

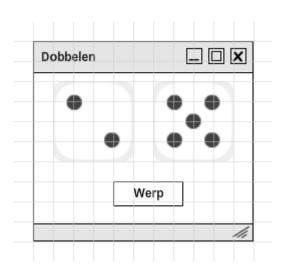
DOBBELEN

We bouwen een eenvoudige dobbel-app. Hiernaast staat de wireframe van de applicatie. Deze is gemaakt met de online tool mockflow.com.

De model-klasse (Dobbelen.java) is al klaar, de Main klasse ook. We moeten enkel de *view* en de *presenter* nog uitwerken.

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

- javafx.scene.layout.GridPane
- javafx.scene.image.Image
- javafx.scene.image.ImageView
- javafx.geometry.Insets



Figuur 1: wireframe (mockflow.com)

Raadpleeg in eerste instantie de <u>JavaFX documentatie</u> als je ergens vast zit!

1 WIREFRAME

Bouw de gegeven wireframe na in een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

De Main klasse is al aangemaakt. Bestudeer de code. In de start-methode worden het model en de view via een presenter gekoppeld.

- Voeg de view toe in een **Scene** op de **primaryStage**.
- Zorg ervoor dat de app als titel "Dobbelen" krijgt
- Toon de primaryStage

3 UI OPBOUWEN

Hier werken we de klasse DobbelenView uit.

We hebben drie controls nodig: twee **ImageView** controls en een **Button** control. Maak deze controls aan als attributen.

3.1 De methode initialiseNodes

Schrijf een nieuwe methode genaamd **initialiseNodes**. Roep deze methode op in de constructor.

In deze methode:

- Voorzie de drie attributen van een waarde.
- De knop moet een geprefereerde breedte van 80 pixels hebben.



3.2 De methode layoutNodes

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

Schrijf een nieuwe methode genaamd layoutNodes. Roep deze methode op in de constructor.

In de methode layoutNodes:

- Maak gebruik van de **add** methode van de klasse **GridPane** om de twee images en de knop op de juiste plaats te krijgen. De knop overspant twee kolommen.
- 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

De private methode handleEvents is aangemaakt in de klasse Presenter en wordt aangeroepen via de constructor. Je hoeft deze methode enkel nog in te vullen.

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

4.2 De methode updateView

Deze methode moet de juiste informatie uit het model ophalen en vervolgens de view up-todate brengen met deze informatie.

- Je haalt het aantal ogen op voor beide teerlingen. Gebruik de juiste methode van de klasse **Dobbelen**.
- Je toont de overeenkomstige images in de imageviews in de **DobbelenView** klasse. Je moet er voor zorgen dat de imageviews, net zoals de knop, bereikbaar zijn via getters die je aan **DobbelenView** toevoegt.
- Tip: een Image tonen in een ImageView doe je als volgt: imageView.setImage(new Image("be\kdg\dobbelen\view\images\die1.png"));