

TIJD

In deze oefening simuleren we de opkomende en ondergaande zon. Met behulp van een slider, die de tijd voorstelt, wisselen we tussen dag en nacht.

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.

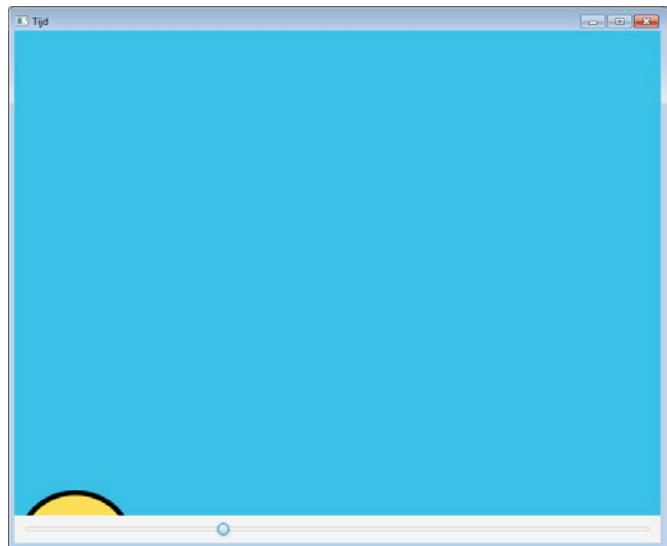
De voornaamste JavaFX klassen die we voor deze oefening nodig hebben zijn:

- [`javafx.scene.control.Slider`](#)
- [`javafx.event.EventHandler`](#)
- [`javafx.scene.input.MouseEvent`](#)

Daarnaast gebruiken we ook:

- [`java.time.LocalDateTime`](#)

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 **TimeModel**. Je kan de tijd instellen en opvragen met behulp van respectievelijk een setter en een getter. Er wordt gebruik gemaakt van de klasse **LocalTime**. Voor deze oefening mag je het model niet aanpassen!

De *view* klasse is **TimeView**. Er wordt gewerkt met een **BorderPane**, een **Slider** 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 de slider verplaatst dient het model, en bijgevolg ook de view, aangepast te worden.

- Zorg voor twee attributen: een attribuut voor het model en een attribuut voor de view.
- Implementeer de constructor: zorg er voor dat beide attributen van een waarde worden voorzien. Hiervoor gebruiken we twee parameters.

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`:

- Hang een event handler aan de slider die je ophaalt uit de view (`getSlider`).
- We willen dat onze event handler opgeroepen wordt wanneer de gebruiker met de muis de slider versleept, dus we werken met `setOnMouseDragged` en **niet** met `setOnAction`!
- De event handler zelf doet het volgende:
 1. Haalt de huidige waarde van de slider op (`getValue`). De huidige waarde wordt teruggegeven als een kommagetal dat we als volgt interpreteren:
 - Gedeelte vóór de komma zijn **uren**
 - Gedeelte na de komma zijn **minuten**.
 - Bijvoorbeeld:
9.5 → 9 uur 30 minuten
13.1 → 13 uur 6 minuten
16.9 → 16 uur 54 minutenSchrijf zelf de nodige code om de omzetting te doen.
 2. Stelt een nieuwe tijd in bij het model. Gebruik `LocalTime.of` om de waarde te bepalen van de parameter die je aan de setter meegeeft.
 3. Roept de methode `updateView` op (zie punt 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 `applyDaylightSun` op van de view. Deze methode verwacht drie parameters die je uit het model kan ophalen. Roep de bestaande methodes op van het model: `getDaylightPercentage`, `getSunHeight` en `getSunPositionX`.
- Vraag de huidige tijd op uit het model en geef de slider de juiste waarde (`setValue`). We gebruiken dezelfde regeld als in 4.1:
 - Bijvoorbeeld:
8 uur 45 minuten → 8.75
10 uur 12 minuten → 10.2
12 uur 33 minuten → 12.55Schrijf zelf de nodige code om de omzetting te doen.