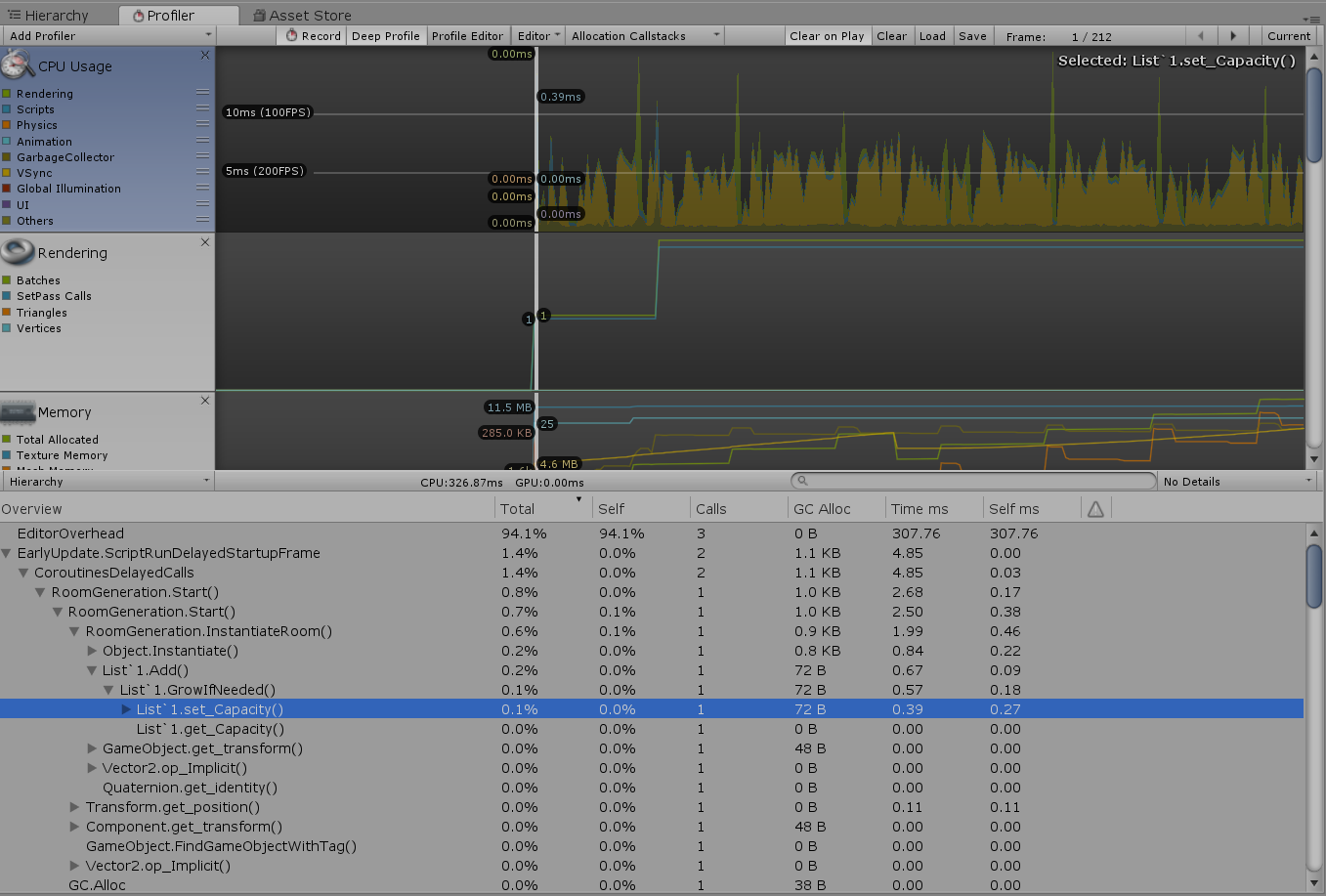
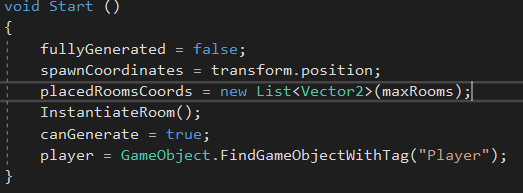
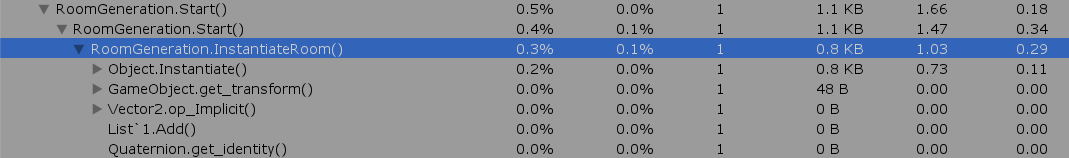
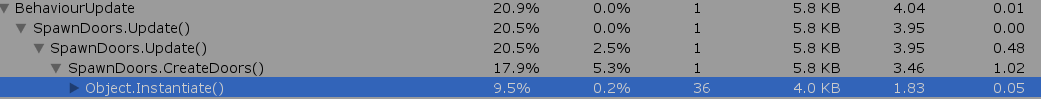
Using the Profiler



After making this change it completely got rid of time and allocation that was having a small affect on the load-up of the game as seen in the updated picture below.

When first using the profiler, I noticed there was a spike on the CPU usage at the very start, the main cause of this was using the ‘Instantiate’ function which was needed, therefore I was unable to do something about. However, adding to the list that contains the coordinates of the rooms was also affecting the usage and it seemed that this was because it had to set the capacity before adding to the list. To try and solve this issue I made sure to set the capacity of the list to the number of max rooms as the list shouldn’t exceed this limit.



The only other spike on the profiler is when the doors are spawned once the rooms have finished spawning. This is also due to them needing to be instantiated. Using an object pool could be useful if the dungeons that were going to be created were going to have lots of rooms. However, as it stands, I don’t believe that using an object pool with the low amount of usage that is used is required. Another way this could be tackled would be by only instantiating the doors once the player has entered the room but this could cause problems once using within a game as there will be many other actions that will need to take place upon the player entering the room that having all doors spawn at the start, possibly slightly increasing the loading time of the dungeon, will be better than the possible framerate issues that will come later down the line when incorporating this within a project.