**Performance Bounceback Exercise**

*by Rahul Bethi*

This Project was done as an exercise of VR nano-degree Udacity program. Its main goal is to implement optimization skills learned in this course.

Optimization is done on the project taken from the exercise. All the gameplay is maintained, infact, some gameplay is added to make it more interesting. Along with the balls spawning every 0.5 second, the game counts a 60 second timer for the player to throw the balls and score. If the timer is run out, player cannot score until he resets by pressing the Y button on the left controller.

****

Timer, score and instructions are shown on the pillar in front of the player stage area. Frames per second (FPS) is also displayed along with them. Menu (start) button of left controller was assigned to Quit the application.

**Following Optimizations are done:**

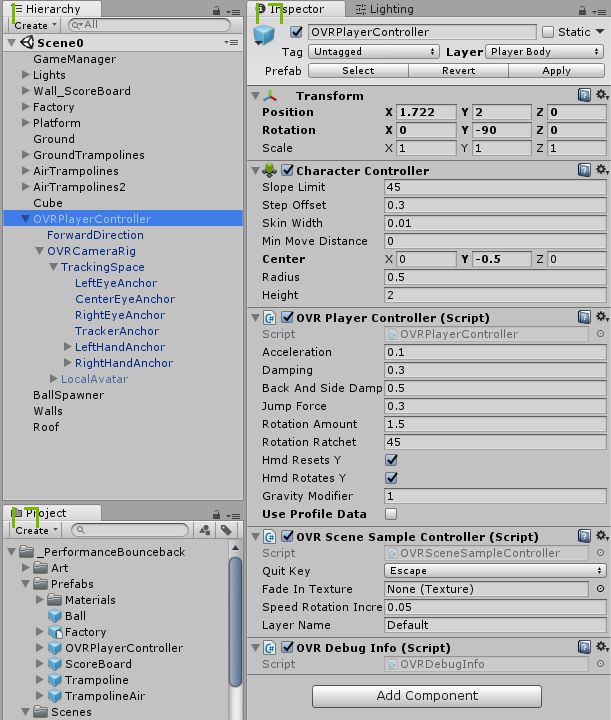
All non-moving objects in the game are made static and Dynamic batching is enabled.

Physics update rate is adjusted to 0.0111111 interval in a second ~ 90 times per second, which is same as the VR framerate 90 FPS.

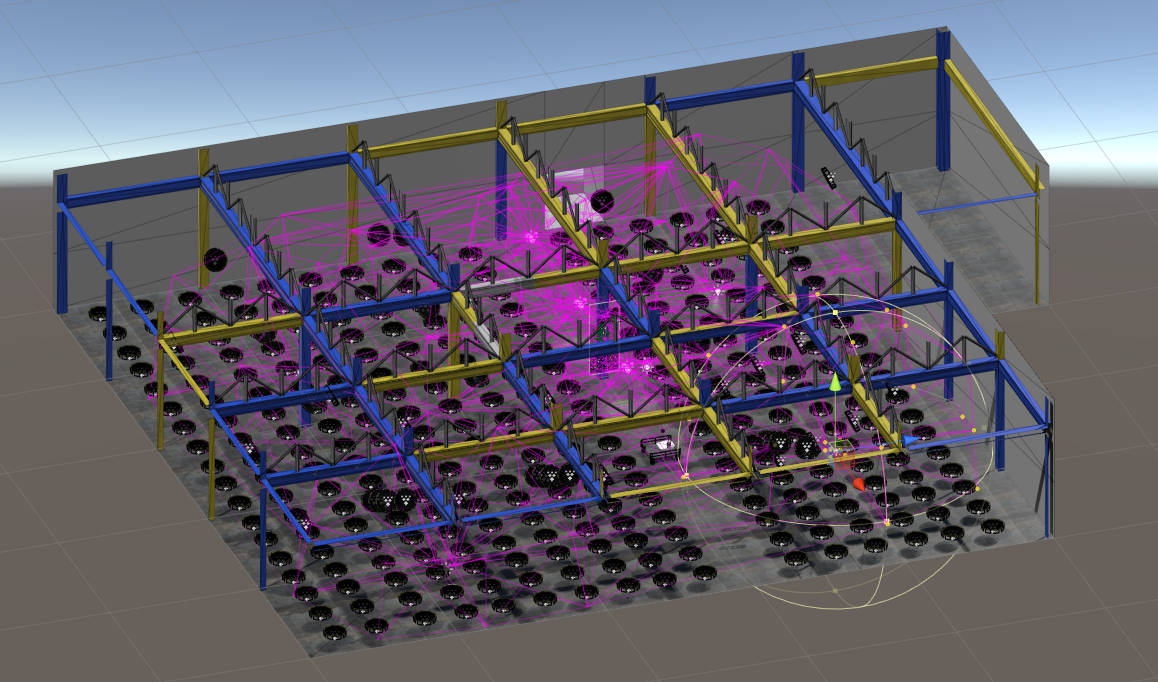
All moving bodies have Rigid-bodies attached to them and are not static.

Balls are reused by checking each ball’s velocity and reusing the balls which are slower than speed 0.9 units per second. This way, only the balls which are slow and not in use (grabbed with hand) are reused. Speed is taken as a factor to decide because a ball which is just thrown cannot be reused in between the scoring. Total balls are limited to 30. This can be reduced by increasing the limit of the ball velocity. As my computer has a good graphics card, this was more than enough for my computer configuration.

Almost all variables are declared only once. Debug messages are controlled by on/off bool variable in GameManager script. Oculus SDK scripts are used for player Controller and Rig. Even Grabbing object was taken from Oculus package.



Lighting rendering path is changed to Forward. And light probes are used around the point lights to get the lighting data around them. Minimum amount light probes are used as the light was sufficient and to get performance. Along with these, 8x MSAA is enabled to get best picture quality. Mixed lighting is used for all lights. This gave the best balance between quality and performance.



Finally, after all these optimizations, game was running at 90 FPS.

***Links:***

Source code: <https://github.com/bethirahul/Performance-Bounceback>

Game: <https://goo.gl/gEjrsr>

My LinkedIn: <https://www.linkedin.com/in/rahulbethi/>