Coursework 2: Mario Kart XML

ECS650U/ECS789P Semi-Structured Data & Adv. Data Modelling

School of Electronic Engineering & Computer Science
Queen Mary University of London

Autumn 2018

This coursework involves working with XML and XSLT. It is an individual coursework. You should complete the XSLT lab before attempting it.

Deadline: 9am, Friday 14th December 2018.

1   Data

Download `mario.zip` from QMPlus. It contains:

- Four JSON files with data on characters and vehicles from *Mario Kart 8 Deluxe*;

- An incomplete DTD `mariokart.dtd` for an XML application describing this data.

Characters have base statistics which are modified by the choice of vehicle body, tire and glider. For example, Baby Peach's base acceleration is 4, but the Landship body provides a 0.5 modifier for acceleration, so their combined acceleration is 4.5.

2   Submission

The coursework submission (via QMPlus) should be a Zip file containing only the required files with no subdirectories. You should name the files as requested.

3   XML (40%)

Files to submit: `mariokart.xml` and `mariokart.dtd`
Marking criteria: A well-formed XML file that validates with your DTD.

3.1 `mariokart.xml`

Create a file `mariokart.xml` containing all the character and vehicle data as XML instead of JSON. You may find the following Oxygen features useful:

- Convert JSON to XML (under *Tools*).

- Format and Indent (under *Document / Source*).

3.2 `mariokart.dtd`

Modify `mariokart.dtd` to reflect your design. Use it to validate `mariokart.xml`.

4 XSLT (60%)

Write three separate XSLT stylesheets that generate the required XML output.

Files to submit: `names.xsl`, `landspeed.xsl`, `accel.xsl`
Marking criteria: Each XSLT stylesheet generates the correct data in the correct format.

4.1 `names.xsl`

Generates an alphabetical list of all character, glider, tire and body names.
    Output format:

```
<results>
  <name>Name 1</name>
  <name>Name 2</name>
  ...
</results>
```

4.2 `landspeed.xsl`

For each combination of tire and body, calculates the combined land speed modifier, i.e. (tire land speed) + (body land speed).
    Output format:

```
<results>
  <result>
    <tire>Character name</tire>
    <body>Body name</body>
    <speed>Combined land speed</speed>
  </result>
  ...
</results>
```

4.3 `accel.xsl`

Uses `for-each-group` to group the characters by their base acceleration and sort the groups.

Output format:

```
<results>
  <result accel="lowest-value">
    <name>Character A</name>
    <name>Character B</name>
    ...
  </result>
  ...
  <result accel="highest-value">
    <name>Character X</name>
    <name>Character Y</name>
    ...
  </result>
</results>
```