Here is the discussion forum for this part of the course. Please either post your comments/observations/questions or share your creations.

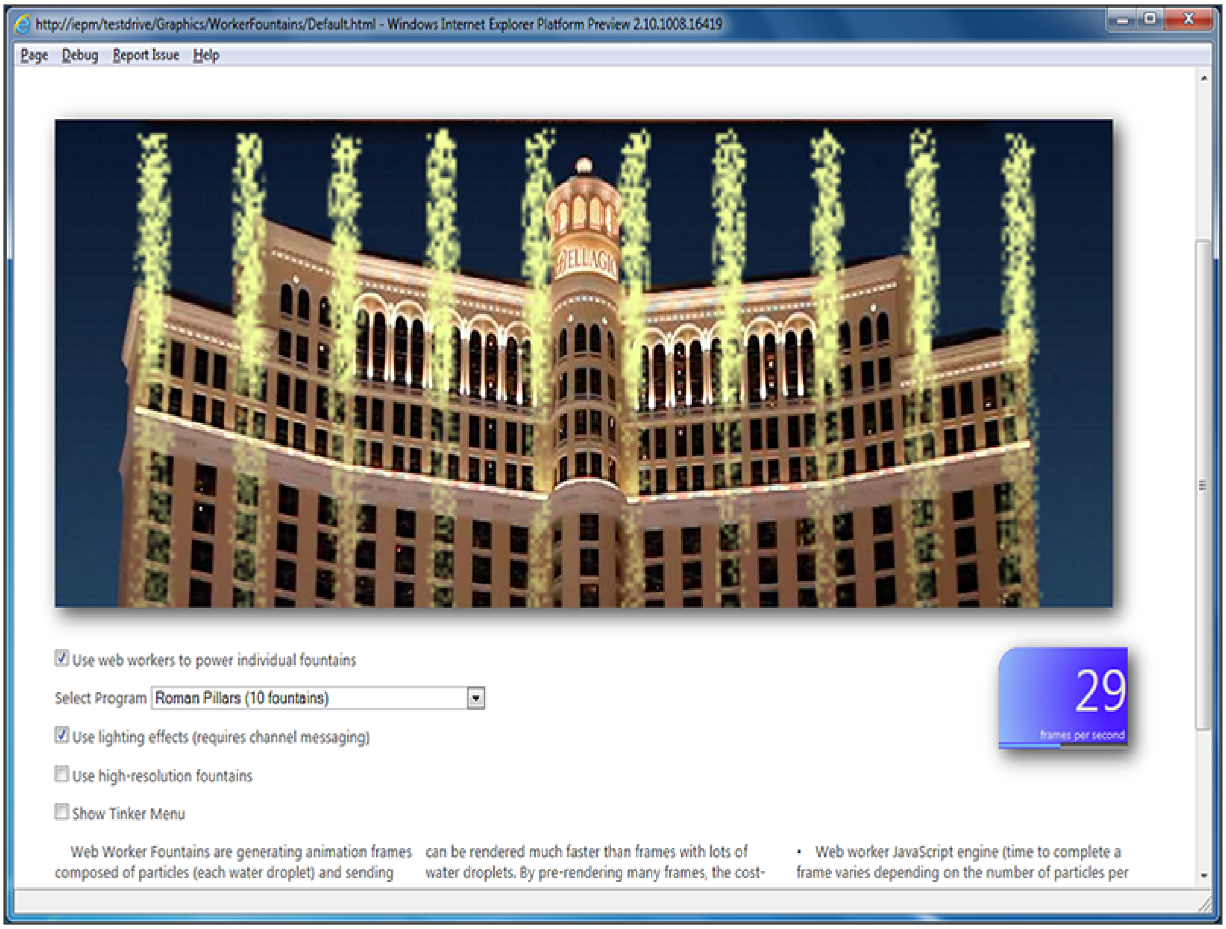
Let us suggest some topics of discussion and optional projects:

SUGGESTED TOPICS

- Did you try the demos from the last lesson? Do you understand why using Web Workers can be a savior in some situations?
- Can you find some explanations on the Web about multi core architectures and Web Workers (e.g., about threads/workers benefiting from multi core processors, leading to greater performance). Please share any relevant articles in the forum!

OPTIONAL PROJECTS

- Please write a small Web app. that uses Web Workers.
- There was a wonderful demonstration about fountain animation with particles, made by Microsoft. Unfortunately, this demo is no longer online. Can you write something similar, but perhaps with fewer options? The idea was the following: compute particle movements in separate workers, and when a new array of particles is ready to be drawn, post it from the Web Worker. The main page has a mainloop for animating at 60 frames per second. When a new set of particles is ready (posted by a Worker), it is drawn and animated. The demo had up to 10 workers working in background.


http://iepm/testdrive/Graphics/WorkerFountains/Default.html - Windows Internet Explorer Platform Preview 2.10.1008.16419
Page
Debug
Report Issue
Help
BELLAGIO
29
frames per second

☑ Use web workers to power individual fountains

Select Program: Roman Pillars (10 fountains)

☑ Use lighting effects (requires channel messaging)

□ Use high-resolution fountains

□ Show Tinker Menu

Web Worker Fountains are generating animation frames composed of particles (each water droplet) and sending

can be rendered much faster than frames with lots of water droplets. By pre-rendering many frames, the cost-

- Web worker JavaScript engine (time to complete a frame varies depending on the number of particles per