Cégep Limoilou

Formatif 5

420-SF3-RE

Exercice 1. Affichage d'une fonction

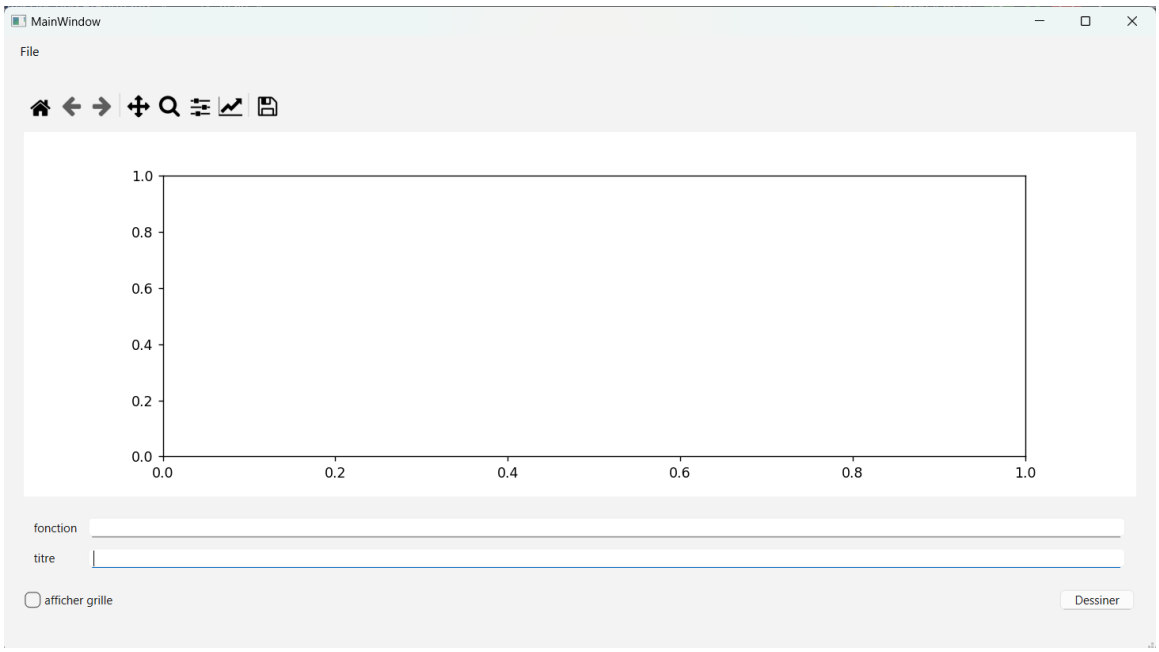
Objectif : Créer une interface graphique qui permet d'afficher le graphe d'une fonction dont on choisit l'expression fonctionnelle dans un QTextEdit. On doit pouvoir choisir si on affiche la grille ou pas, choisir un titre pour le graphique avec un QTextEdit et dans un menu choisir la couleur du graphique. Un message d'erreur doit s'afficher si l'expression fonctionnelles n'est pas valide.

Contrainte : la vue doit clairement être séparée du modèle.

Fichiers : Votre projet doit contenir 5 fichiers :

- main.py, qui contient un main, aussi court que possible
- function_view.py qui contient la classe FonctionView qui représente et gère la fenêtre principale.
- function_model.py, qui contient la classe FonctionModel responsable de stocker les éléments affichés dans la vue (fonction, titre, bool pour affichage de la grille et couleur au minimum)
- mpl_canvas.py qui contient la classe MplCanvas en charge de l'affichage du graphique
- ui/function_view.ui le descripteur uiml généré par QTDesigner.

Voici un aperçu de l'interface que vous devez réaliser :



Cégep Limoilou

Formatif 5

420-SF3-RE

Fonctionnalités demandées :

- Lorsqu'on a choisi la fonction, le titre ou la CheckBox, on doit mettre à jour la classe de modèle pour refléter ce choix.
- Lorsqu'on change la taille de la fenêtre le Canvas doit prendre la place restante. Verticalement les champs textes ne doivent pas changer de taille mais ils doivent le faire horizontalement.
- Lorsqu'on appuie sur le bouton dessiner, on doit récupérer ces données dans le modèle et afficher le graphique dans le canvas, avec grille et titre en fonction du choix de l'utilisateur.
- Le canvas doit être muni de la NavigationToolBar
- Lorsqu'on termine l'édition de la fonction, on doit interroger le modèle par une méthode pour savoir si elle est valide et sinon, afficher un message dans une fenêtre modale (QMessageBox)
- Lorsqu'on sélectionne le menu couleur, un sélecteur de couleur doit apparaitre et la couleur sélectionnée doit être transmise au modèle pour le prochain affichage.

Indications :

- Pour transformer une string en fonction utilisable par matplotlib, on doit d'abord la transformer en fonction sympy :

```
fonction = sp.sympify(f_str)
```

  C'est celle qu'on stocke dans le modèle. Ensuite on doit en faire une fonction numérique de type numpy évaluable en spécifiant la variabsle :

```
f=sp.lambdify(variable, fonction, 'numpy')
```

- Pour valider une expression fonctionnelle, il faut d'abord la transformer en fonction sympy comme ci-dessus. Ensuite on peut limiter les variables à la seule variable autorisée en testant :

```
fonction.free_symbols <= { variable }
```

- Pour transformer une QColor en couleur utilisable par matplotlib on doit construite un 4-uple :

```
color = (qcolor.redF(), qcolor.greenF(), qcolor.blueF(),
qcolor.alphaF())
```

- **Ce formatif doit être réalisé avec soin, il est le point de départ pour un TP2 réussi!**

Cégep Limoilou

Formatif 5

420-SF3-RE

Éléments travaillés :

- Intégration de graphisme matplotlib
- Séparation Vue/Modèle
- Utilisation de boites de dialogue
- Validation de champs
- Gestionnaire de disposition (layouts) et redimensionnement de fenêtres