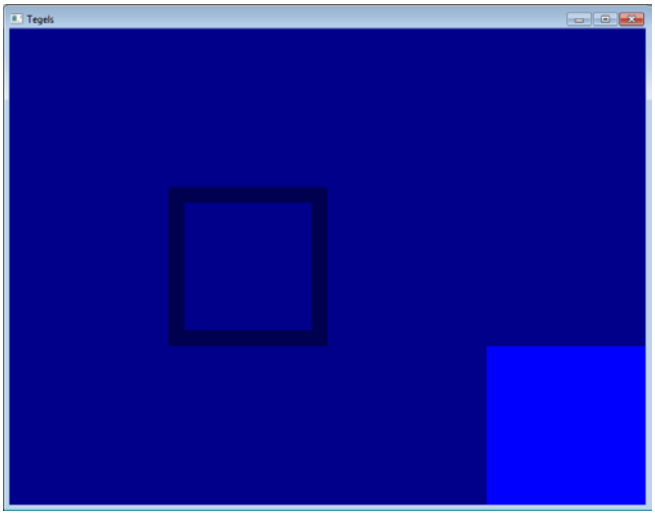
Karel de Grote Hogeschool

TEGELS



In deze oefening implementeren we een veld of speelbord van 4x3 tegels waarvan er maximaal één geselecteerd kan worden. Als de gebruiker met de muis over een tegel beweegt dan geven we dit grafisch weer.

De view en het model zijn gegeven, je hoeft dus enkel de presenter te implementeren. Na enkele lesweken kan je terugkeren naar deze oefening om de view te leren begrijpen en eventueel te gebruiken als voorbeeld.

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

- `javafx.event.EventHandler`
- `javafx.scene.input.MouseEvent`

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

1 WIREFRAME

De wireframe die bij deze oefening hoort is triviaal. Je mag deze als oefening bouwen met behulp van een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

De *model* klasse is **`TileModel`**. Het aantal rijen en kolommen zijn hierin vastgelegd. Het model houdt bij welke tegel momenteel geselecteerd is. Voor deze oefening hoef je het model niet aan te passen!

De *view* klasse is **`TileView`**. Er wordt gewerkt met een **`BorderPane`** en een **`Canvas`** (zie één van de volgende lesweken). Voor deze oefening hoef je de view niet aan te passen!

De *presenter* klasse is **`Presenter`**. Deze klasse dien je volledig uit te werken in punt 4.

De **`Main`** klasse is gegeven.

3 UI OPBOUWEN

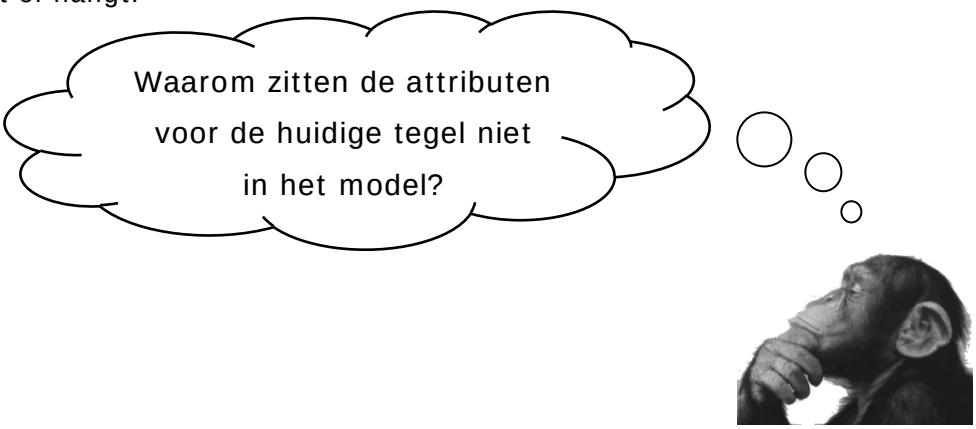
De view is reeds uitgewerkt voor deze oefening.

4 AFHANDELEN EVENTS

De **`Presenter`** klasse zorgt voor de afhandeling van events. Telkens de gebruiker met de muis beweegt willen we kunnen zien over welke tegel hij zich bevindt. Telkens de gebruiker met de muis klikt willen we de juiste tegel selecteren of de-selecteren.

- Zorg voor twee attributen: een attribuut voor het model en een attribuut voor de view.

- We hebben twee extra attributen nodig die bijhouden over welke tegel (**rij** + **kolom**) de muis beweegt of hangt.



- Implementeer de constructor: zorg er voor dat het model en de view met behulp van parameters een waarde krijgen. De twee andere attributen (rij + kolom van de huidige tegel) initialiseer je beiden als "**-1**", wat betekent dat de muis momenteel niet boven een tegel beweegt of hangt.

4.1 De methode addEventHandlers

Voeg de methode **addEventHandlers** toe aan **Presenter** en zorg er voor dat ze opgeroepen wordt in de constructor.

In de methode **addEventHandlers**:

- Vraag het **Canvas** op uit de view (**getCanvas**) en hang **drie** event handlers aan deze control:
  - `setOnMouseMoved` wordt opgeroepen wanneer de gebruiker over het **Canvas** beweegt met de muis.
    - We gaan aan de slag met het `MouseEvent` object (de parameter van onze event handler) en gebruiken de methodes `getX` en `getY` om te achterhalen waar de muis zich bevindt op het **Canvas**.
      - Als we de X-waarde delen door het aantal kolommen (**TileModel.COLUMNS**) dan krijgen we een getal van **nul** tot **aantal kolommen – 1**… dit getal kunnen we gebruiken als kolom-index!
      - Als we de Y-waarde delen door het aantal rijen (**TileModel.ROWS**) dan krijgen we een getal van **nul** tot **aantal rijen – 1**… dit getal kunnen we gebruiken als rij-index!

      Je hebt nu de kolom-index en de rij-index bepaald. Gebruikt deze om de attributen die de huidige rij en kolom voorstellen aan te passen.
    - Roep de methode **updateView** op (zie 4.2).
  - `setOnMouseClicked` wordt opgeroepen wanneer de gebruiker op het **Canvas** klikt met de muis.
    - Ga opnieuw aan de slag met het **MouseEvent**. Gebruik dit keer de berekende kolom-index en rij-index om de geselecteerde tegel in het model vast te leggen (**setSelectedTile**).

      Je hebt dezelfde berekeningen gedaan als in de vorige event handler. Pas je klasse aan zodat de code voor de berekeningen hergebruikt wordt!
    - Roep de methode **updateView** op (zie 4.2).

- `setOnMouseExited` wordt opgeroepen wanneer de gebruiker het **Canvas** verlaat met de muis:
  - De attributen voor de huidige rij en kolom worden beiden op "**-1**" gezet.
  - Roep de methode **updateView** op (zie 4.2).

4.2 De methode updateView

Voeg de methode **updateView** toe aan **Presenter** en zorg er voor dat ze opgeroepen wordt in de constructor.

In de methode **updateView**:

- Roep de methode **displayCurrentTiles** op van de view. Deze methode verwacht vier parameters:
  - Kolom waar de muis zich momenteel bevindt. **attribuut**
  - Rij waar de muis zich momenteel bevindt. **attribuut**
  - Kolom van de geselecteerde tegel. Opvragen uit het **model**.
  - Rij van de geselecteerde tegel. Opvragen uit het **model**.