Programmation Orientée Objet

TD 2 : Programmation évènementielle & UI

1 Fonctions et namespaces

1.1 Les fonctions sont des variables comme les autres

```
une_liste = []
truc = une_liste.append
truc("?")

print(une_liste)
```

Dans le code ci-dessus, que représente `append` ? Exécutez-le en montrant toutes les lignes de codes et les *namespaces* utilisés.

1.2 Fonctions de fonction

On souhaite créer une fonction `ajoute` avec un paramètre (entier) `x`. Le retour de cette fonction doit être une fonction à un paramètre `y` qui rend `x + y`

1.2.1 Tests

Comment testeriez vous cette fonction ?

1.2.2 Implémentation

Codez cette fonction et vérifiez (à la main) qu'elle passe bien vos tests en notant toutes les variables et les namespaces rencontrés.

2 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 ?

2.1 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 ?

2.1.1 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.")
```

2.1.2 Ajout de boutons

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

3 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` ?