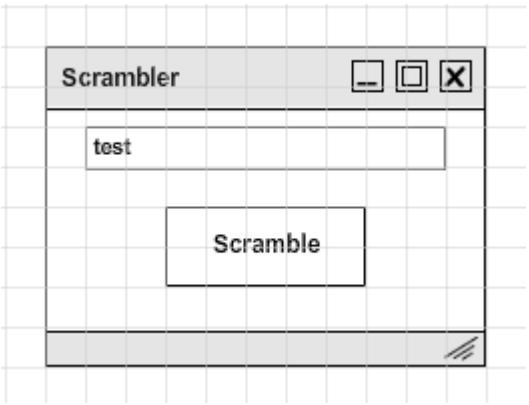
Karel de Grote Hogeschool

SCRAMBLE



We bouwen een eenvoudige app om een tekst door elkaar te gooien. De gebruiker typt in het textveld en wanneer er op Scramble wordt geklikt wordt de tekst door elkaar gehaspeld.

Enkel het model en de wireframe zijn gegeven. **`Main`**, view en **`Presenter`** moeten nog geïmplementeerd worden.

De voornaamste JavaFX klassen die we voor deze oefening nodig hebben zijn:

- **`javafx.scene.layout.GridPane`**
- **`javafx.geometry.Insets`**
- **`javafx.event.ActionEvent`**
- **`javafx.event.EventHandler`**

Raadpleeg in eerste instantie de JavaFX documentatie als je ergens vast zit!

1 WIREFRAME

Bouw het wireframe van deze app na in een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

Vul de **`Main`** klasse en de **`Presenter`** klasse aan volgens de geleerde structuur. De view klasse hoort ook thuis in dit verhaal, maar voorlopig kan je hiervoor de bestaande lege klasse gebruiken genaamd **`ScramblerView`** (in punt 3 werken we deze verder uit).

- De **`Presenter`** klasse:
  - Maak twee attributen: het model en de view
  - Schrijf een constructor om beide attributen van een waarde te voorzien
- De **`Main`** klasse
  - Zorg er voor dat de **`Main`** klasse kan starten.
  - Implementeer de **`start`** methode:
    - Maak een nieuw model-, view- en presenter-object aan
    - Voeg de view toe in een **`Scene`** op de **`Stage`**.
    - Zorg ervoor dat de **`Stage`** de juiste titel krijgt
    - Toon de **`Stage`**

3 UI OPBOUWEN

We werken de klasse **`ScramblerView`** uit.

- Deze klasse erft van **`GridPane`**
- De controls moeten toegevoegd worden als attributen. Raadpleeg de wireframe om te weten welke controls je nodig hebt.

3.1 De methode initialiseNodes

Voeg een methode genaamd **initialiseNodes** toe aan **ScramblerView** en roep deze methode op in de constructor.

In de methode zelf initialiseer je de controls die je als attributen hebt aangemaakt.

3.2 De methode layoutNodes

We plaatsen de controls op de **GridPane** (dit is de klasse **ScramblerView** zelf).

Voeg een methode genaamd **layoutNodes** toe aan **ScramblerView** en roep deze methode op in de constructor.

In de methode **layoutNodes** zelf:

- Maak gebruik van de **add** methode van de klasse **GridPane** om de controls op de juiste plaats te krijgen.
- Zorg voor een horizontale en verticale “gap” van 10 pixels tussen de cellen van de **GridPane**. Zorg ook voor een “padding” van 10 pixels.

Zoek in de documentatie van **GridPane** naar de juiste methodes!

4 AFHANDELEN EVENTS

We zijn geïnteresseerd in het action event op de knop. Implementatie van de eventhandlers gebeurt in de **Presenter** klasse.

4.1 De methode handleEvents

Voeg de methode **handleEvents** toe aan de klasse **Presenter** en roep ze op in de constructor. Vul de methode in als volgt:

- Hang een eventhandler aan de knop. Om dit te kunnen doen moet je er voor zorgen dat de button bereikbaar is via een getter die je aan de klasse **ScramblerView** toevoegt (package-private).
- Zorg er voor dat de eventhandler de methode **updateView** (zie punt 4.2) oproept.

4.2 De methode updateView

Voeg de methode **updateView** toe aan de klasse **Presenter**. Deze methode doet het volgende:

- Je laat het model een bewerking uitvoeren door de juiste methode van de klasse **Scrambler** op te roepen.
- Je haalt de nodige informatie op uit het model en toont deze in het tekstveld. Je moet er voor zorgen dat het tekstveld, net zoals de knop, bereikbaar is via een getter die je aan **ScramblerView** toevoegt.