

Projet ISN

Création d'un jeu avec Python et Pygame

MINIER Arnaud T°S1
2018-2019

Principe du jeu et inspirations



028 30/30 picarats 0:020

Un amateur de chats est si fier de sa collection de presse-papiers qu'il aimerait la mettre en valeur sur une grille de quatre carrés de côté. Il ne veut ni laisser de case vide, ni avoir plus de deux chats de la même couleur côte à côte, que ce soit sur une rangée ou une colonne. Les premiers chats sont déjà en place et ne peuvent être modifiés. À vous de compléter cet arrangement. Touchez une case pour y placer un chat, changer sa couleur ou le faire disparaître.



005 50 picarats 0:16

Une horloge à aiguilles standard est composée d'une aiguille pour les heures et d'une aiguille plus longue et plus rapide pour les minutes.

En supposant que cette horloge soit parfaitement à l'heure, combien de fois l'aiguille des minutes dépasse-t-elle l'aiguille des heures entre midi et minuit?



Cahier des charges

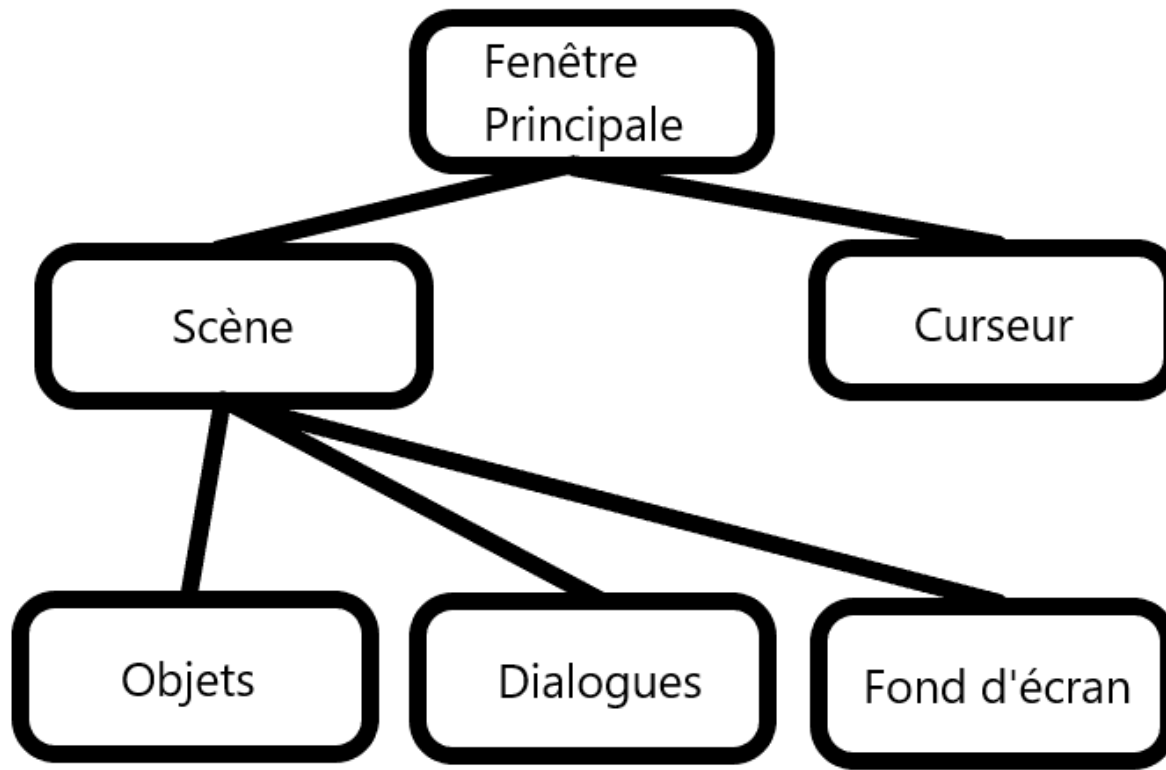
Le jeu doit contenir :

- une douzaine de salles où l'on peut se déplacer
- une demi-douzaine d'énigmes à résoudre
- une histoire complète, c'est-à-dire une introduction, un développement et une fin, transcrite à travers des dialogues
- un système de sauvegarde, pour ne pas avoir à recommencer quand on ferme le jeu

Répartition des tâches

| Arnaud | Valentin | Yuehan |
|---|---|---|
| <ul style="list-style-type: none">-« Moteur » du jeu :<ul style="list-style-type: none">→ structure (Fenêtre principale, Scène, Objet, etc.)→ gestion des sauvegardes→ gestion des dialogues→ gestion des évènements | <ul style="list-style-type: none">-Menu-Salles :<ul style="list-style-type: none">-Pilote-Vaisseau-SalleUne-CombinaisonsEquipage-CouloirUn-SalleControle-CouloirDeux-SalleDeux-CouloirTrois-SalleSoins-Enigmes :<ul style="list-style-type: none">-A contretemps-Et la perdante est...-Jour après Jour-Infirmière-Roi de la glisse-Digicode (1, 2 et 3)-Dialogues-Graphismes-Musique-Histoire | <ul style="list-style-type: none">-Salles :<ul style="list-style-type: none">-GrosseSalle-SalleTrois-SalleRepos-ChambreUne-CouloirCinq-Generateur-SalleFinale-Enigmes :<ul style="list-style-type: none">-ArchéoLogique-Fragment d'histoire-Freres et Sœurs-Câble électricité |

Structure du jeu



Fichier Main.py :

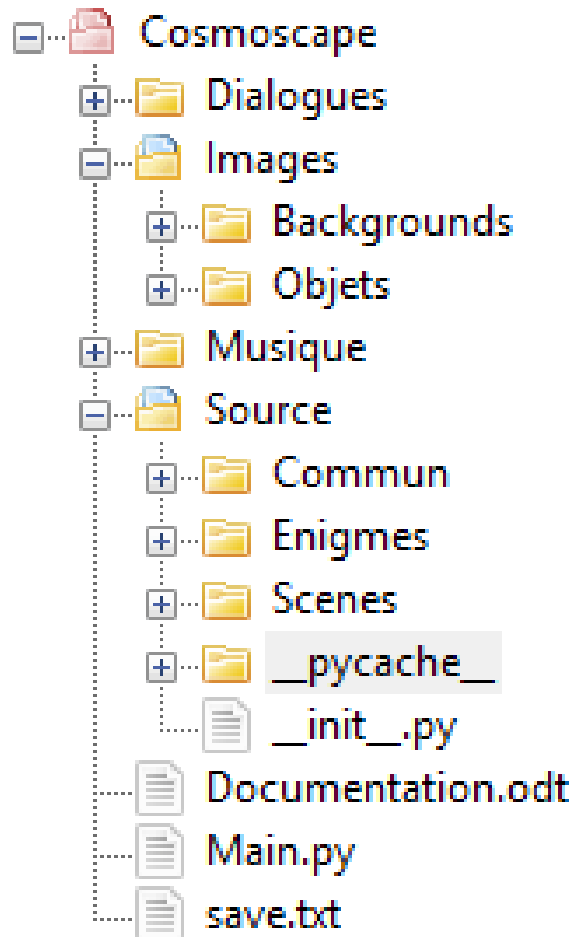
```
import pygame

from Source.Commune.FenetrePrincipale import *

if __name__ == '__main__':
    pygame.init()
    fenetre= FenetrePrincipale()
    fenetre.exec()
```

Organisation des fichiers :

Arbre des fichiers :



Exemple d'importation des fichiers dans FenetrePrincipale.py:

```
import pygame
from pygame.locals import *

from Source.Enigmes.EnigmeUne import *

from Source.Scenes.CombinaisonsEquipage import *
from Source.Scenes.CouloirUn import *
from Source.Scenes.Menu import *
from Source.Scenes.Pilote import *
from Source.Scenes.SalleUne import *
from Source.Scenes.Vaisseau import *
from Source.Scenes.SalleControle import *
from Source.Scenes.CouloirDeux import *
from Source.Scenes.SalleDeux import *
from Source.Scenes.GrosseSalle import *
from Source.Scenes.CouloirTrois import *
from Source.Scenes.SalleSoins import *
from Source.Scenes.SalleTrois import *
from Source.Scenes.CouloirCinq import *
from Source.Scenes.SalleRepos import *
from Source.Scenes.ChambreUne import *
from Source.Scenes.SalleQuatre import *
from Source.Scenes.Generateur import *
```

La fenêtre principale

```
def exec(self):
    while 1:
        for event in pygame.event.get():
            if event.type is QUIT:
                pygame.display.quit()
                self.arreteMusique()
                return
            elif event.type is MOUSEBUTTONDOWN:
                #Le bouton a l'index 1 est le clic gauche
                if event.__dict__["button"] == 1:
                    self.scene.click(event.__dict__["pos"])
            elif event.type is MOUSEMOTION:
                self.positionCurseur=event.__dict__["pos"]
                self.bougerCurseur()
            elif event.type is USEREVENT:
                if "changerScene" in event.__dict__:
                    self.changerScene(event.__dict__["changerScene"])
                else:
                    self.scene.manageEvent(event)
        self.draw()
        pygame.display.flip()
        self.scene.onFrameUpdate()
        pygame.time.wait(self.tempsParFrame)

def changerScene(self,nomScene):
    print("Nom de scene changée:",nomScene)
    self.scene = self.dictScenes[nomScene]()
    self.changeTitre(nomScene)
    self.bougerCurseur()
```

La fenêtre principale gère :

- l'initialisation de tous les éléments du jeu
- les différents événements
- le rendu du jeu
- la musique
- la fermeture du programme
- le changement de scène

Dessiner les objets

Représentation d'un rectangle dans pygame :

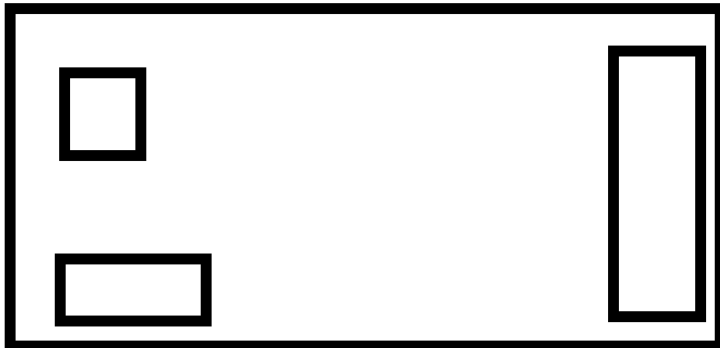
(x,y)



Largeur

Hauteur

Schéma d'une scène :



Fonction FenetrePrincipale.draw() :

```
def draw(self):  
    self.scene.draw(self.ecran)  
    self.curseur.draw(self.ecran)
```

Fonction Scene.draw() :

```
def draw(self,surface):  
    surface.blit(self.background, (0,0))  
    for objet in self.listeObjet:  
        objet.draw(surface)  
    if not self.dialogueActifFerme :  
        self.listeDialogues[self.dialogueActif].draw(surface)
```

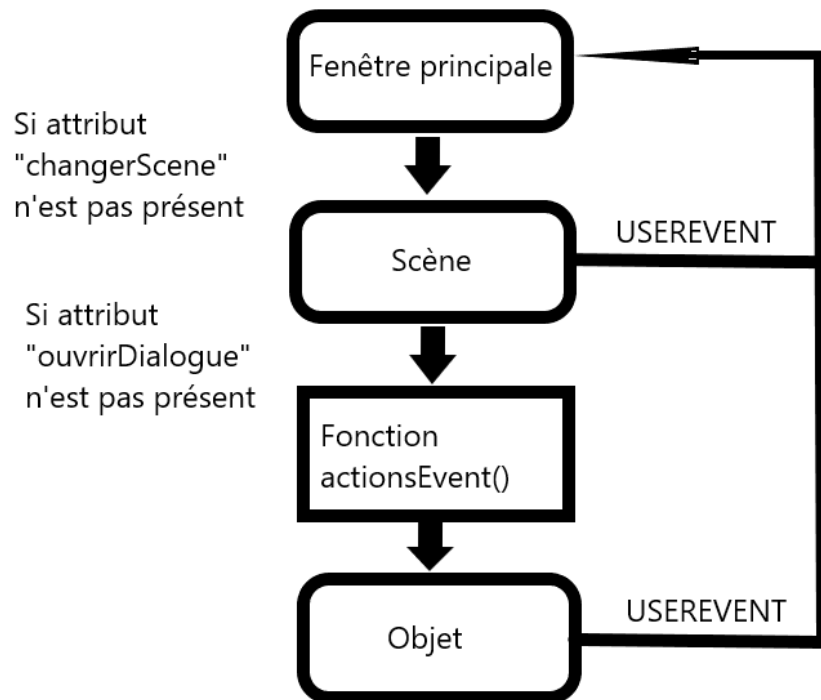
Fonction Objet.draw() :

```
def draw(self,surface):  
    if self.visibility == True:  
        surface.blit(self.image, self.rect)
```

Fonction Dialogue.draw() :

```
def draw(self,surface):  
    surface.blit(self.imageFinale,self.rect)
```


Gestion des évènements



Fonction commune à Scene et à Objet :

```
def ordonnerAction(self,nomAction,valeurAction=""):  
    dictarg = {nomAction:valeurAction}  
    evt = pygame.event.Event(USEREVENT,dictarg)  
    pygame.event.post(evt)
```

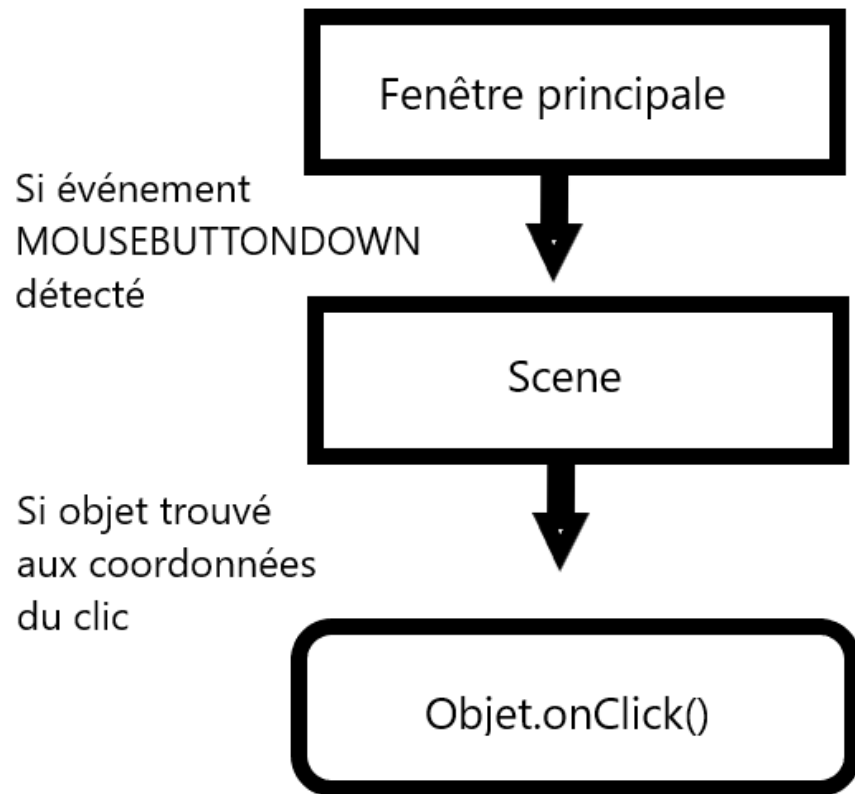
Partie s'occupant des événements
USEREVENT de la fenêtre principale :

```
elif event.type is USEREVENT:  
    if "changerScene" in event.__dict__:  
        self.changerScene(event.__dict__["changerScene"])  
    else:  
        self.scene.manageEvent(event)
```

Partie s'occupant des événements de
Scene :

```
def manageEvent(self,event):  
    if "ouvrirDialogue" in event.__dict__:  
        balise = event.__dict__["ouvrirDialogue"]  
        self.ouvrirDialogue(balise)  
    else:  
        self.actionsEvent(event.__dict__)
```

Gestion des clics



Partie s'occupant des événements MOUSEBUTTONDOWN de la fenêtre principale :

```
elif event.type is MOUSEBUTTONDOWN:
    #Le bouton a l'index 1 est le clic gauche
    if event.__dict__["button"] == 1:
        self.scene.click(event.__dict__["pos"])
```

Fonction Scene.click() :

```
def click(self, positionSouris):
    if self.dialogueActifFerme:
        objet = self.getObjet(positionSouris)
        if objet is not None:
            objet.onClick()
    elif self.dialogueActifFini :
        action = self.fermerDialogue()
        self.actionsDialogue(action)
    else:
        self.avancerDialogue()
```

Conclusion

