

Nolan LANCASTRE
Alexandre SAVARIRADJANE

Rapport TBA : Time Traveler

SOMMAIRE :

1. Scénario.....	3
2. Différents lieux.....	3
3. Carte du jeu.....	4
4. Comment jouer ?.....	5
4.1 Commencement	
4.2 Les époques classiques	
4.3 Les époques apocalyptiques	
5. Scénario gagnant.....	6
6. Scénario perdant.....	6
7. Commandes du jeu.....	7
8. Diagramme de classes.....	11
9. Perspectives de développement.....	14
10. Vidéo de démonstration.....	15

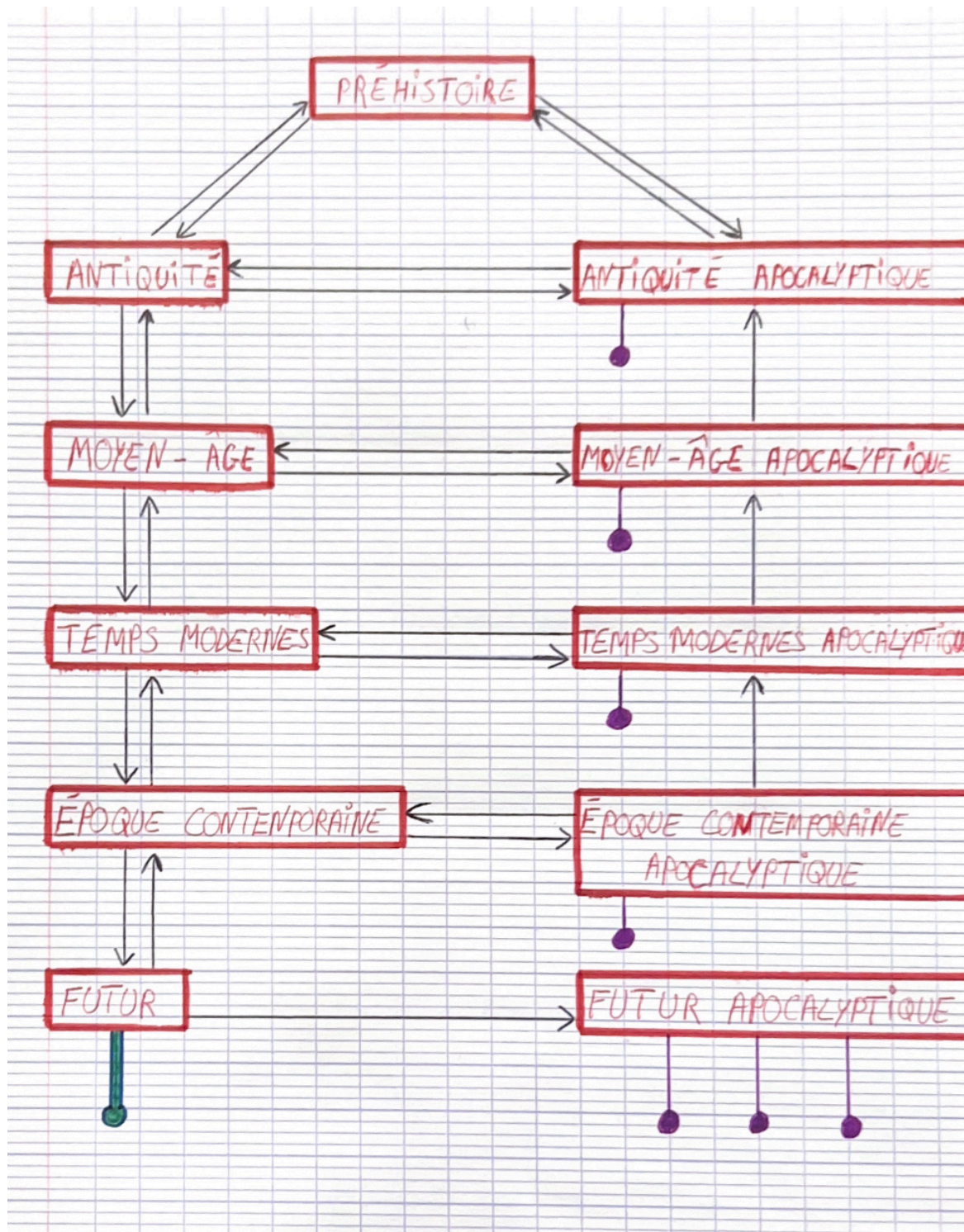
1.Scénario :

Un étudiant de l'Esiee est envoyé dans le passé. Son objectif est de revenir à notre époque. Pour cela, il doit résoudre des énigmes et prendre des décisions qui le feront changer d'époque selon les réponses qu'il donnera : il change d'époque en avançant et en reculant dans le temps et est aussi susceptible de passer dans des mondes apocalyptiques parallèles. Les lieux de notre scénario sont les différentes époques et leur monde apocalyptique parallèle.

2. Différents lieux :

1. La Préhistoire
2. L'antiquité
3. L'Antiquité Apocalyptique
4. Le Moyen âge
5. Le Moyen âge Apocalyptique
6. Les Temps Modernes
7. Les Temps Modernes Apocalyptique
8. L'époque contemporaine
9. L'Époque contemporaine Apocalyptique
10. Le Futur
10. Le Futur Apocalyptique

3. Carte du jeu :



- Les **rectangles rouges** représentent les différents lieux, dans notre jeu ils correspondent à des époques.
- Les **flèches noires** représentent les correspondances possibles entre les lieux.
- Les **chemins violets** représentent les scénarios perdants.
- Le **chemin vert** représente le scénario gagnant.

4. Comment jouer ?

4.1 Le commencement

Le jeu commence à la Préhistoire, une question vous est posée, si votre réponse est correcte vous êtes envoyé vers “ l'Antiquité ”, si elle est fausse vous êtes envoyé vers “ l'Antiquité apocalyptique ”.

4.2 Les époques classiques

Lorsque vous vous situez dans une époque standard (Antiquité, Moyen-Age, Temps Modernes, Époque contemporaine, Époque futuriste), une question concernant l'époque concernée vous est posée.

Vous aurez systématiquement trois réponses possibles.

Votre choix influencera directement votre voyage, en effet parmi les réponses possibles, une seule sera correcte et vous permettra d'avancer à l'époque suivante, une autre sera fausse et vous renverra à l'époque précédente, la dernière réponse elle, est absurde, si vous avez eu le malheur de la choisir elle vous enverra vers une époque parallèle apocalyptique.

4.3 Les époques apocalyptiques

Lorsque vous vous situez dans une époque apocalyptique, vous n'avez plus droit à l'erreur : tout comme les époques standards, trois réponses vous sont proposées, la bonne réponse vous renverra vers l'époque parallèle standard, la mauvaise, vous enverra vers l'époque apocalyptique précédente et si vous choisissez la réponse absurde vous avez perdu.

5. Scénario gagnant

Pour gagner, il faut que le joueur arrive dans le futur en ayant ramassé l'item "machine_enigma" et qu'il trouve la bonne réponse à la question posée dans le futur.

6. Scénario perdant

Concrètement il existe 7 façons de perdre dans notre jeu:

- Vous pouvez perdre directement dans chaque époque apocalyptique si la réponse que vous avez choisi est absurde.
(4 morts possibles, une par époque apocalyptique sauf pour le futur apocalyptique)

- Vous pouvez aussi perdre si votre réponse à la question posée à l'époque futuriste est absurde, car après cette réponse vous serez transporté vers un futur futur apocalyptique, dans ce futur peu importe votre réponse, vous aurez perdu.
(3 morts possibles, une pour chaque réponse à la question)

7. Commandes du jeu

Voici les différentes commandes que le joueur peut effectuer :

- go <direction> : permet au joueur d'avancer dans la direction spécifiée. Les directions possibles sont 1, 2, ou 3 selon la configuration des pièces.

```
> go 1
lucy se déplace vers Antiquity_apocalyptic.
socrate reste sur place.
jeanne_d_arc se déplace vers Modern_period.
napoleon reste sur place.
alan_turing se déplace vers Modern_period.
elon_musk reste sur place.
elon_musk se déplace vers Contemporary_times.
```

Vous êtes à l'antiquité.Quand ont eu lieu les premiers Jeux Olympiques ?

1 : Avant l'Antiquité.

2 : Après l'Antiquité.

3 : Pendant l'Antiquité.

Que choisissez-vous ?1, 2, 3

Vous avez déjà visité les pièces suivantes:

Vous n'avez visité aucune époque

- quit : pour quitter le jeu. Le jeu va renvoyer le message :
"Merci d'avoir joué !"

```
> quit  
  
Merci Alexandre d'avoir joué. Au revoir.  
  
Merci d'avoir joué !
```

- help : affiche les commandes du jeu disponible

```
> help  
  
Voici les commandes disponibles:  
- help : afficher cette aide  
- quit : quitter le jeu  
- go <direction> : se déplacer dans une direction cardinale (N, E, S, O)  
- back : retourner en arrière  
- inventory : afficher l'inventaire des objets  
- inventory lieu : afficher l'inventaire des objets présents dans la pièce  
- look : afficher les objets présents dans la pièce actuelle  
- take : prendre les objets présents dans la pièce actuelle  
- drop : déposer les objets présents dans la pièce actuelle  
- talk : faire parler les PNJ
```

- back : permet au joueur de retourner dans l'époque précédente.

```
> back  
  
Vous êtes de retour dans l'époque précédente :  
Vous êtes à l'antiquité.Quand ont eu lieu les premiers Jeux Olympiques ?  
1 : Avant l'Antiquité.  
2 : Après l'Antiquité.  
3 : Pendant l'Antiquité.  
  
Que choisissez-vous ?1, 2, 3
```


- inventory : permet au joueur de regarder son inventaire afin de voir quel item il a ou n'a pas

```
> inventory
```

Vous disposez des items suivants :

```
- torch : une torche flamboyante éclairant comme le soleil, (1 kg)
```

- inventory_lieux : permet au joueur de regarder l'inventaire des items présents dans l'époque où il se trouve

```
> inventory_lieux
```

La pièce contient:

```
-- torch : une torche flamboyante éclairant comme le soleil, (1 kg)
```

- look : fait appel à inventory_lieux() pour regarder l'environnement et avoir une description de l'époque ainsi que des items présents

```
> look
```

On voit:

```
-- vase : un vase décoré avec des motifs de l'Antiquité grecque, (2 kg)  
-socrate, un vieil homme sage
```

- take <item> : permet au joueur de ramasser un item présent dans l'époque, si elle en contient

```
> take vase
```

```
vous avez ramasser,vase!
```

- drop <item> : permet au joueur de déposer un objet dans la pièce actuelle s'il en a.

```
> drop vase
```

```
vous avez déposé,vase!
```

- talk <pnj> : permet d'interagir avec les pnj

```
> talk Socrate
```

```
Connais-toi toi même...
```

```
> talk Socrate
```

```
Je ne suis ni Athénien ni grec, je suis un citoyen du monde
```

```
> talk Socrate
```

```
Tout ce que je sais, c'est que je ne sais rien
```

8. Diagramme des classes

Class Room :

Room
characters : dict description exits : dict inventory : dict inventory_lieux : set items : list name question : str reponse : dict
add_character(character) get_exit(direction) get_exit_string() get_inventory() get_inventory_lieux() get_long_description() remove_character(character_name)

Class Characters :

Characters
current_room description msgs name
get_msgs() move()

Class Actions :

Actions
commands player
back(game, list_of_words, number_of_parameters) drop(game, list_of_words, number_of_parameters) go(game, list_of_words, number_of_parameters) help(game, list_of_words, number_of_parameters) inventory(game, list_of_words, number_of_parameters) inventory_lieux(game, list_of_words, number_of_parameters) look(game, list_of_words, number_of_parameters) quit(game, list_of_words, number_of_parameters) take(game, list_of_words, number_of_parameters) talk(game, list_of_words, number_of_parameters)

Class Command :

Command
action command_word help_string number_of_parameters
execute()

Class Inventory :

Inventory
items : set
add_item(item) get_inventory()

Class Game :

Game
Actions : Actions characters : dict commands : dict defeat : bool finished : bool items : dict player : NoneType, Player rooms : list victory : bool
play() print_welcome() process_command(command_string): None setup()

Class Player :

Player
current_room : NoneType game : NoneType history : list inventory : dict inventory_lieux : tuple name question_answered : bool
get_history() get_inventory() get_inventory_lieux() move(direction)

Class Item :

Item
description name weight
get_weight()

-

9. Perspectives de développement

Concernant les perspectives de développement du jeu, nous avons pensé à plusieurs points.

Premièrement, nous aurions pu implémenter une fonction qui puisse faire en sorte que le joueur puisse utiliser son item (sans compter le fait qu'il puisse le ramasser et le déposer).

Ensuite, nous souhaitons faire une fonction où les questions étaient posées par les pnj, ce qui aurait rendu le jeu plus interactif.

Nous aurions également aimé avoir plus de dialogue avec les pnj.

De plus, une interface graphique aurait été plus agréable.

10.Vidéo de démonstration

Lien pour la vidéo de démonstration :

<https://drive.google.com/file/d/1PnP75rLnt-xTbGEYnINTcrk4VsoaZTP9/view?usp=sharing>