js-transition-perf-measures

Alex Masselot

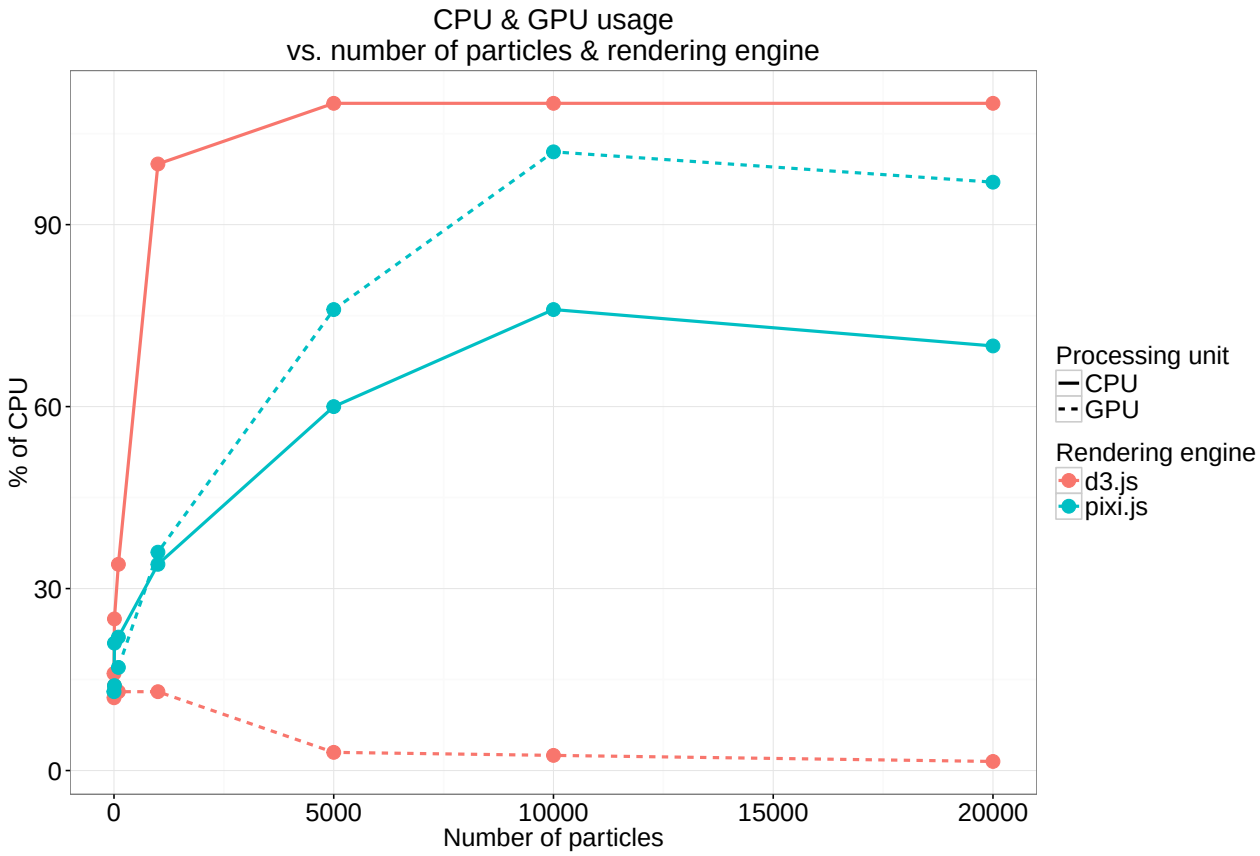
12/23/2016

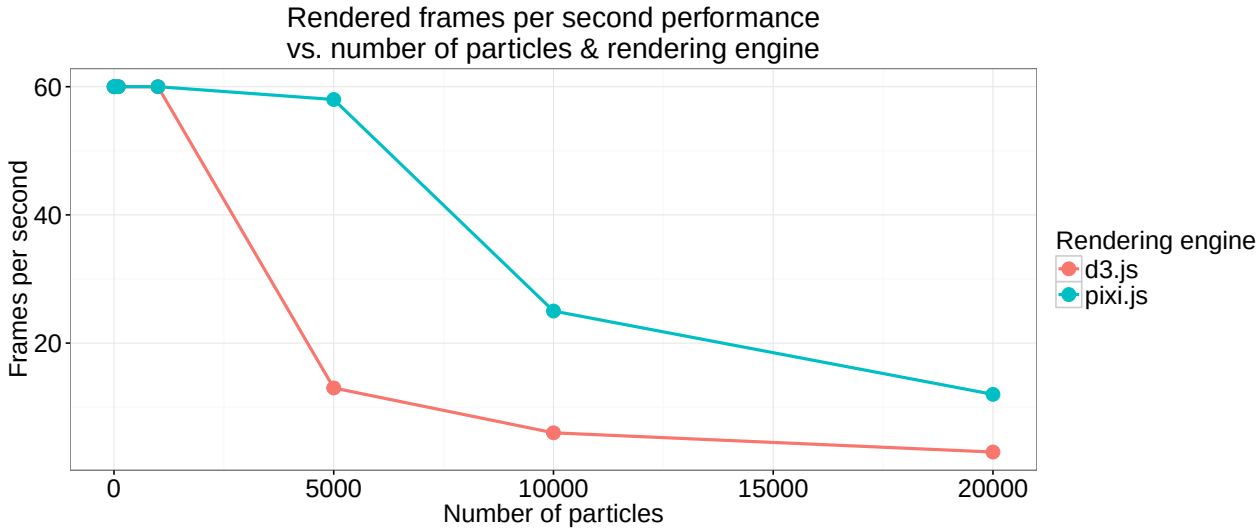
```
library(ggplot2)
```

You can also embed plots, for example:

```
dt.cycle.times = read.delim('data/cycle-times.txt')
dt.perf = read.delim('data/perf-data.txt')
dt.perf$engine=paste(dt.perf$engine, '.js', sep='')
dt.perf$is.transparent = dt.perf$opacity <1
dt.perf = subset(dt.perf, browser == 'chrome')
```

```
.dt = subset(dt.perf, ! is.transparent)
dt.process.unit = rbind(
  data.frame(engine=.dt$engine,
             n=.dt$n,
             usage=.dt$cpu,
             process.unit='CPU'
             ),
    data.frame(engine=.dt$engine,
               n=.dt$n,
               usage=.dt$gpu,
               process.unit='GPU'
               )
)
```






```
dt.cycle.times$nb.particles = factor(dt.cycle.times$nb.particles)
q = ggplot(dt.cycle.times, aes(factor(nb.particles), cycle.time)) +
  geom_hline(yintercept = 1000, linetype='dotted', lwd=1)+
  geom_boxplot(aes(colour=engine)) +
  ylim(c(800, max(dt.cycle.times$cycle.time))) +
  theme_bw() +
  theme(
    plot.title = element_text(size = rel(1.6)),
    legend.title = element_text(size = rel(1.4)),
```

```
    legend.text = element_text(size = rel(1.4)),
    axis.title = element_text(size=rel(1.4)),
    axis.text = element_text(size=rel(1.4)),
    axis.text.x = element_text(angle=45, hjust=1)
    ) +
  geom_label(x=7,
             y=950,
             label="optimal cycle time",
             hjust=0.8) +
  guides(colour = guide_legend(title = "Rendering engine")
         ) +
  labs(
    title="Cycle times distribution\nvs rendering engine & number of particles\n(1000ms is perfect cycle
    x='Number of particles',
    y='Cycle time (ms)'
  )
print(q)
```