

# Introduction



The animation consists of running horses, flying parrots and flamingos. A cube texture is used to create the open environment and for GLTF objects are used for animals.

## Technologies / Tools used

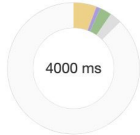
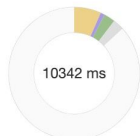
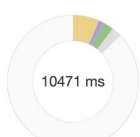
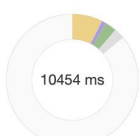
1. ThreeJS
2. WebGL
3. Stats library
4. Google Chrome Developer tools

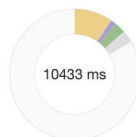
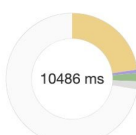
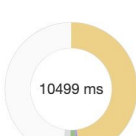
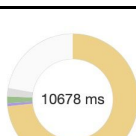
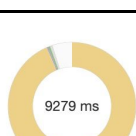
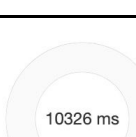

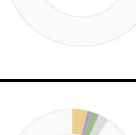
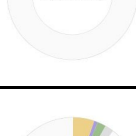
## Analysis

System specifications:

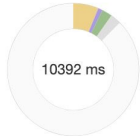

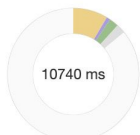
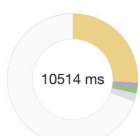
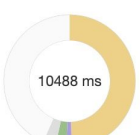
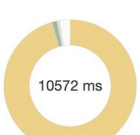
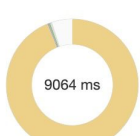
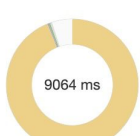
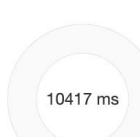
Device	MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)
Processor	2.3 GHz Intel Core i5
Memory	8 GB 2133 MHz LPDDR3
Graphics	Intel Iris Plus Graphics 640 1536 MB

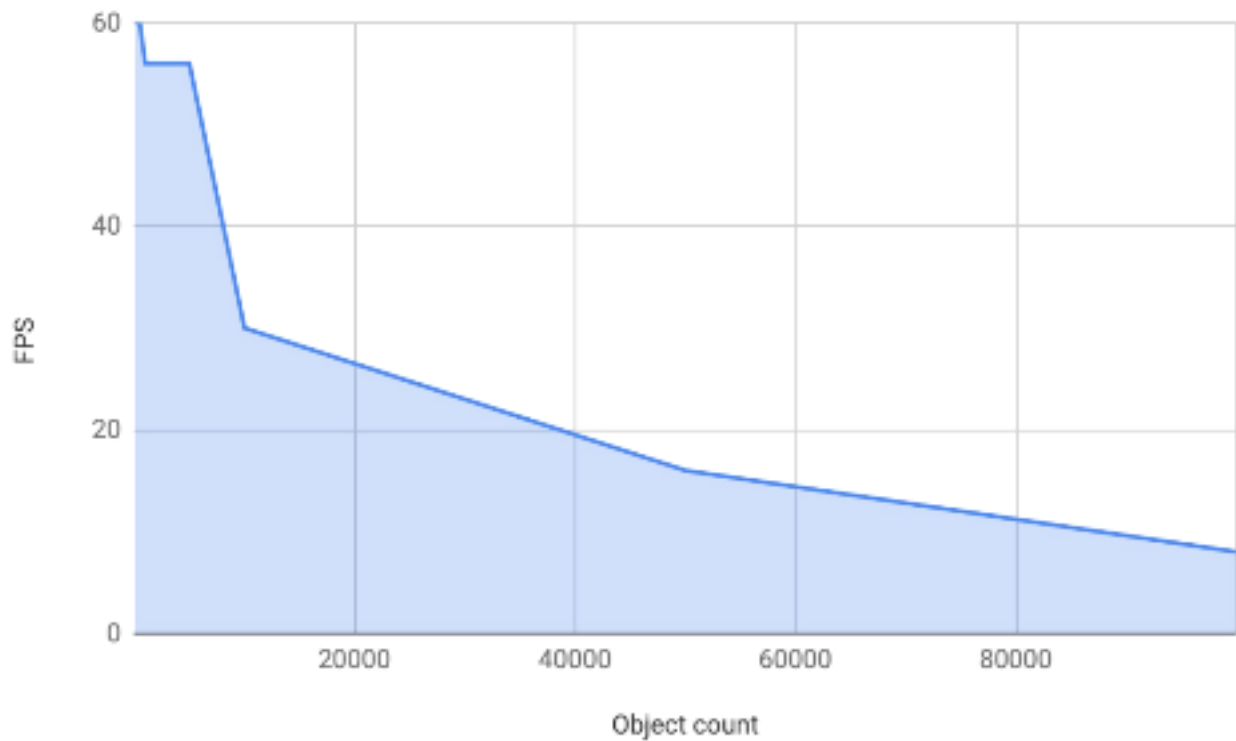
## Performance

Vertices count	Polygon count	Object count	Time to render (ms)	Memory (MB)	FPS	Other
361	554	1	17	16	60	 <p>4000 ms</p> <ul style="list-style-type: none"> <li>225.9 ms Scripting</li> <li>39.9 ms Rendering</li> <li>116.3 ms Painting</li> <li>118.6 ms System</li> <li>3499.5 ms Idle</li> </ul>
361	554	5	17	21	60	 <p>10342 ms</p> <ul style="list-style-type: none"> <li>688.9 ms Scripting</li> <li>105.6 ms Rendering</li> <li>286.1 ms Painting</li> <li>286.3 ms System</li> <li>8975.1 ms Idle</li> </ul>
361	554	10	17	21	60	 <p>10471 ms</p> <ul style="list-style-type: none"> <li>682.6 ms Scripting</li> <li>118.6 ms Rendering</li> <li>281.4 ms Painting</li> <li>266.4 ms System</li> <li>9121.5 ms Idle</li> </ul>
361	554	50	17	21	60	 <p>10454 ms</p> <ul style="list-style-type: none"> <li>780.3 ms Scripting</li> <li>109.6 ms Rendering</li> <li>292.3 ms Painting</li> <li>272.4 ms System</li> <li>8999.8 ms Idle</li> </ul>

361	554	100	17	22	60	 <p>1001.5 ms Scripting 121.6 ms Rendering 306.3 ms Painting 316.2 ms System 8687.1 ms Idle</p>
361	554	500	17	34	60	 <p>2382.6 ms Scripting 84.9 ms Rendering 213.5 ms Painting 209.4 ms System 7595.5 ms Idle</p>
361	554	1000	18	41	56	 <p>5023.1 ms Scripting 75.5 ms Rendering 176.3 ms Painting 228.8 ms System 4995.4 ms Idle</p>
361	554	5000	18	65	56	 <p>7826.9 ms Scripting 81.1 ms Rendering 176.2 ms Painting 223.2 ms System 2370.6 ms Idle</p>
361	554	10000	30	162	30	 <p>8681.7 ms Scripting 24.3 ms Rendering 54.9 ms Painting 63.3 ms System 454.7 ms Idle</p>
361	554	50000	60	485	16	 <p>10326.5 ms Idle</p>
361	554	100000	120	860	8	 <p>10326.5 ms Idle</p>
820	996	1	17	16	60	 <p>401.0 ms Scripting 91.9 ms Rendering 215.0 ms Painting 279.6 ms System 9536.3 ms Idle</p>
820	996	5	17	21	60	 <p>546.3 ms Scripting 93.7 ms Rendering 219.8 ms Painting 229.5 ms System 9355.8 ms Idle</p>



820	996	10	17	28	60	 <p>648.6 ms Scripting 111.8 ms Rendering 264.7 ms Painting 276.3 ms System 9090.4 ms Idle</p> <p>10392 ms</p>
820	996	50	17	21	60	 <p>820.9 ms Scripting 118.7 ms Rendering 295.0 ms Painting 281.6 ms System 9081.1 ms Idle</p> <p>10597 ms</p>
820	996	100	17	22	60	 <p>917.5 ms Scripting 95.5 ms Rendering 249.8 ms Painting 244.6 ms System 9232.3 ms Idle</p> <p>10740 ms</p>
820	996	500	17	29	60	 <p>2716.3 ms Scripting 85.4 ms Rendering 193.5 ms Painting 219.4 ms System 7298.9 ms Idle</p> <p>10514 ms</p>
820	996	1000	17	34	60	 <p>5190.2 ms Scripting 132.9 ms Rendering 283.6 ms Painting 327.3 ms System 4554.3 ms Idle</p> <p>10488 ms</p>
820	996	5000	30	132	30	 <p>10111.8 ms Scripting 33.1 ms Rendering 83.1 ms Painting 80.5 ms System 263.3 ms Idle</p> <p>10572 ms</p>
820	996	10000	40	160	15	 <p>8509.3 ms Scripting 19.6 ms Rendering 42.8 ms Painting 43.4 ms System 449.0 ms Idle</p> <p>9064 ms</p>
820	996	50000	250	530	5	 <p>8509.3 ms Scripting 19.6 ms Rendering 42.8 ms Painting 43.4 ms System 449.0 ms Idle</p> <p>9064 ms</p>
820	996	100000	500	886	2	 <p>10417.2 ms Idle</p> <p>10417 ms</p>



## Optimization Techniques

- Reducing the number of polygons will increase the performance
- Try to reduce the number of objects display to increase the performance
- When rendering select some portion of objects and render others in next round. It will reduce the rendering objects and it will increase the performance
- Not rendering in every round and render in specific time periods.

