

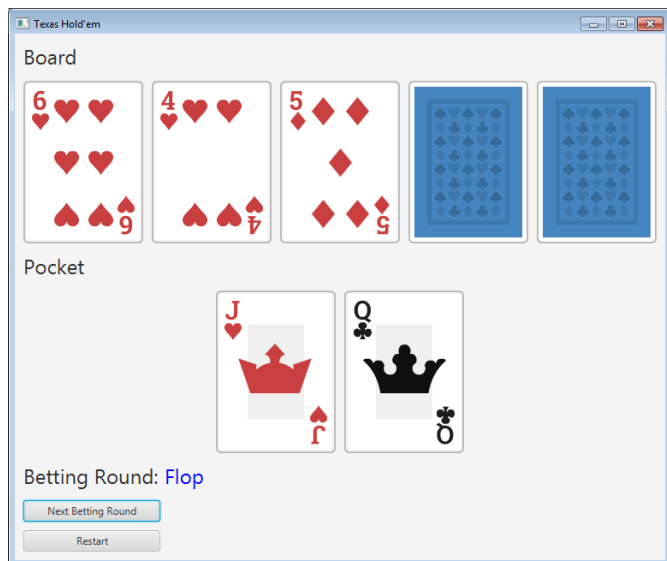
TEXAS HOLD'EM

In deze oefening maken we een applicatie om het verloop van een rondje [Texas Hold'em](#) te simuleren zonder in te gaan op inzet en dergelijke. We doen dit vanuit het oogpunt van één speler die twee kaarten in de hand heeft (*pocket*). Er verschijnen in totaal vijf kaarten op tafel (*board*): eerst drie tegelijk, dan één en tenslotte nog één.

Het doel van deze oefening is om te oefenen met **HBox** en **VBox** en deze layouts in elkaar te “nesten”.

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

- [javafx.scene.layout.HBox](#)
- [javafx.scene.layout.VBox](#)
- [javafx.scene.image.Image](#)
- [javafx.scene.image.ImageView](#)
- [javafx.scene.paint.Color](#)
- [javafx.geometry.Pos](#)



Raadpleeg in eerste instantie de [JavaFX documentatie](#) als je ergens vast zit!

1 WIREFRAME

Bouw de gegeven wireframe na in een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

De *model* klasse is **Game**. Een **Game** heeft o.a. een boek kaarten (**Deck / Card**) en een speler (**Player**). Het model hoeft je niet aan te passen voor deze oefening.

De belangrijkste *view* klasse is **GamePane**. Deze klasse werk je helemaal uit in punt 3. De view-klassen **CardImages** en **ViewConstants** zijn gegeven en hoeft je niet aan te passen. Deze klassen helpen je op weg om de juiste images en fonts te kunnen gebruiken.

De *presenter* klasse is **Presenter**. Deze klasse is gegeven en hoeft je niet aan te passen.

De **Main** klasse is eveneens gegeven.

3 UI OPBOUWEN

De klasse **GamePane** erft over van **VBox**. In totaal bevat deze view klasse **dertien** controls. Maak een attribuut aan voor elke control:

- Een **Label** dat we later de tekst “Board” zullen geven.
- **Vijf ImageView** controls; één voor elke kaart die op tafel ligt. Hiervoor kan je een array gebruiken.

- Een **Label** dat we later de tekst "Pocket" zullen geven.
- **Twee ImageView** controls; één voor elke kaart van de speler. Hiervoor kan je eveneens een array gebruiken.
- Een **Label** dat we later de tekst "Betting Round" zullen geven.
- Een **Label** waarvan de tekst regelmatig aangepast wordt (bvb. "Flop").
- Twee **Button** controls:
 - Een knop om naar de volgende fase te gaan.
 - Een knop om een nieuwe spelronde te beginnen.

Vul volgende *package-private* getters en setters aan:

- **getNextPhase**: Hier geef je de knop terug die het spel naar de volgende fase brengt.
- **getRestart**: Hier geef je de knop terug die het spel herstart.
- **getBrValueLabel**: Hier geef je het **Label** terug waarvan de tekst kan veranderen (het blauwe Label in de screenshot)
- **setBoardImage**: Deze methode heeft twee parameters:
 - **index**: een getal van **nul** tot **vier**.
 - **cardImage**: een **Image** die getoond moet worden op één van de vijf **ImageViews** onder "Board".

Implementeer deze methode zodat de juiste van de vijf **ImageViews** een aangepaste **Image** krijgt.

1. Met behulp van **index** bepaal je de juiste **ImageView** control.
2. Gebruik de methode **setImage** van de klasse **ImageView** om **cardImage** op de **ImageView** te plaatsen.

- **setPocketImage**: Deze methode heeft twee parameters:
 - **index**: een getal van **nul** tot **één**.
 - **cardImage**: een **Image** die getoond moet worden op één van de twee **ImageViews** onder "Pocket".

Implementeer deze methode zodat de juiste van de twee **ImageViews** een aangepaste **Image** krijgt.

1. Met behulp van **index** bepaal je de juiste **ImageView** control.
2. Gebruik de methode **setImage** (van de klasse **ImageView**) om **cardImage** op de **ImageView** te plaatsen.

3.1 De methode **initialiseNodes**

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

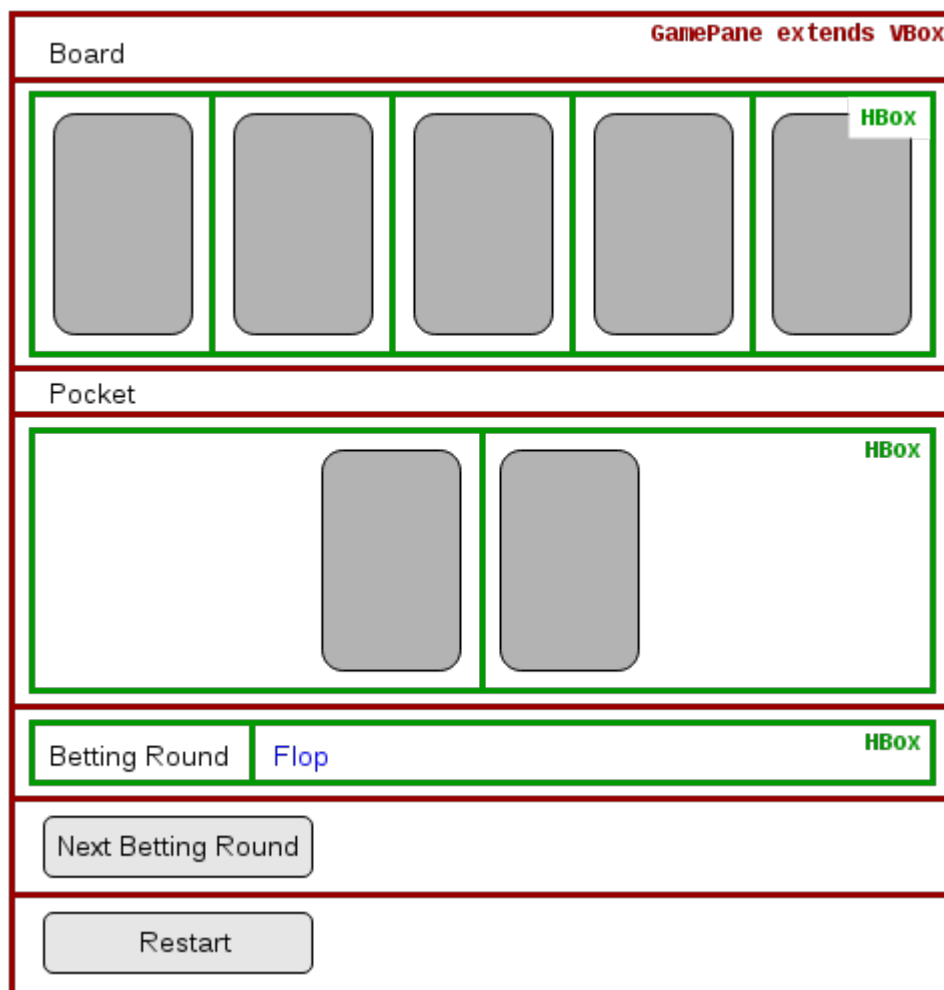
In de methode **initialiseNodes**:

- Geef elk attribuut van deze klasse een waarde:
 - Drie van de vier **Labels** kan je met een vaste **string** initialiseren (raadpleeg de screenshot of je wireframe). Het vierde **Label** mag je als een leeg **Label** initialiseren.
 - De twee knoppen initialiseer je zodat ze de juiste tekst bevatten (raadpleeg de screenshot of je wireframe).
 - De zeven **ImageView** controls initialiseer je als **lege ImageView**s. Hiervoor gebruik je de **constructor zonder parameters**.

3.2 De methode `layoutNodes`

In onderstaand schema kan je de algemene layout zien van **GamePane**. **GamePane** is een **VBox** en bevat zeven cellen die vertikaal onder elkaar staan. De **VBox** is aangegeven in een donkerrode kleur.

Als we in één van de cellen componenten **naast** elkaar willen plaatsen dan moeten we in de betreffende cel gebruik maken van een **HBox**. Elke **HBox** is aangegeven in een groene kleur. We vinden een **HBox** terug in cel 2, 4 en 5 van de **VBox**.



Voeg de methode `layoutNodes` toe aan **GamePane** en zorg er voor dat ze opgeroepen wordt in de constructor.

In de methode `layoutNodes`:

- We zetten de *padding* van de **VBox** op `ViewConstants.INSETS`. Gebruik de methode `setPadding`.
- We zetten de *spacing* van de **VBox** op `ViewConstants.SPACING`. Gebruik de methode `setSpacing`.
- We zetten de *alignment* van de **VBox** op `Pos.CENTER_LEFT`. Gebruik de methode `setAlignment`.

- Zet het font van **elk Label** op `ViewConstants.LARGE_LABEL_FONT`. Gebruik de methode `setFont`.
- Zet de kleur van het variërende **Label** (naast "Betting Round") op `Color.BLUE`. Gebruik de methode `setTextFill`.
- Zet de minimumbreedte van beide knoppen op `160`.
- Maak een nieuwe **HBox** aan die de bovenste rij van vijf kaarten zal voorstellen. Let op, het gaat hier over een lokale variabele, geen attribuut!
 - Zet de *spacing* van je **HBox** op `ViewConstants.SPACING`.
 - Voeg de vijf **ImageViews in volgorde** toe aan deze **HBox**. Een component toevoegen doe je als volgt:


```
mijnHBox.getChildren().add(mijnImageView);
```
- Maak een nieuwe **HBox** aan die de rij van twee kaarten zal voorstellen. Het gaat hier opnieuw over een lokale variabele, geen attribuut.
 - Zet de *spacing* van deze **HBox** op `ViewConstants.SPACING`.
 - Zet de *alignment* van deze **HBox** op `Pos.CENTER`.
 - Voeg de twee **ImageViews in volgorde** toe aan deze **HBox**.
- Maak een nieuwe **HBox** aan die de twee **Labels** zal bevatten. We werken opnieuw met een lokale variabele.
 - Voeg de twee **Labels in volgorde** toe aan deze **HBox**. Eerst het **Label** met de tekst "Betting Round" en vervolgens het blauwe **Label**.
- Voeg volgende componenten **in volgorde** toe aan de **VBox**:
 - Het eerste **Label** met de tekst "Board"
 - De eerste **HBox** met de vijf **ImageViews**
 - Het tweede **Label** met de tekst "Pocket"
 - De tweede **HBox** met de twee **ImageViews**
 - De derde **HBox** die de twee **Labels** bevat
 - De eerste knop
 - De tweede knop

4 AFHANDELEN EVENTS

De presenter is reeds uitgewerkt voor deze oefening.