jClarity

CON6265 - Visualising GC with JavaFX

Ben Evans (@kittylyst)
James Gough (@javajimlondon)

Who are these guys anyway?

CRISIS
Fighting for hope for homeless people

LJC

The Well-Grounded
Java
Developer
Covers Java 7

Benjamin J. Evans
Martijn Verburg

MANNING

Morgan Stanley

Beginnings

- This story, as with so many others, starts with beer...

Beginnings

- **It was late at night**
  - We were talking about what makes the JVM special
- **The Java platform has a fundamental design principle**
- **Use runtime information to influence dynamic management of running Java / JVM processes**

- **Examples**
  - Young generational hypothesis underlies GC
  - Platform "ergonomics" / dynamic heap sizing

Beginnings

- So we decided to make some teaching tools to illustrate
- GC was the first one we tackled
- We planned to have several to show you
- But GC acquired more complexity
  - So we decided to just do this one
  - But also try out some of our other ideas
  - There's still **plenty** to talk about!
- If you want to grab the code & have a play
- https://github.com/kittylyst/jfx-mem

Flash?

- Original version was a Flash demo
- Let's take a look...
- Thanks to Anna Barraclough for the Flash version
  - & her work on the L & F for the JavaFX version

Why convert to JavaFX?

- **Flash version is “dumb”**
  - Just an animation
  - Can’t be changed easily
  - Has no “awareness” of what it’s doing
- **JavaFX is another technology for developing UIs**
  - But Java!
  - Bundled with J7u6 and up
- **Java-based version could be more flexible**
  - Actually model memory
  - Have a cut-down version of GC algorithms

JavaFX Design Goals

- **Actually model GC & have proper domain model**
  - Memory blocks, areas & allocating threads are modelled
- **Keep a clean UI / code split**
  - Allow headless unit testing
- **Use an interpreter model**
  - Enables multiple implementations of the sources of memory operations
- **Showcase JavaFX features**
- **Clean, well-documented code**
  - This is for teaching purposes

Code Introduction

- **Some examples from building JavaFX Memory Visualizer**
- **Teaching tool for JavaFX**
  - Explain our explorations with it
- **Teaching tool for the Java Memory Model**
  - Model the parallel collectors

FXML

- **Declare layouts in FXML (you can do this via builders in code too)**
- **Example**

```
<VBox xmlns:fx="http://javafx.com/fxml"
fx:controller="com.jclarity.anim.memory.MemoryController">
    <Label text="Memory Demo" fx:id="applicationtitle" />
    <HBox>
        <ComboBox  fx:id="resourceType">
            <items>
                <FXCollections
fx:factory="observableArrayList">
                    <String fx:value="File" />
                </FXCollections>
            </items>
        </ComboBox>
        <Label text="Path:" />
        <TextField fx:id="resourcePath" />
    </HBox>
```

What about Our IDs?

- **Controller class**

```
public class MemoryController implements Initializable
{

@FXML
private TextField resourcePath;

@FXML
private ComboBox resourceType;

@FXML
private void beginSimulation()
```

```
<Button text="Begin" onAction="#beginSimulation" fx:id="beginButton" />
```

How is FXML invoked?

```
     @Override
 public void start(Stage stage) throws Exception {
     Parent root = 
FXMLLoader.load(getClass().getResource("MemoryMainView.fxml"));
     Scene scene = new Scene(root, 600, 500);

scene.getStylesheets().add(getClass().getResource("Memory.css").
toExternalForm());
     stage.setScene(scene);
     stage.setTitle("JavaFX Memory Visualizer");
     stage.show();
 }
```

Custom Component (Simple Memory Block)

```
public class MemoryBlockView extends StackPane {

    private Rectangle box;

    private Text text;

    private ObjectProperty<MemoryStatus> memoryStatus;
```

- Bindings
  - Recognise change without array of listeners and anonymous inner classes everywhere

  - Sample to follow

Bindings in Constructor

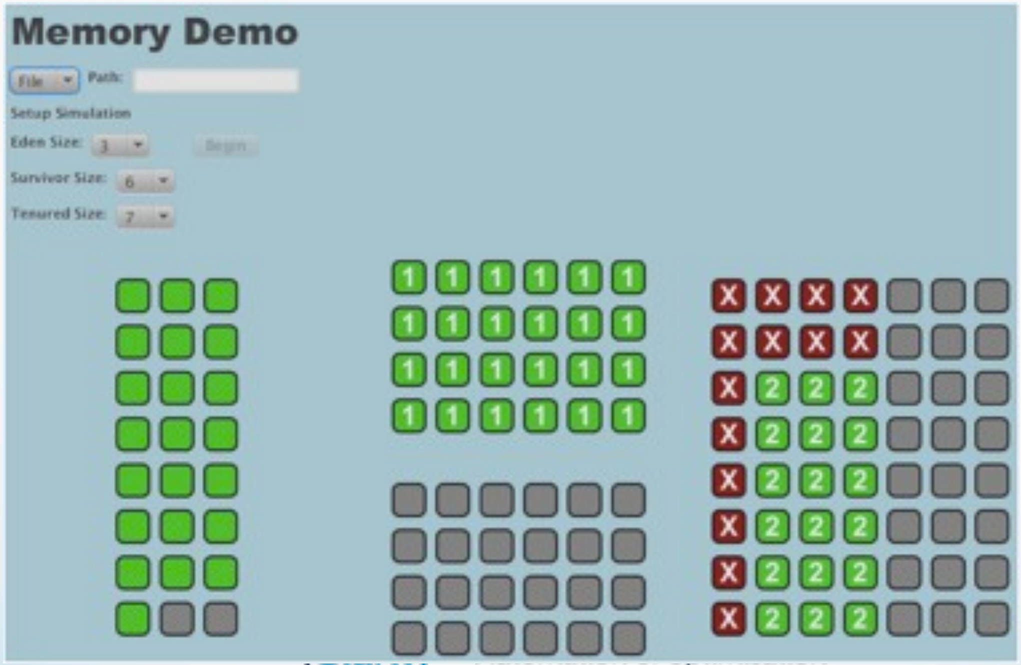
```
box.styleProperty().bind(Bindings.when(memoryStatus.isE
qualTo(MemoryStatus.FREE))

                .then("-fx-fill: gray")


.otherwise(Bindings.when(memoryStatus.isEqualTo(MemoryS
tatus.ALLOCATED))

                .then("-fx-fill: limegreen")


.otherwise(Bindings.when(memoryStatus.isEqualTo(MemoryS
tatus.DEAD))

                .then("-fx-fill: darkred")

                .otherwise("")

                .concat(";"))));
```

Constructor Continued

```
public MemoryBlockView() {

    super(); // Bindings

    box.setStrokeType(StrokeType.INSIDE);

    box.setStroke(Color.web("black"));

    box.setStrokeWidth(2);

    box.setArcWidth(15);

    box.setArcHeight(15);

    text = new Text("");

    text.setFont(Font.font("Arial", FontWeight.BOLD, 24));

    text.setFill(Color.WHITE);

    getChildren().addAll(box, text);

}
```

Where are we now?

- We have some blocks that can change
- We have some states that can be set



Memory Demo

File Path:

Setup Simulation

Eden Size: 3 Begin

Survivor Size: 6

Tenured Size: 7

The Controller

```
initialiseMemoryView(model.getEden(), edenGridPane);


private void initialiseMemoryView(MemoryPool pool,
GridPane gridPane) {

        for (int i = 0; i < pool.width(); i++) {

            for (int j = 0; j < pool.height(); j++) {

                MemoryBlockView block = pool.get(i, j);


gridPane.add(PaneBuilder.create().children(block).build(),
i, j);

            }

        }

    }
```

jClarity Automation of optimisation

Background Thread

```
private final ExecutorService srv =
Executors.newScheduledThreadPool(2);



begin() {

....

AllocatingThread at0 = new
AllocatingThread(memoryInterpreter, model);

srv.submit(at0);

}
```

Platform Run Later - Animation

```
Platform.runLater(new CustomMemoryBlockViewTransition(this));



class CustomMemoryBlockViewTransition implements Runnable {

    private final MemoryBlockView view;

    @Override

    public void run() {

             FadeTransition fadeOldBlockOut = new
 FadeTransition(Duration.millis(10), view);

            fadeOldBlockOut.setFromValue(1.0);

            fadeOldBlockOut.setToValue(0.0);

            fadeOldBlockOut.setCycleCount(1);

            fadeOldBlockOut.setAutoReverse(false);

            fadeOldBlockOut.play();
```

Memory Model and Demo

- Jump into the wilds of the code
- Time for a demo
- https://github.com/kittylyst/jfx-mem
- Spot any bugs, submit a pull request :-)

Thanks

- Anna Barraclough
- Stephen Chin
- FJ van Wingerde
- Otter photo owned by Flickr user moff
  - Reused under a CC Attribution license

jClarity

Thank You

Ben Evans (@kittylyst)
James Gough (@javajimlondon)