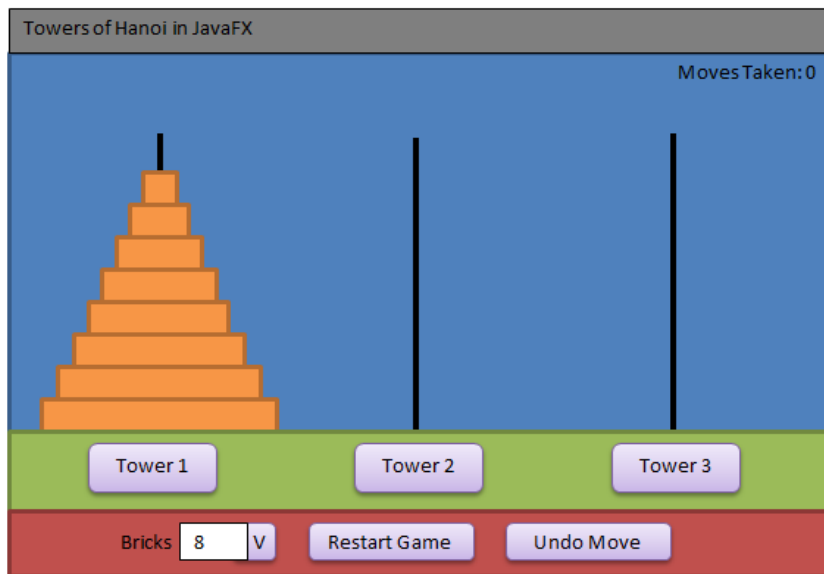


### “Create a Towers of Hanoi game in JavaFX.”

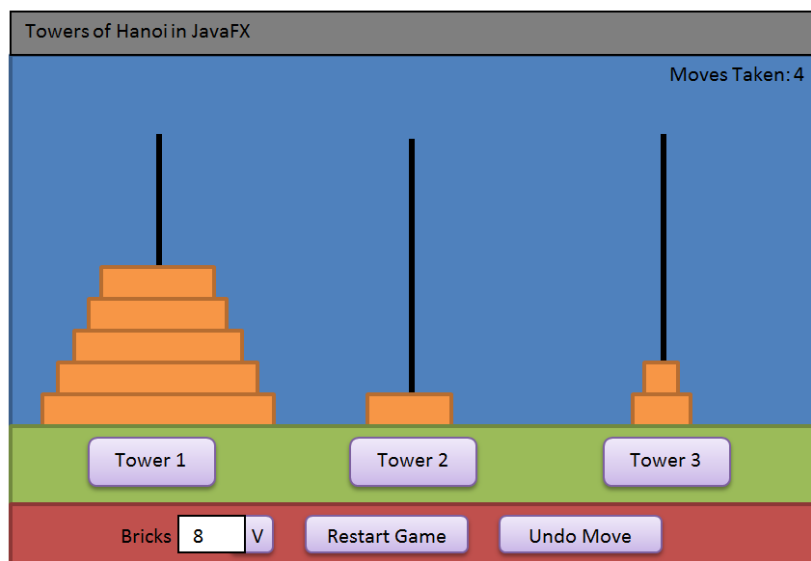
The objective of this CA exercise is to implement an event-driven version of the classic Towers of Hanoi game in JavaFX.

The following is a mock-up of the user interface you need to develop:



The Towers of Hanoi game involves 3 towers. At the start of the game, Tower 1 contains a number of bricks stacked so that smaller bricks are always on top of larger bricks. The goal is to move all bricks to Tower 2 in the fewest number of moves. However, only the top brick can be moved at a time from any tower, and a brick can never be placed on top of a smaller brick.

For instance, from the starting position above, the moves 1-2, 1-3, 2-3, 1-2 would result in this:



The FXML-based interface should use the general layout shown (location of controls etc.), but you should use CSS styling extensively to improve the aesthetics as much as you can (e.g. perhaps use a sky image for the background, grass image for the grass, stone image or gradient for the bricks, drop shadow for the move counter, etc. – exactly what you do is up to you, but showcase your ability to use CSS in JavaFX).

The game should start with a default number of bricks (your choice) between 3 and 8. The number of bricks used should be selectable (minimum 3, maximum 8) in the bottom left area using a combo box, choice box, or text field (your choice). The game restarts if the number of bricks changes.

Bricks can be moved from tower to tower in 3 different ways:

- Using Buttons. Each tower has a button directly beneath it. Users can click the button for the tower to move from, followed by the button for the tower to move to. Some form of colour indication should indicate that a tower is selected (e.g. top brick changes colour/style). Users can deselect a tower by pressing the same button again to abort a move.
- Using Mouse Clicks. Users can alternatively select the tower to move from and to by clicking on them (the towers/tower areas) in sequence. As above, some indication of tower selection, ability to deselect, etc. is required.
- Dragging. The final way of moving bricks is by users dragging them from one tower to another. Only the top brick should be draggable on any tower. Some form of suitable indication of the drag in progress (e.g. moving brick, ghost brick etc.) should be used.

The game should also provide:

- A button to restart the game (using the same number of bricks).
- A button to undo the last move. The undo button could also be repeatedly pressed to undo the last sequence of moves e.g. click 4 times will undo 4 moves.
- A move counter in the top right of the game window (note that undo decrements the counter).
- Some indication of when the game is over (i.e. all bricks on Tower 2) and an ability to restart at that point.

---

Notes:

- You must use JavaFX in your implementation.
  - It is recommended that you use FXML and CSS wherever possible (use Scene Builder etc.)
  - The layout of the interface must be as shown in the mock up, but I want you to stylise things using CSS as much as you can. Create a nice slick interface that shows off what JavaFX lets us do!
  - You cannot provide any other features in the implementation except those asked for.
  - You must develop the game by yourself and be able to explain all code when you demonstrate it to me. You do not need to get everything working in order to pass! (see marking scheme)
  - It is recommended that you concentrate on getting basic button-based moves working before you consider dragging bricks, etc.
-

Indicative Marking Scheme:

- General look, styling behaviour and aesthetics (20%)
- Ability to make legal moves (in some fashion) (15%)
- Ability to make legal moves using tower buttons (10%)
- Ability to make legal moves by clicking on the towers (10%)
- Ability to make legal moves by dragging bricks (15%)
- Ability to select number of bricks to use in game using combo box/choice box/text field (10%)
- Ability to restart game using button (5%)
- Ability to undo single move (5%)
- Ability to undo multiple moves (5%)
- Move counter (5%)