

ENIB 2012 : S2P MDD

Medius

—
Document de conception



Valérien Saliou (v1saliou) & Kwon-Young Choi (k1choi)

March 26, 2012

1 Rappel des versions

1.1 Version 1 : accéder au menu

- afficher le fond animé (2 couches)
- configurer le son (effets et musique)

1.2 Version 2 : accéder aux niveaux

- afficher les niveaux
- lancer un niveau
- obtenir des détails sur un niveau (document d'aide)

1.3 Version 3 : accéder aux combats #1

- combattre une bactérie
- afficher les organismes ennemis
- gérer les tirs
- gérer les vies
- gérer les collisions
- gérer la vitesse

1.4 Version 4 : accéder aux combats #2

- combattre un virus
- gérer les maladies (1 scénario par maladie, virus ou bactérie)

2 Analyse nom-verbe

- Fond de menu
 - afficher
- Boutons
 - afficher
 - créer handlers de clic
- Jeu
 - accéder à la liste des niveaux
- Configuration
 - afficher les checkboxes
 - callback au changement de valeur

3 Définition des types de données

```
type menu = struct
    background : Background
    decor : Decor
    bouton : Bouton
    checkbox : Checkebox
fstruct

type Background = struct
    sprite : Sprite
    speed : Vec
fstruct

type Decor = struct
    spriteTop : Sprite
    spriteBottom : Sprite
    spriteLogo : Sprite
    true : Bool
fstruct

type Bouton = struct
    sprite : Sprite
    target : Function
    true : Bool
fstruct

type Checkebox= struct
    sprite.uncheck : Sprite
    sprite.check : Sprite
    target : Sound
    check : Bool
    true : Bool
fstruct

type Game :
    background : Background
    bacteria : Bacteria
    phagocyte : Phagocyte

type Bacteria =struct
    sprite : Sprite
    speed : Vec
    animation.attaque : Anim
    animation.duplication : Anim
```

```

        animation.destruction : Anim
    fstruct
type Phagocyte =struct
    sprite : Sprite
    speed : Vec
    animation.attaque : Anim
    animation.destruction : Anim

Sprite = struct
    image : Image
    position : Vec
    size : Vec
    fstruct

Vec = struct
    a : int
    b : int
    fstruct

Anim = struct
    image1 : Image
    image2 : Image
    image3 : Image
    image4 : Image
    image5 : Image
    position : Vec
    fstruct

Function = struct
    module : Text
    function : Text
    fstruct

Sound = struct
    pathSound1 : Text
    pathSound2 : Text
    pathSound3 : Text
    pathSound4 : Text
    lengthSound1 : float
    lengthSound2 : float
    lengthSound3 : float
    lengthSound4 : float
    fstruct

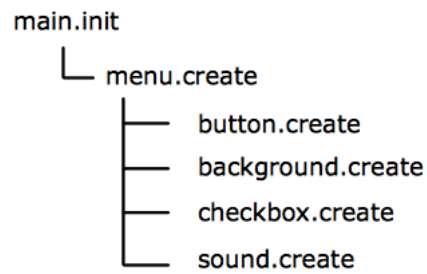
```

4 Modules

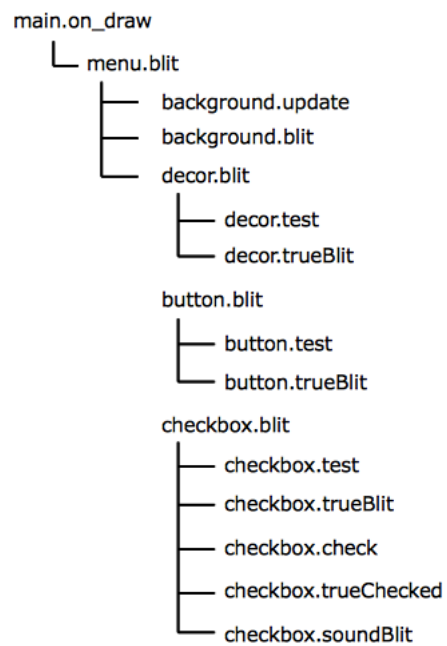
main.py : initialisation et définition des fonctions callback.
menu.py : gère le menu
background.py : gère le fond d'écran
decor.py : gère le décor
button.py : gère les boutons dans le menu
checkbox.py : gère les checkbox du menu
game.py : gère les appels de fonction pour le jeu
bacteria.py : gère les bactéries
phagocyte.py : gère le héros
sprite.py : gère les sprites
vec.py : gère les vecteurs
anim.py : gère les animations
function.py : gère les fonctions cibles des boutons
sound.py : gère les sons

5 Arbre des appels de fonctions

5.1 Initialisation



5.2 Affichage



6 Description synthétique des fonctions

```
main.init :  
    enchaîne les fonctions d'initialisation  
  
menu.create :  
    enchaîne les fonctions d'initialisation  
  
levels.create :  
    lister les niveaux  
  
levels.help :  
    obtenir de l'aide sur un niveau  
  
game.create :  
    créer un jeu  
  
background.create :  
    crée en mémoire une représentation du fond d'écran (image + position + speed)  
  
decor.create :  
    crée en mémoire une représentation du décor (images + position)  
  
button.create :  
    crée en mémoire une représentation d'un bouton (image + position + size + target)  
  
checkbox.create :  
    crée en mémoire une représentation d'un checkbox (images + position + size + target)  
  
sound.create :  
    crée en mémoire une représentation du son  
  
main.on_draw :  
    affiche l'image, appelé régulièrement par Pyglet  
  
menu.blit :  
    enchaîne les fonctions qui créent les représentations graphiques des différents objets  
  
background.update :  
    met à jour la position du fond d'écran  
  
background.blit :  
    affiche le fond d'écran  
  
decor.blit :  
    affiche le décor que si cela est requis.
```



```

decor.test :
    test si le décor doit être afficher ou non.
    Renvoie True si le décor doit être visible, False sinon

decor.trueBlit :
    affiche le décor

button.blit :
    affiche le bouton que si cela est requis.

button.test :
    test si le bouton doit être afficher ou non.
    Renvoie True si le bouton doit être visible, False sinon

button.trueBlit :
    affiche le bouton

checkbox.blit :
    affiche le checkbox que si cela est requis et en fonction de son état checké ou non

checkbox.test :
    test si le checkbox doit être afficher ou non.
    Renvoie True si le checkbox doit être visible, False sinon

checkbox.trueBlit :
    affiche le checkbox non checké

checkbox.check :
    test si le checkbox est checké ou non.
    Renvoie True si le checkbox doit être visible et checké, False sinon

checkbox.trueChecked :
    affiche le checkbox checké

main.on_mouse_press :
    gère les cliques de souris, appelée régulièrement.

menu.mousePressEvent :
    enchaîne les fonction lors d'un clic de souris.

menu.mousePosition :
    récupère la position de la souris lors d'un clic et l'envoie aux fonctions concernées

button.mousePressCheck :
    vérifie et renvoie True si le clic de la souris s'est effectué sur le bouton, False sinon

```

```
button.target :  
    appelle la fonction cible du bouton (par exemple la fonction game pour le bouton)  
  
checkbox.mousePressCheck :  
    vérifie si le clic s'est effectué sur la checkbox et marque si la checkbox est  
    cochée  
  
checkbox.target :  
    appelle la cible de la checkbox (par exemple de la musique)
```

7 Calendrier de développement

Notre développement est progressif. Dans un premier temps nous nous concentrons sur un menu propre et complètement fonctionnel, de même pour le sélecteur de niveaux. Seulement ensuite, nous développerons les niveaux avec la gestion des combats.

Il s'agit de poser des bases de travail dans un premier temps, nous permettant de disposer d'un environnement de jeu paramétrable sur lequel les niveaux viennent se greffer.

- Mars 2012 :
 - Création de `menu.create`
 - Création de `background.create`
 - Création de `button.create`
 - Création de `checkbox.create`
- Avril 2012 :
 - Création de `levels.create`
 - Création de `levels.help`
 - Création de `game.create`
- Mai 2012 (1ère moitié) :
 - Création de `bacteria.add`
- Mai 2012 (2nde moitié) :
 - Création de `virus.add`

8 Annexe : Description détaillée des fonctions

8.1 Module main.py

8.1.1 Fonction init

PARAMETRES :
aucun

RESULTAT :
aucun

ROLE :
enchaîne les fonctions d'initialisation

PRECONDITIONS :
le script est appelé en tant que parent, et non inclus dans un script parent

POSTCONDITIONS :
la configuration est chargée, le menu est créé, les boutons associés sont ajoutés