Programmation Orientée Objet

TD 4 : Programmation évènementielle & UI

1 Programmation d'UI

On veut créer une fenêtre contenant un champ texte de valeur initiale 0 et un bouton. Lorsque l'on clique sur le bouton, la valeur du texte augmente de 1. Proposez un design de fenêtre permettant de réaliser cette interface. Comment la testeriez-vous ?

1.1 Modèle MVC

Un façon pratique de développer une interface est d'utiliser le modèle MVC (Modèle, Vue, Contrôleur) :

- le **Modèle** régit l'accès aux données
- la **Vue** est tout ce qui est affiché
- **Contrôleur** est ce qui fait le lien entre Vue et Modèle

Proposez un modèle UML en 3 classes (Modèle, Vue et Contrôleur) permettant de réaliser cette application.Comment la testeriez-vous avec des tests unitaires ?

1.1.1 MVC et appjar.info

Le code suivant utilise la bibliothèque `appjar.info` pour coder l'UI. Associez le code ci-dessus au modèle MVC. Où sont les différentes parties ?

```
from appJar import gui

model = {
    "value": 0
}

def press(button):
    model["value"] += 1
    app.setLabel("value", str(model["value"]))
```

```
app = gui()
app.addLabel("value", str(model["value"]), 0, 0)
app.addButton("+1", press, 0, 1)

app.go()
print("c'est␣fini.")
```

1.1.2 Ajout de boutons

Ajoutez à l'application un bouton '-1' qui décrémente la valeur affichée de 1 si cette valeur est > 0.

2 Les dés

Proposez l'UI et le modèle UML d'une fenêtre permettant de lancer un dé. Comment coderiez vous le tout avec `appjar.info` ?