

GWT Animation - GWT FX / A Google Web Toolkit használatáról

Google Web Toolkit (GWT) is a development toolkit for building and optimizing complex browser-based applications. <https://developers.google.com/web-toolkit/>

You write your code in Java language and the GWT compiler will produce to you the client side javascript(/ajax) and html files.

There are lot of additional java libraries which you can use to improve your GWT application for free:

<http://code.google.com/p/gwt-google-apis/downloads/list>

For example to do animations, you can use the custom gwt animation tool and components, but there is also a java package named gwt-fx, reachable from the list above mentioned.

With gwt-fx you have additional animation possibilities:

<http://code.google.com/p/gwt-fx/>

I write a short application to demonstrate some animation functionalities using the gwt-fx package and the custom gwt animation component. This application is running on the free Google Application Engine server:

<http://lehelsipos-gwt1.appspot.com/>

Using the custom GWT animation component, a short example demonstrates us, a function of moving widgets:

```
class CustomAnimation extends Animation {
    private int centerX = 120;
    private int centerY = 120;
    private int radius = 100;

    @Override
    protected void onComplete() {
        super.onComplete();
        //write code here, what to do after the animation stops
    }
    @Override
    protected void onStart() {
        super.onStart();
        //write code here, what to do before the animation starts
    }

    @Override
    protected void onUpdate(double progress) {
        double radian = 2 * Math.PI * progress;
        updatePosition(your_gwt_widget1, radian, 0);
        updatePosition(your_gwt_widget2, radian, 0.5 * Math.PI);
        updatePosition(your_gwt_widget3, radian, Math.PI);
        updatePosition(your_gwt_widget4, radian, 1.5 * Math.PI);
    }
    private void updatePosition(Widget w, double radian, double offset) {
        radian += offset;
        double x = radius * Math.cos(radian) + centerX;
        double y = radius * Math.sin(radian) + centerY;
        YourPanelContainingTheWidgets.setWidgetPosition(w, (int) x, (int) y);
    }
}
```

At your button click event you can use the following code to start and stop this custom gwt animation:

CustomAnimation animation;

```
.....  
animation.run(4000);  
animation.cancel();
```