

CAHIER DE CONCEPTION



ESCAPE GAME
ESCAPE GAME



Type	Cahier de Conception
Nom du projet	Escape Game
Auteur	Corentin Colivet (c9colive@enib.fr), Éléonore Chalkidis (e9chalki@enib.fr)
Version	1.0
Date	26 / 04 / 2020

Table des Matières

1. Rappel du cahier des charges.....	4
1.2 Fonctionnalités	4
1.3 P1 : Prototype P1	5
1.4 P2 : Prototype P2	5
2. Principes des solutions techniques	5
2.1 Langage	5
2.2 Architecture du logiciel	5
2.3 Interface utilisateur	5
2.3.1 Boucles de simulation.....	5
2.3.2 Affichage	5
2.3.3 Gestion du clavier	6
2.3.4 Image ascii-art	6
2.4 Grille, avatar,	6
3. Analyse.....	6
3.1 Analyse noms / verbes :.....	6
3.2 Types de donnée.....	6
3.3 Dépendance entre modules	7
3.4 Analyse descendante :	8
3.4.1 Arbre principal :	8
3.4.2 Arbre d'affichage	8
3.4.3 Arbre d'interaction	8
4. Description des fonctions	9
4.1 Programme Principal : Main.py.....	9
4.2 Player.py	10
4.3 Background.py	11
4.4 Enigma.py	12
4.5 Help.py	12
4.6 interactEnigmes.py	13
4.7 Information.py	13
4.8 timerEG.py	14
4.9 Introduction.py	14

5.	Calendrier et suivi de développement	15
5.1	P1 :	15
5.1.1	Fonctions à développer	15
5.1.2	Autre	16
5.2	P2 :	16
5.2.1	Autre	17

1. Rappel du cahier des charges

1.1 Contraintes techniques

- ❖ Le logiciel est associé à un cours, il doit donc fonctionner sur les machines de TP de l'ENIB pour que les élèves puissent le tester.
- ❖ Le langage utilisé en cours est Python 2.7.5. Le développement devra donc se faire en python 2.7.5.
- ❖ Les notions de programmation orientées objet n'ayant pas encore été abordées, le programme devra essentiellement s'appuyer sur le paradigme de la programmation procédurale.
- ❖ Le logiciel devra être réalisé en conformité avec les pratiques préconisées en cours de IPI : barrière d'abstraction, modularité, Unicode, etc...
- ❖ L'interface sera réalisée en mode texte dans un terminal

1.2 Fonctionnalités

❖ F1 : jouer une partie

❖ F2.1 Afficher le jeu

- Grille
- Le décor
- Le personnage (Jean Michel)
- Barre informations
- Carnet de notes

❖ F2.2 Se déplacer sur la carte

❖ F2.3 Entrer en collision avec un objet

❖ F2.4 Intéragir avec le menu d'un objet

❖ F2.5 Trouver une énigme

❖ F2.6 Répondre à l'énigme

❖ F2.7 Demander de l'aide

❖ F2.8 Obtenir un indice

❖ F2.9 Finir la partie

❖ F3 Finir la partie

- ❖ F3.1 Afficher le résultat
- ❖ F3.2 Afficher le temps mis pour finir le jeu
- ❖ F3.3 Quitter

❖ F4 Sauvegarder la partie

1.3 P1 : Prototype P1

Ce prototype porte essentiellement sur la création de la carte et de l'affichage.

Mise en œuvre des fonctionnalités : F1, F2.1, F2.2, F2.3, F2.7, F3.3.

Livré dans une archive au format .zip ou .tgz. Contient un manuel d'utilisation dans le fichier readme.txt

1.4 P2 : Prototype P2

Ce prototype réalise toutes les fonctionnalités.

Livré dans une archive au format .zip ou .tgz. Contient un manuel d'utilisation dans le fichier readme.txt

2. Principes des solutions techniques

2.1 Langage

Conformément aux contraintes énoncées dans le cahier des charges, le codage est réalisé avec langage python.
Nous choisissons la version 2.7.5.

2.2 Architecture du logiciel

Nous mettons en œuvre le principe de la barrière d'abstraction. Chaque module correspond à un type de donnée et fournit toutes les opérations permettant de le manipuler de manière abstraite.

2.3 Interface utilisateur

L'interface utilisateur se fera via un terminal de type linux.

Nous reprenons la solution donnée en cours de IPI en utilisant les modules : sys, tty, os, select, et termios

2.3.1 Boucles de simulation

Le programme mettra en œuvre une boucle de simulation qui gèrera l'affichage et les événements clavier.

2.3.2 Affichage

L'affichage se fait en communiquant directement avec le terminal, en envoyant des chaînes de caractères sur la sortie standard de l'application.

2.3.3 Gestion du clavier

L'entrée standard est utilisée pour détecter les actions utilisateur. Le module tty permet de rediriger les événements clavier sur l'entrée standard. Pour connaître les actions de l'utilisateur il suffit de lire l'entrée standard.

2.3.4 Image ascii-art

Pour dessiner certaines parties de l'interface nous utilisons des « images ascii ».

Dans l'idée de séparer le code et les données, les différentes images ASCII seront stockées dans des fichiers textes : grid.txt, map.txt, win.txt, lose.txt par exemple.

2.4 Grille, avatar, ...

Pour modéliser la carte du jeu, on importe une image du type ASCII : map.txt. Des caractères de la forme d'un X seront présent sur la carte du jeu. Ils correspondent à des objets et donc à diverses zones à explorer afin de trouver des indices et se libérer.

3. Analyse

3.1 Analyse noms / verbes :

❖ Verbes :

Choisir, jouer, afficher, déplacer, entrer, finir, quitter, trouver, valider, répondre, demander, obtenir, mettre, sauvegarder, récupérer

❖ Noms :

Partie, jeu, personnage (Jean Michel), carte, collision, temps, menu, barre d'information, indice, énigme, aide, partie, résultat, zones à explorer ou objet.

3.2 Types de donnée

Type : Game = structure

Time	: entier
Index	: chaîne de caractères/entier.....
Enigma	: chaîne de caractères
Result	: chaîne de caractères
Player	: caractère

Fin structure

3.4 Analyse descendante :

3.4.1 Arbre principal :

Main.main ()

```
+ - - Main.init ()
|       + - - Background.create ()
|       + - - Notebook.create ()
|       + - - Introduction.create ()
|       + - - Help.Create ()
|       + - - timerEG.create ()
|       + - - Information.Create ()
|       + - - grille.create ()
|       + - - Player.create ()
|       + - - lagrossedefaite.create ()
|
+ - - Main.run ()
      + - - Main.show ()
      + - - Main.Interact ()
      + - - Main.move ()
      + - - Main.time.sleep ()
```

3.4.2 Arbre d'affichage

Main.show ()

```
+ - - Background.show()
+ - - Time.show ()
+ - - Help.show ()
+ - - Information.show ()
+ - - Player.show ()
+ - - grille.show ()
```

3.4.3 Arbre d'interaction

Main.Interact ()

```
+ - - MoveRight ()
+ - - MoveLeft ()
+ - - MoveForward ()
+ - - MoveBackward ()
+ - - helpME ()
+ - - QuitGame()
+ - - Information ()
```


4. Description des fonctions

4.1 Programme Principal : Main.py

- Main.main()
- Main.init()
- Main.run()
- Main.move()
- Main.show()
- Main.interact()
- Main.QuitGame()

Main.main () à rien

Description : fonction principale du jeu

Paramètres : aucun

Valeur de retour : aucune

Main.init () à rien

Description : initialisation du jeu

Paramètres : aucun

Valeur de retour : aucune

Main.run () à rien

Description : boucle de simulation

Paramètres : aucun

Valeur de retour : aucune

Main.move () à rien

Description : gère les déplacements du joueur et les collisions

Paramètres : aucun

Valeur de retour : aucune

Main.show () à rien

Description : gère l’affichage du jeu en entier (joueur, carte, information, help, ...)

Paramètres : aucun

Valeur de retour : aucune

Main.interact () à rien

Description : gère les événements clavier (interactions avec les touches)

Paramètres : aucun

Valeur de retour : Affichage

Main.QuitGame () à rien

Description : terminer le jeu

Paramètres : aucun

Valeur de retour : Rien

4.2 Player.py

- Player.createPlayer ()
- Player.show ()
- Player.getNextPosition ()
- Player.move ()
- Player.MoveRight ()
- Player.MoveLeft ()
- Player.MoveForward ()
- Player.MoveBackward ()

Player.createPlayer () à rien

Description : création du dictionnaire MyPlayer

Paramètres : color, x, y, xMax, yMax, speed, direction

Valeur de retour : MyPlayer

Player.show () à rien

Description : affichage du joueur

Paramètres : a

Valeur de retour : Rien

Player.getNextPosition () à rien

Description : gère la position et la vitesse du joueur

Paramètres : dt, a

Valeur de retour : x, y

Player.move () à rien

Description : gère les mouvements du joueur (ou Player)

Paramètres : dt, a

Valeur de retour : Rien

Player.MoveRight () à rien

Description : gère le déplacement du Player vers la droite

Paramètres : a

Valeur de retour : Rien

Player.MoveLeft () à rien

Description : gère le déplacement du Player vers la gauche

Paramètres : a

Valeur de retour : Rien

Player.MoveForward () à rien

Description : gère le déplacement du Player vers le haut

Paramètres : a

Valeur de retour : Rien

Player.MoveBackward () à rien

Description : gère le déplacement du Player vers le bas

Paramètres : a

Valeur de retour : Rien

4.3 Background.py

- Background.create ()
- Background.GetCaractere ()
- Background.show ()

Background.create () à rien

Description : création du dictionnaire background

Paramètres : filename

Valeur de retour : background

Background.GetCaractere () à rien

Description : récupère l'affichage du background et de caractère (Player)

Paramètres : background, x, y

Valeur de retour : c

Background.show () à rien

Description : gère l'affichage du background et de caractère (Player)

Paramètres : background

Valeur de retour : Rien

4.4 Enigma.py

- Enigma.enigme1 ()
- Enigma.enigme2 ()
- Enigma.enigme3 ()
- Enigma.enigme4 ()
- Enigma.enigme5 ()
- Enigma.clue1 ()
- Enigma.clue2 ()
- Enigma.clue3 ()

Enigma.enigme1 () à rien

Description : gère l’affichage de l’énigme 1

Paramètres : aucun

Valeur de retour : text1

Dans chaque cas, que ce soit une énigme ou un indice (clue) :

- La description reste la même (seul le numéro change et la précision indice ou énigme)
- Il n’y a aucun paramètre
- La valeur de retour est toujours texti, avec i variant (dans l’ordre) de 1 à 8

4.5 Help.py

- Help.Create ()
- Help.HelpME ()
- Help.show ()

Help.Create () à rien

Description : création du dictionnaire help

Paramètres : helpme

Valeur de retour : help

Help.HelpME () à rien

Description : gère l’affichage textuel pour la fenêtre help

Paramètres : aucun

Valeur de retour : e

Help.show () à rien

Description : gère l’affichage de la fenêtre help

Paramètres : help

Valeur de retour : Rien

4.6 interactEnigmes.py

- interactEnigmes.Enigme1 ()
- interactEnigmes.Enigme2 ()
- interactEnigmes.Enigme3 ()
- interactEnigmes.Enigme4 ()
- interactEnigmes.Enigme5 ()
- interactEnigmes.Enigme6 ()
- interactEnigmes.Enigme7 ()

interactEnigmes.Enigme1 () à rien

Description : gère l’affichage du menu de l’objet 1 (par exemple un bureau)

Paramètres : aucun

Valeur de retour : aucune

Dans chaque cas :

- La description reste la même (sauf l’objet change et son menu)
- Il n’y a aucun paramètre
- Il n’y a pas de valeur de retour

4.7 Information.py

- Information.Create ()
- Information.show ()

Information.Create () à rien

Description : création du dictionnaire info

Paramètres : information

Valeur de retour : info

Information.show () à rien

Description : gère l’affichage de la fenêtre information

Paramètres : info

Valeur de retour : Rien

4.8 timerEG.py

- timerEG.create ()
- timerEG.minuteur ()
- timerEG.showMinuteur ()

timerEG.create () à rien

Description : création du dictionnaire minuteur

Paramètres : inb_sec, timelni

Valeur de retour : minuteur

timerEG.minuteur () à rien

Description : définition du minuteur

Paramètres : timer

Valeur de retour : aucune

timerEG.showMinuteur () à rien

Description : gère l’affichage du minuteur

Paramètres : minuteur

Valeur de retour : aucune

4.9 Introduction.py

- Introduction.create ()
- Introduction.show ()

Introduction.create () à rien

Description : création du dictionnaire introduction

Paramètres : introduc

Valeur de retour : introduction

Introduction.show () à rien

Description : gère l’affichage du message d’introduction

Paramètres : introduction

Valeur de retour : Rien

5. Calendrier et suivi de développement

5.1 P1 :

5.1.1 Fonctions à développer

Fonctions	codées	testées	commentaires
Main. main()			
Main. init()			
Main. run()			
Main. interact()			
Main. move()			
Main. show()			
Player. createPlayer ()			
Player. show()			
Player. getNextPosition()			
Player. move()			
Player. MoveRight()			
Player. MoveLeft()			
Player. MoveForward()			
Player. MoveBackward()			
Background. create()			
Background. GetCaractere()			
Background. show()			
interactEnigmes. Enigme1() à interactEnigmes. Enigme7 ()			
Introduction. create ()			
Introduction. show ()			
Enigma. enigma1 () à Enigma. enigma5 ()			
Enigma. clue1 () à Enigma. clue3 ()			
Help. Create()			
Help. HelpME()			
Help. show()			

5.1.2 Autre

Grille_mots_croises.txt, Help.txt, Image.txt, information.txt, message_intro.txt

5.2 P2 :

Fonctions	codées	testées	commentaires
Main. main()			
Main. init()			
Main. run()			
Main. interact()			
Main. move()			
Main. show()			
Main. QuitGame()			
Player. createPlayer()			
Player. show()			
Player. getNextPosition()			
Player. move()			
Player. MoveRight()			
Player. MoveLeft()			
Player. MoveForward()			
Player. MoveBackward()			
Background. create()			
Background. GetCaractere()			
Background. show()			
interactEnigmes. Enigme1() à interactEnigmes. Enigme7 ()			
Introduction. create ()			
Introduction. show ()			
Enigma. enigma1 () à Enigma. enigma5 ()			
Enigma. clue1 () à Enigma. clue3 ()			
Help. Create()			

Help. HelpME()
Help. show()
Information. Create()
Information. show()

5.2.1 Autre

lose.txt, win.txt