre, diamant).
recueillir(Ressource).	-- pour recueillir une ressource (bois, liane).
crafter(HautGauche,HD,BG,BD).	-- pour crafter un objet.
crafter(Objet).	-- pour crafter un objet que vous avez déjà crafté.
construire(Objet).	-- pour construire un objet spécifique.
attaquer.	-- pour attaquer un ennemi.
manger(Objet).	-- pour manger un objet.
mode_emploi.	-- pour afficher le mode d'emploi de nouveau.
terminer.	-- pour terminer la partie.
mort.	-- pour se suicider

Vous, chevalier légendaire, vous reveillez avec la pale lueur du soleil traversant la fenêtre de votre chambre.

Une sensation glacante mais familière vous envahit alors : Toujours le même cauchemar.

Un monde détruit, ou chaos et monstres ont pris le dessus sur la nature et la paix.

Dans votre cauchemar apparaît ce qui semble être un sombre cachot dans lequel une forte lumière émane d'un coffre vous éblouit.

Ce rêve, tinte de réalité, vous porte à croire que trouver ce coffre règlera vos problèmes. Vous vous mettez donc en quête de ce dernier.

Préparez vous à rentrer dans Xaaxaroth!

MAISON

Vous arrivez chez vous.

Vous : Ma bonne vieille maison, mais bon je me reposerai plus tard !

Vous pouvez désormais utiliser le voyage rapide pour ce lieu grâce à la commande aller(maison)

Vous : Ou vais-je continuer mon chemin ?

DIRECTIONS POSSIBLES

Aller au carrefour : Sud (s.)
 Aller à la rivière : Est (e.)
 Aller à votre atelier : Ouest (o.)

true.

Dirigeons nous alors vers l'ouest à l'atelier et ramassons une pioche, indispensable pour récolter un minerai.

Choisissons alors de récolter des pierres :

```
?- o.
```

ATELIER

Un atelier se dresse devant vous, c'est le votre.

Vous : Mon atelier ... avant ces cauchemars j'y passais mes journées.

Vous : On dirait même que mes premiers outils sont toujours fonctionnels, cela pourrait m'être utile.

Vous pouvez désormais utiliser le voyage rapide pour ce lieu grâce à la commande aller(atelier)

Vous : Je pourrais crafter des objets.

Astuce : - crafter(HautGauche,HD,BG,BD) => remplacez chaque emplacement HG/HD/BG/BD par pierre/cuivre/diamant/cuir/bois/liane/rien
 - crafter(Objet) si vous en avez déjà fait Objet (peut importe le minerai).
 - ramasser(Objet) (ex: ramasser epee de base => ramasser(epee_base)).

Vous : Ou bien je continue mon chemin...

DIRECTIONS POSSIBLES

Aller à la mine : Sud (s.)
 Aller à votre maison : Est (e.)

Il y a un(e) **pioche de base** ici.
 Il y a un(e) **hache de base** ici.
 Il y a un(e) **marteau de base** ici.

true.

```
?- ramasser(pioche_base).
```

Vous ramassez pioche de base.

true.

Suite à cela, équipons nous de la pioche et collectons de la pierre. Observons l'inventaire afin de vérifier que nous avons bien récolté la pierre en question et partons vers la grotte attaquer l'orc qui s'y trouve.

```
?- s.
=====

MINE

Vous vous dirigez vers la mine. Une lueur etrange eclaire l'endroit. Vous vous engouffrez alors dans le tunnel le plus proche.
Vous arrivez face a un gisement.

Vous pouvez desormais utiliser le voyage rapide pour ce lieu grace a la commande aller(mine)

Vous : On dirait que ces gisements sont encore exploitables. Ou est ma pioche ?
Astuce : - il y a une pioche de base a l'atelier.
Astuce : - equiper(pioche) pour equiper votre meilleure pioche.
          - miner(Minerai) => remplacez Minerai par pierre/cuivre/diamant
          - suivant la rarete de votre pioche, vous minerez plus facilement les mineraux.

Vous : Ou bien je continue mon chemin...

DIRECTIONS POSSIBLES

Aller a votre atelier : Nord (n.)
Aller a la grotte    : Sud (s.)
Aller au carrefour   : Est (e.)

true.

?- equiper(pioche).
Vous avez equipe le/la/les meilleur(e)(s) pioche : pioche de base
true.

?- miner(pierre).
Vous avez mine 1 pierre(s).

true.

?- i.
Appuyer sur la touche ";" pour faire defiler l'inventaire
INVENTAIRE
Item : pierre
Quantitee : 1

true ;

Rien.

?- s.
=====
```

ENTREE DE LA GROTTE

Vous arrivez face a la grotte. Devant l'entree un orc fume tranquillement de l'herbe a pipe.
 Outre son odeur, vous remarquez que la bete est bigrement bien arme mais qu'elle est completement ivre

Vous pouvez desormais utiliser le voyage rapide pour ce lieu grace a la commande aller(entree_grotte)

Vous : Je pourrais l'attaquer ... mais suis-je assez puissant ?
 Astuce : - sequiper pour equiper vos meilleurs equipements.
 - stats pour connaitre vos stats.
 - stats(Ennemi) pour connaitre les stats d'un ennemi.
 - attaquer(Ennemi) pour attaquer.

Vous : Ou bien je continue mon chemin...

DIRECTIONS POSSIBLES

```
Aller a la mine      : Nord (n.)
Aller dans la foret  : Est (e.)

Il y a un(e) orc ici.

true.

?- attaquer(orc).
Vous decider d'attaquer orc
Il vous a vaincu.
GAME OVER
La partie est termine. Tapez la commande "halt."
true.
```

L'orc étant plus fort que le joueur sans arme ni armure d'où la fin de la partie.

CONCLUSION

DIFFICULTES RENCONTREES

GIT

La principale difficulté en termes de versionning est l'acte de *merging* imposé par la forme du code, qui a impliqué parfois des problèmes de conflit sur certains passages du code impliquant la non-compilation du code.

PROLOG

Outre les difficultés liées au langage en lui-même qui diffère vraiment des autres langages informatiques, deux difficultés ont été rencontrées lors du projet à savoir :

- La gestion du « ! » dans les prédicats qui a engendré un certain nombre d'erreurs dans le développement notamment des réactions non-voulues du programme lors des requêtes.
- La coordination de différents fichiers (le choix fait étant de fractionner le programme en plusieurs fichiers par « fonctionnalités » pour gagner en compréhension du code) qui a été solutionné par la création de prédicats *consult/1* au sein du code. Dans notre cas, le fichier *LesChroniquesDePerceval.pl* contient les prédicats suivants :

POINT D'AMÉLIORATION

SYSTEME DE COMBAT

La logique de combat est relativement binaire dans le sens où le succès d'un combat est régi par la comparaison de coefficients qui sont plus ou moins forts selon la puissance du monstre ou du joueur : En d'autres termes, si le joueur a un coefficient plus grand que ce qu'il affronte, le joueur gagnera. Cependant, la formulation indique deux coefficients : un d'attaque et un de défense qui ont pour but de donner la possibilité d'améliorer le système de combat, en implémentant un système de combat au tour par tour comme dans beaucoup de RPG.

DEVELOPPER LE PREDICAT CONSTRUIRE

Le prédicat construire est très peu exploité (il est utilisé pour la construction du bateau) :

```
construire(bateau) :-
    joueur_est(riviere),
    il_y_a(bateau, riviere),
    ecrire_gras_couleur('Il y a déjà un bateau ici, pas la peine d'en reconstruire un !',
red), nl, !.
construire(bateau) :-
    joueur_est(riviere),
    equipement(arme, Objet),
    equivalent(Objet, marteau, _),
    verification(bois, 100, montrer_erreur),
    verification(cuivre, 20, montrer_erreur),
    verification(cuir, 20, montrer_erreur),
    verification(or, 2000, montrer_erreur),
    assert(il_y_a(bateau, riviere)),
    retirer(bois, 100),
    retirer(cuivre, 20),
    retirer(cuir, 20),
    retirer(or, 2000),
    ecrire_gras_couleur('Vous avez construit un bateau !', blue), nl,
    !.
construire(bateau) :-
    (joueur_est(riviere) ; ecrire_gras_couleur('Vous n'etes pas a la riviere !', red), nl
),
```

```
ecrire_gras_couleur('Il vous faut 100 bois, 20 cuivres, 20 cuirs, et 2000 or pour une  
telle construction.', red), nl,  
!.
```

C'est pour cette raison qu'il constitue une piste d'amélioration à court-termes afin d'ajouter d'autres objets au jeu en dehors des armes et des outils.

PERMETTRE L'ACHAT D'OBJET AUPRES DU MARCHAND

L'idée est de créer un marché permettant d'acheter des objets auprès du marchand puisqu'actuellement, il est seulement possible de lui vendre des objets.

APPORTS DU PROJET

Le projet a permis la manipulation du langage Prolog dans un cadre alternatif. Il a également permis d'avoir des considérations sur la façon de « construire » des jeux vidéo permettant alors d'appliquer le cours de « Fateur Humain » sur l'ergonomie des jeux vidéo, notamment en termes de gameplay et de level-design.