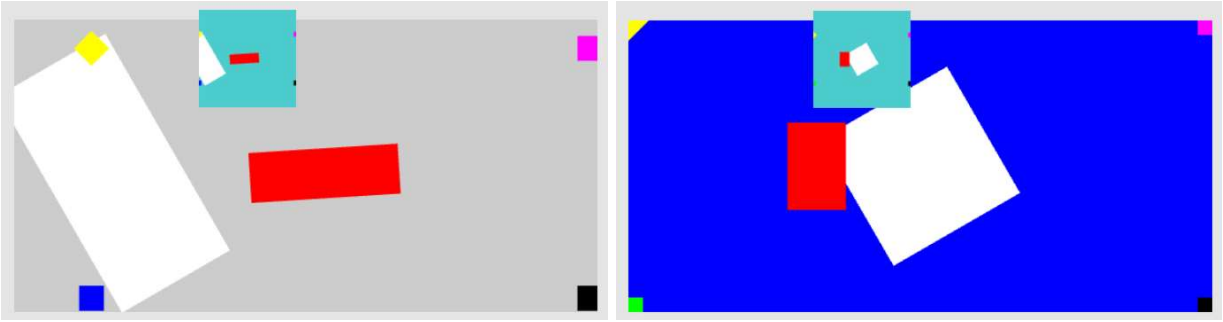
# CSS 452: Programming Assignment #3
## Resource Management and Scenes

**Due time:** Please refer to our course web-site

### Objective
In this programming assignment we will work in real-time environment, input from a JSON scene file (another popular format), and verify our understanding of viewport, and WC space.

---

### Assignment Specification:
Here is an example of the results from this assignment:



Assignment specifications:

- **Two Scenes**: You must support at least two scenes: (please refer to this file: https://myuwbclasses.github.io/CSS452/CourseMaterials/MP3/assets.zip)
  - First Scene: Gray Scene: specified by: **scene.json**.
  - Second Scene: Blue Scene: specified by: **BlueLevel.xml**
  - The scenes are transitioned with the 'Q' command.
- **Real time movements**: In the Gray scene, notice that:
  - **The red rectangle:** rotates at a rate of one complete revolution per 5 seconds
  - **The white rectangle:** moves towards the left and wraps around at a speed of 20 units per 3 seconds.
- **Small Viewport (in green)**: You can control the Device Coordinate (DC, or pixel positions) location of this viewport with the WASD keys.
- **Large WC Coordinate**: You can control the WC coordinate systems of the large view with the FCVB keys for translation and ZX for zooming in and out.
  - **Warning:** you will have to modify the input component to support additional key codes.
- **Input support:** Modify the input component to support "KeyReleased" event (when a key state transitions from pressed to released).
  - **Small viewport:** left-ward movement (the A-key control) is triggered by the "Key Released" event.
- **Saved game state information:** The small camera view is preserved over scene transitions. This can be confirmed by the location of the viewport for this camera: the DC location is preserved over scene transitions.

---

### Hints:
1. My implementation is based on book Example-4.6 (*AudioSupport*). You do _NOT_ need to support audio in this assignment.

2. Go read up on JSON file format and how it is supported in JavaScript. The parsing is trivial. A couple of points,
   a. I learned how to parse JSON by examining these two sites:
      i. http://www.w3schools.com/js/js_json.asp
      ii. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse

b. I load the JSON file as a plain text file (NOT an XML file)
c. To parse the JSON file in my Gray Scene, I did:

```
var jsonString = gEngine.ResourceMap.retrieveAsset(this.kSceneFile);
var sceneInfo = JSON.parse(jsonString);
```

After the above line, ***sceneInfo*** can be used as a defined JavaScript object.

3. "Zooming" can be accomplished by increasing/decreasing WC-width.

4. Don't forget, you will have to modify ***Engine_Input.js*** to support the "KeyRelease" event, and, to support the additional keycodes that are required.

5. You are recommended (please!) to use the ***ResourceMap*** component for saving and restoring game state information. In this case, define additional function(s) to support the storing and retrieving of the small view camera.

---

## Credit Distribution

Here is how the credits are distributed in this assignment:

| | | | |
|---|---|---|---|
| 1. | Parsing JSON scene file and scene transitions | | 30% |
| | a. Parsing and displaying of the JSON scene file<br>b. Parsing and displaying of the XML scene file<br>c. Support scene transitions with "Q" key | 20%<br>10%<br>20% | |
| 2. | Small camera view: Viewport control | | 15% |
| | a. WASD manipulate the Viewport<br>b. The "A" key is triggered by KeyRelease event | 5%<br>10% | |
| 3. | Large camera view: WC control | | 15% |
| | a. FCVB manipulate the WC Window<br>b. ZX zooms in/out | 10%<br>10% | |
| 4. | Keyboard control + Speed | | 20% |
| | a. Support KeyRelease event ("A"-Key)<br>b. Support all above keys properly<br>c. Rotation speed (1 revolution / 5 sec)<br>d. Movement speed (20 units / 3 sec) | 10%<br>5%<br>5%<br>5% | |
| 5. | Saving of game state (in ResourceMap) | | 10% |
| | a. Modify ResourceMap class<br>b. Small view camera Viewport is preserved over different scenes | 10% | |
| 6. | Proper submission | | 10% |
| | a. Zip file names with **NO SPACES**<br>b. No extra unused files/folders (E.g., Test folder)<br>c. Styles (project name, variable names, etc.) | 10%<br>10%<br>10% | |

This programming assignment will count 11% towards your final grade for this class.

---

**Creativity and Extra Credits:** Your first two scenes MUST BE defined by the two provided files!! Sorry, but to facilitate easy grading, the first two scenes _MUST_ be the same as mine. You are free to create additional scenes if you like. BUT, the first two scenes must be identical to mine.

- Please do feel free to include your own scenes, HOWEVER, please make sure you support transitions between scenes with the Q key. Also, please make sure the small view camera is preserved between Gray and Blue scenes.