



PyMT, Framework multitouch





Historique de PyMT

- Créé en septembre 2008 par Thomas Hansen
- Projet soutenu par l'université de l'Iowa
- Équipe active de 5 développeurs
- Sortie de la version 0.1 le 09 février
- Projet soutenu activement par NuiGroup
- Sortie de la version 0.2 le 30 mars



Pourquoi PyMT

- Il faut revoir les interfaces actuelles pour le multitouch :
 - Événements multiples
 - Interactions simultanées (édition, manipulation...)
 - Changement du repère visuel (rotation des interfaces dans l'orientation de l'utilisateur)
- Aucune librairie intégrant multitouch et interface graphique en python (TouchPy manque de fonctionnalités)
- Flash/AS3 mal supporté sous linux



Pyglet, qu'est ce que c'est ?

- Librairie graphique OO basée sur OpenGL
- Fait pour le développement de jeu ou d'applications
- Fonctionne sous Linux/Windows/MacOSX
- Excellent support des médias (ogg, xvid...), grâce à avbin



Pyglet et les événements

- Pyglet fourni les événements basiques comme :
 - `on_mouse_down`, `on_mouse_up`, `on_mouse_move`
 - `on_key_down`, `on_key_up`, `on_key_press`
- Possibilité d'ajouter facilement des événements à un objet existant



La couche événement PyMT

- Serveur UDP pour l'écoute des messages TUIO envoyés par le tracker (tBeta, reactivision...)
- Transformation des messages en événements :
 - `on_touch_down`, `on_touch_up`, `on_touch_move`
 - `on_object_down`, `on_object_up`, `on_object_move`
- Reconnaissance du double-tap



PyMT, les fonctionnalités Core

- Reconnaissance de geste
- Abstraction OpenGL (FBO, Stencil, Shader, Blending)
- Fonctions pour tracer un rectangle, cercle, ou peindre un trait
- Lecture et affichage de mesh 3D
- Gestion des préférences utilisateurs (plein-écran, temps de détections du double-tap, ip/port d'écoute UDP...)



PyMT, l'interface graphique

- Abstraction des animations, avec fonctions de transitions prédéfinies
- Parseur CSS étendu
- Simulateur d'événements multitouch avec la souris
- Fenêtre multitouch opengl avec fond d'écran inclut



PyMT, les widgets 1/2

- Bouton simple, à 2 états, avec une image, matrice de boutons
- Un afficheur de texte
- Sélecteur de couleurs
- Slider 1D, 2D, matrice, radial
- Reconnaissance de geste
- Fenêtre embarquée, modale, popup
- Formulaire



PyMT, les widgets 2/2

- Layout (Box, Grid)
- Scatter (ZUI)
- SVG
- XML (création d'interface provenant d'un fichier)
- Kinetic abstraction + Kinetic List
- Tabs



Exemple de widget

```
from pymt import *
```

```
class MTEExample(MTWidget):
```

```
    def on_touch_move(self, touches, touchID, x, y):  
        self.pos = x, y
```

```
    def draw(self):
```

```
        set_color(1, 1, 0, .5)
```

```
        drawRectangle(pos=self.pos, size=self.size)
```

```
        set_color(1, 0, 0, .5)
```

```
        drawRectangle(pos=self.pos, size=(20, 20))
```

```
w = MTWindow()
```

```
w.add_widget(MTEExample())
```

```
runTouchApp()
```



Exemple de widget 2

```
from pymt import *
```

```
class MTEExample(MTScatterWidget):
```

```
    def draw(self):
```

```
        set_color(1, 1, 0, .5)
```

```
        drawRectangle(pos=(0,0), size=self.size)
```

```
        set_color(1, 0, 0, .5)
```

```
        drawRectangle(pos=(0,0), size=(20, 20))
```

```
w = MTWindow()
```

```
w.add_widget(MTEExample())
```

```
runTouchApp()
```



Application de photos

```
from pymt import *  
from glob import glob
```

```
w = MTWindow()  
kinetic = MTKinetic()
```

```
for filename in glob('pictures/*.jpg'):  
    kinetic.add_widget(MTScatterImage(filename=filename))
```

```
w.add_widget(kinetic)  
runTouchApp()
```



Utilisation des CSS

```
from pymt import *
mycss = '''
.simple {
  bg-color: #ff5c00;
  border-radius: 20;
  border-radius-precision: .1;
}
'''

css_add_sheet(mycss)

w = MTWindow()
w.add_widget(MTButton(label='coucou', pos=(10, 10)))
w.add_widget(MTButton(label='coucou', cls='simple', pos=(10, 150)))

runTouchApp()
```



Et des travaux en cours

- Animation imbriquée
 - `self.pos = (500, 200)`
 - `self.size = (400, 400)`
 - → déclenche une animation de la position et la taille sur l'ancienne valeur vers la nouvelle
 - → les paramètres de l'animation peuvent être contrôlés au runtime



Et des travaux en cours

- Lecture de vidéo provenant d'OpenCV et de Gstreamer
- Widget de graphique 2D
- Widget Book (visualisation de PDF comme un livre)
- Application de Map Browser basé sur ModestMap
- Portage de Webkit sur Python (sans passer par une plateforme Windows, Gtk, Qt ou Mac.)



Questions ?



- Website: <http://pymt.txzone.net/>
 - 32.7 LOC / ~10 contributeurs
 - 1 étudiant GSOC
 - Partenariat avec l'université d'Iowa, NUIGroup, l'université de Napier (Edimbourg)
-
- Mathieu Virbel <tito@bankiz.org>