Migrate Power of Desktop Applications to the Browser

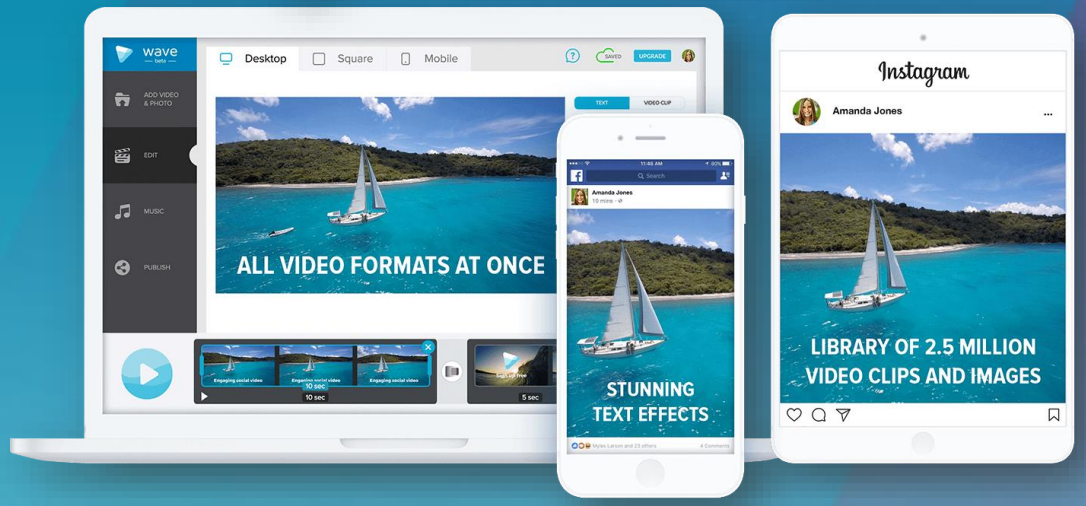
by Iaroslav Kobyliukh

ANIMATRON

Who we are?

ANIMATRON funded by JET BRAINS

wave.video



Incredibly easy to use online video maker for creating marketing and social videos in minutes

STUDIO

Universal animation maker for small businesses and agencies

ANIMATRON

What we do?

- Tween based vector animation editor
- Rich vector tools
- Import (images, audio, video, SVG)
- Export (HTML5, GIF, Video, SVG+SMIL)
- Share (Facebook, Twitter, YouTube, Dropbox, etc.)
- Collaborative editing

ANIMATRON

What is Rich Web Application?

Rich Web application *(or rich Internet application (RIA))* is a Web application designed to deliver the same features and functions normally associated with desktop applications.

Google Docs

SketchUp

TINKERCAD

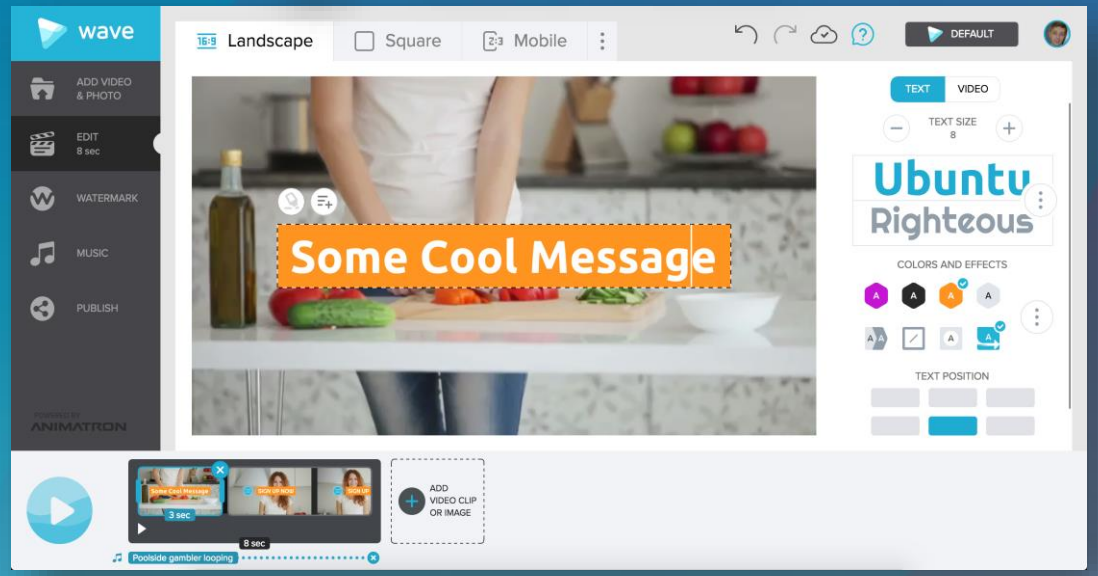
AUTODESK® TINKERCAD®

AUTODESK® PIXLR®

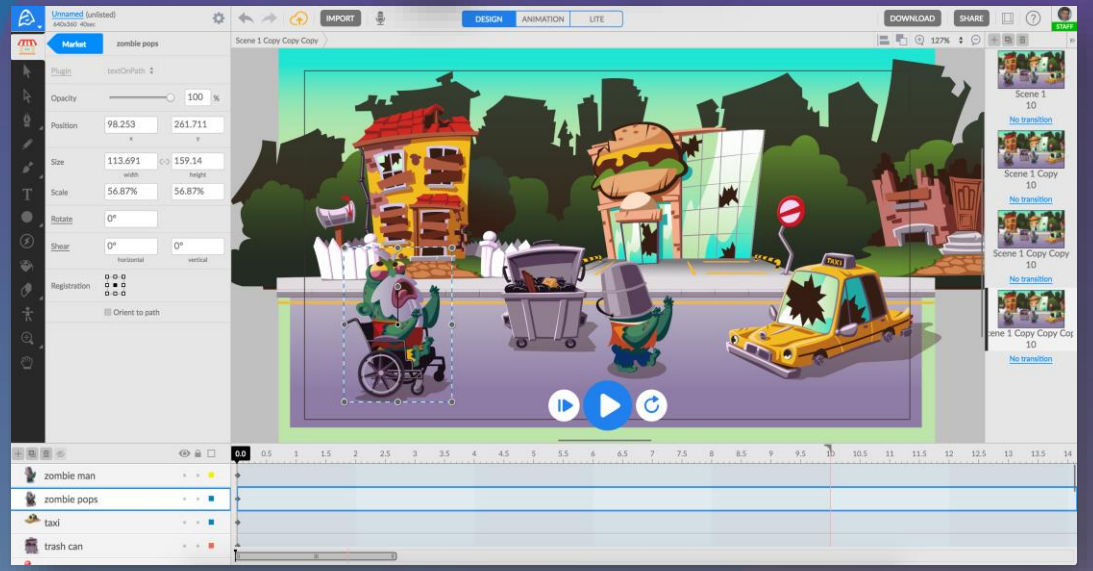
ANIMATRON

Rich Web Applications: Editors

wave.video



STUDIO



ANIMATRON

Rich Web Applications: Principles

- Pages should render fast
- Act immediately on user input
- React to data changes
- Control the data exchange with the server
- Listen internet connection
- Predict behavior

ANIMATRON

Rich Web Application: Technologies

GWT

ANGULAR

vaadin

Microsoft

ASP.NET

React

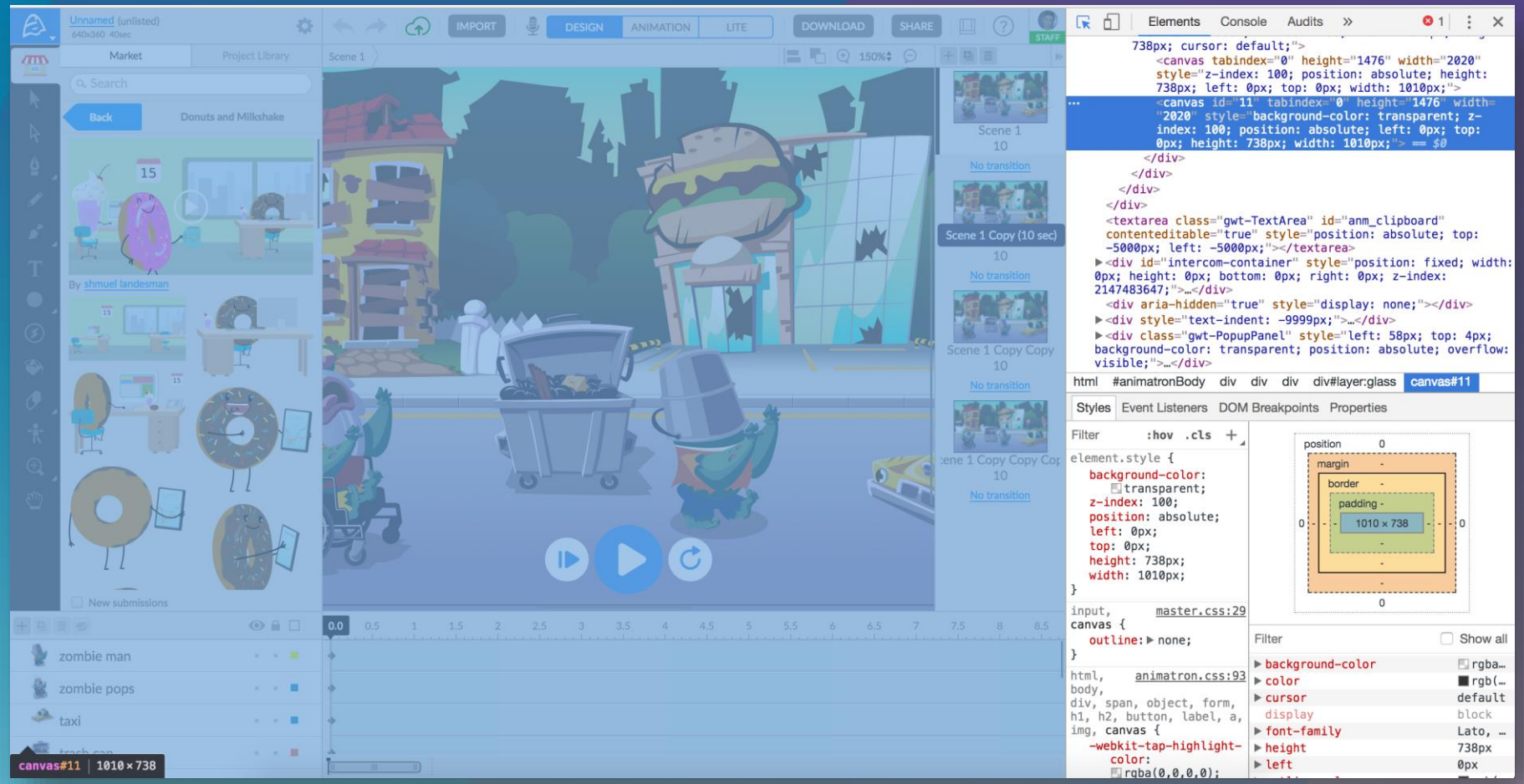
PhoneGap

ANIMATRON

Creating GWT UI

- Create UI Binding Template
- Write UI Java code
- Hand it over to CSS + HTML guy
- The guy styles it
- Next day you get it (at best)
- Roundtrips to fix browser's problems

Everything is Canvas



ANIMATRON

Canvas Based Widgets

Are You F**ing Crazy???

ANIMATRON

Canvas Based Widgets

- Write only Java code
- No CSS + HTML guy
- No roundtrips
- No browser inconsistences

ANIMATRON

Canvas Based Widgets

- This is NOT Swing
- Much simpler than Swing
- Much easier than Swing
- Exactly one look and feel
- Everything is pure Java code
- Many components on the one canvas

ANIMATRON

Examples

```
public class DemoPanel extends TPanel.BorderLayout {
     public DemoPanel() {
          super(20);

          TLabel.Ui title = new TLabel.Ui("Demo Panel");
          title.setHAlignment(Alignment.H.center);

          TTextField.Ui.ForString field = new TTextField.Ui.ForString("");

          TTextArea textArea = new TTextArea("");
          textArea.setPlaceholder(“Type something”);

          TLink.Ui.BroserUrl link = new TLink.Ui.BrowserUrl("GWTCon17", "http://www.gwtcon.org");

          TPanel.FlowLayout flowLayout = new TPanel.FlowLayout(Direction.vertical, 10);
          flowLayout.add(field);
          flowLayout.add(textArea);
          flowLayout.add(link);

          north(title);
          center(flowLayout);
     }
}
```

ANIMATRON

Examples

Demo Panel

Type something

Type something

Welcome to GWTCon 2017

ANIMATRON

Painting Components

```
protected void paintComponent(ACanvas canvas);
public final void invalidate();
public final TComponent revalidate();
public final void repaint();
```

```
public class CustomComponent extends Tcomponent.Ui {
     @Override
     protected void paintComponent(ACanvas canvas) {
          Rectangle rectangle = new Rectangle(0, 0, 100, 50);
          Fill red = new Fill("red");
          canvas.fillRect(rectangle, red);
     }
}
```

```
TButton.Ui button = new TButton.Ui("Please click on me");
button.onMouseDown(new AMouseDownEvent.Handler() {
     @Override
     public void process(AMouseDownEvent event) {
          button.setText("Awesome!");
          button.revalidate();
          button.repaint();
     }
});
```

ANIMATRON

Sizes & Dimensions

```
public final Rectangle getBounds();
public final Rectangle getAbsoluteBounds();
public final Rectangle getContentBounds();
public final Dimension getPrefferedSize();
public <C extends TComponent> C setPrefferedSize(Dimension size);
public <C extends TComponent> C setMaximumSize(Dimension size);
public <C extends TComponent> C setMinimumSize(Dimension size);
```

```
public class CustomComponent extends Tcomponent.Ui {
     @Override
     protected void paintComponent(ACanvas canvas) {
          Rectangle rectangle = getContentBounds();
          Fill green = new Fill("green");
          canvas.fillRect(rectangle, green);
     }
}
```

```
TPanel.BorderLayout layout = new TPanel.BorderLayout();
layout.setMaximumSize(new Dimension(640, 480));
layout.setMinimumSize(new Dimension(32, 32));
```

ANIMATRON

Lists & Grids

```
List<String> someList = Arrays.asList("one", "two", "three");
TList.Model.Simple<String> simpleModel = new TList.Model.Simple<>(someList);
TList.Ui<String> uiList = new TList.Ui<>(simpleModel);

uiList.getSelectionModel().setSelection("two");
String selectedItem = uiList.getSelectionModel().getSelected();

someList.add("four");
uiList.getModel().fireModelChanged();
```

```
List<CustomObject> someList = getCustomObjects();
TList.Model.Simple<CustomObject> simpleModel = new TList.Model.Simple<>(someList);

TGrid.Ui<String> uiGrid = new TGrid.Ui<>(simpleModel);
uiGrid.setColumns(2);
uiGrid.setRenderer(new CustomRenderer());
```

```
class CustomRenderer extends TLabel.Ui implements TFlyweight.Renderer.Active<CustomObject> {
     @Override
     public void prepareToRender(CustomObject object, boolean selected) {
          setText(object.getName());
     }
}
```

ANIMATRON

Viewports

```
List<String> someList = Arrays.asList("one", "two", "three");
TList.Model.Simple<String> simpleModel = new TList.Model.Simple<>(someList);
TList.Ui<String> uiList = new TList.Ui<>(simpleModel);

TViewport.Ui viewport = new TViewport.Ui(uiList);
viewport.setViewWidthTracksViewport(true);
viewport.setViewHeightTracksViewport(true);

TPanel.BorderLayout layout = new TPanel.BorderLayout();
layout.center(viewport);
layout.setPreferredSize(new Dimension(640, 320));
```

ANIMATRON

Styling

```
interface Button extends TLook {
     TLookKey<Boolean> OPAQUE = new TLookKey<>("button.opaque", false);
     TLookKey<Fill> BG = new TLookKey<>("button.background", new Fill("#FFFFFF");
     TLookKey<Fill> FG = new TLookKey<>("button.foreground", new Fill("#000000");
     TLookKey<Fill> BG_PRESSED = new TLookKey<>("button.background.pressed", BG);
     TLookKey<Fill> FG_PRESSED = new TLookKey<>("button.foreground.pressed", FG);
     TLookKey<Fill> BG_HOVERED = new TLookKey<>("button.background.hovered", BG);
     TLookKey<Fill> FG_HOVERED = new TLookKey<>("button.foreground.hovered", FG);
     TLookKey<TFont> FONT = new TLookKey<>("button.font", new TFont("Lato", 10);
}
```

```
public interface THasLook {
     TLookSet getLook();
     <C extends TComponent> C setLook(TLookSet lookSet);
}
```

```
TButton.Ui button = new TButton.Ui("Click");
button.getLook().set(Button.FG, new Fill("red"));

TPanel.FlowLayout flowLayout = new TPanel.FlowLayout(Direction.vertical, 10);
flowLayout.add(field);

flowLayout.getLook().set(Button.FONT, new TFont("Consolas", 12));
```

ANIMATRON

Mouse Events

```java
/**
 * Handler interface for {@link MouseDownEvent} events.
 */
public interface MouseDownHandler extends EventHandler {
  /**
   * Called when MouseDown is fired.
   * @param event the {@link MouseDownEvent} that was fired
   */
  void onMouseDown(MouseDownEvent event);
}
```

```java
public final <C extends TComponent> C onMouseDown(new AMouseDownEvent.Handler());
public final <C extends TComponent> C onMouseUp(new AMouseUpEvent.Handler());
public final <C extends TComponent> C onMouseMove(new AMouseMoveEvent.Handler());
public final <C extends TComponent> C onMouseWheel(new AMouseWheelEvent.Handler());
public final <C extends TComponent> C onMouseDoubleClick(new AMouseDoubleClickEvent.Handler());
```

```java
TButton.Ui button = new TButton.Ui("Click");
button.onMouseDown(new AMouseDownEvent.Handler() {
    @Override
    public void process(AMouseDownEvent event) {
        //TODO: actions
    }
});
```

ANIMATRON

Mouse Events

```
public class GlobalMouseHandler implements MouseDownHandler, … , MouseUpHandler {
     @Override
     public void onMouseDown(MouseDownEvent event) {
          dispatch(new AMouseDownEvent(event));
     }

     @Override
     public void onMouseMove(MouseMoveEvent event) {
          dispatch(new AMouseMoveEvent(event));
     }

     …

     @Override
     public void onMouseUp(MouseUpEvent event) {
          dispatch(new AMouseUpEvent(event));
     }

     private void dispatch(AMouseEvent event) {
          …
     }
}
```

ANIMATRON

Canvas Based Widgets

- Core framework development takes time
- Hard to create automated tests
- Everything is done manually
- Big Canvas can be slow
- Text rendering can cause problems
- Sometimes require special hacks

ANIMATRON

Future of Java in Rich Web Apps

GWT

vaadin

Java + Clojure

ANIMATRON

Relocation to Lviv:

- Front-End Developer (Java)
- Backend Developer (Java/Scala)
- Front-End Developer (Typescript/AngularJS/React)
- DevOps Engineer (with Amazon AWS Experience)

For more information: ykobyliuk@animatron.com

ANIMATRON