Lab 5: SVG 2 – Scalable Vector Graphics

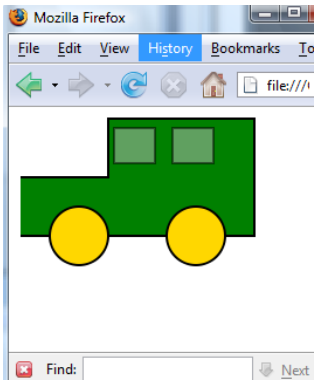
The aim of this exercise is to give you experience with programming SVG, through a series of tasks. You may need to refer to the online specification at http://www.w3.org/TR/SVG/

You will work through a series of tasks. You will need to use an editor (such as Notepad) and the FireFox 3browser (which should be installed on your desktop).

Note 1. The Firefox browser does **not** include a full implementation of the whole SVG specification (e.g. it does not include the animate node) however, it is complete enough for this exercise.
Note 2. You will need to use the full header:

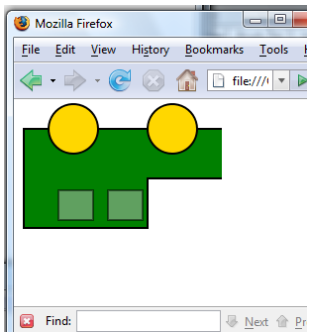
```
<?xml version="1.0"?>
<svg version="1.1"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
…. put your code here …
</svg>
```

Tasks



1. Now make a car: - The car could be two green rectangles (one on top of the other) and with two *gold* circles. Make the windows transparent (and silver).
   - Hint, this car is 200 long, 100 high (excluding the wheels); each wheel has a 25 radius.

2. Now change the car such that it is designed by a polyline rather than two rectangles.



3. Make the car svg code as generic as you can; e.g. only one def for green; only one def for 1 wheel.

4. Experiment with translating the car – flip it over: shear the car… etc. using <g>

5. Now make a pattern. Patterns contain regular, repeating shapes (that may be symmetrical and could be circular). E.g., you could do a square based pattern that alternates red and green squares (a checkerboard pattern). Or a circular pattern that looks like a rose.

   There are different ways to do this. Think how you can define a polyline or a path to create a regular pattern, or how to use transformation to repeat a regular object. This will turn into your second assessment.

6. If you have time experiment with text
   - add some text;
   - then put the text on a path (e.g. a circle).