1. Platforma sprzętowa

Profilowanie zostało wykonane na laptopie o podanej specyfikacji:

- CPU: Intel Core i5-7300HQ

- GPU: Nvidia GeForce GTX 1050

- RAM: 8 GB

2. Testowana aplikacja

Testową grą jest mój stary projekt zaliczeniowy pt. „Capsule City Brawl”, do którego repozytorium do wglądu znajduje się tutaj: Link do repozytorium

Gra posiada zbudowana jest na silniku Unity i posiada trzy sceny:

- Menu główne (MainMenu)

- Mapę „Warehouse” (Level1)

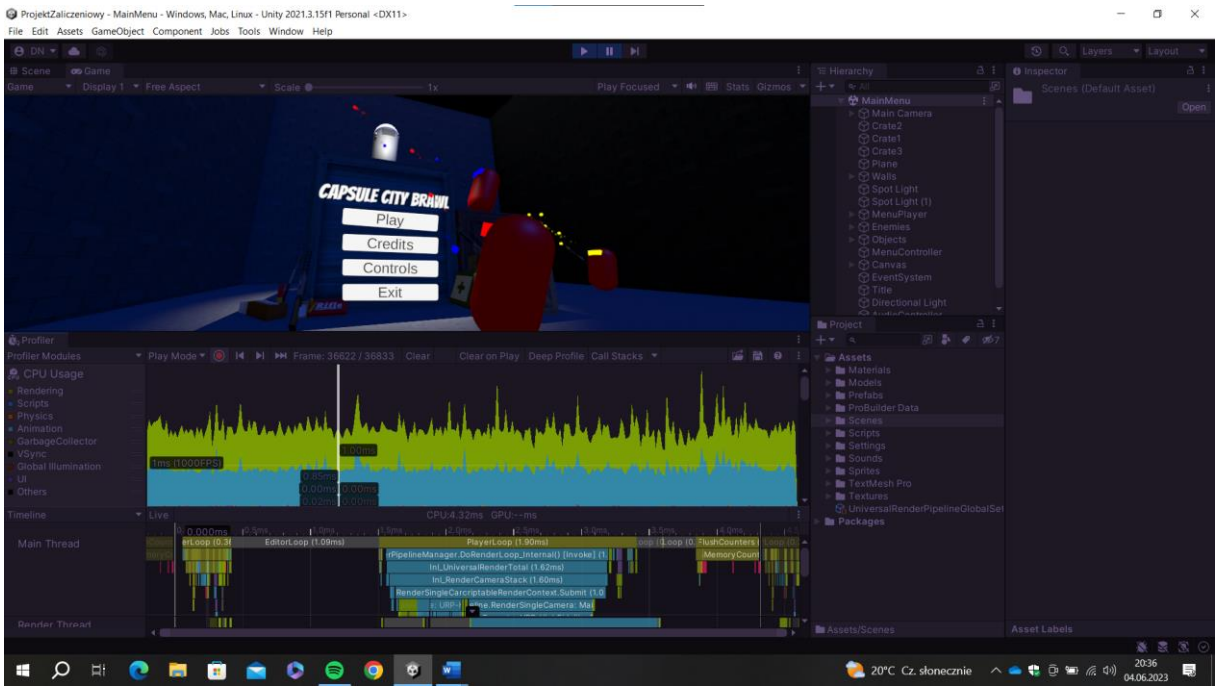
- Mapę „Whiskey Lovers Club (Level2)

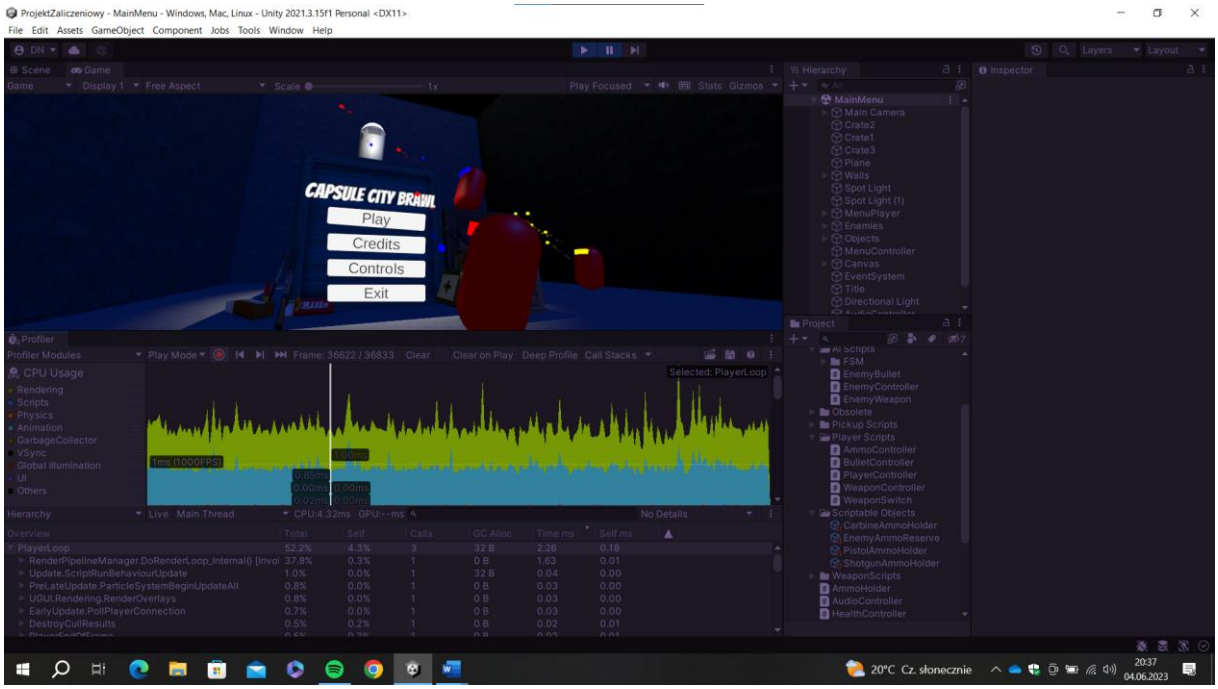
Profilowanie zostanie przeprowadzone na każdej z tych scen, jako odniesienia będę używał nazw w nawiasach. Wyniki profilowania map z przeciwnikami będą realizowane przy maksymalnej ilości przeciwników, jaka jest dozwolona do bycia na ekranie.

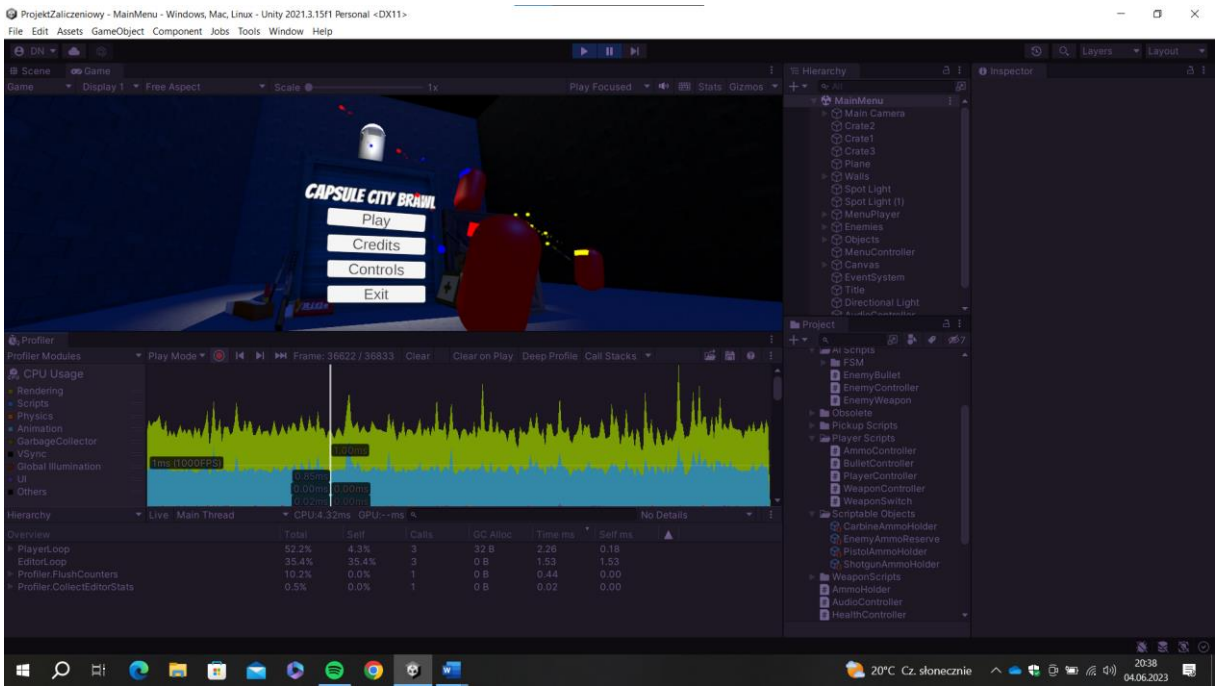
3. Wyniki profilowania

a) MainScene

Wyniki profilowania dla sceny „MainScene”

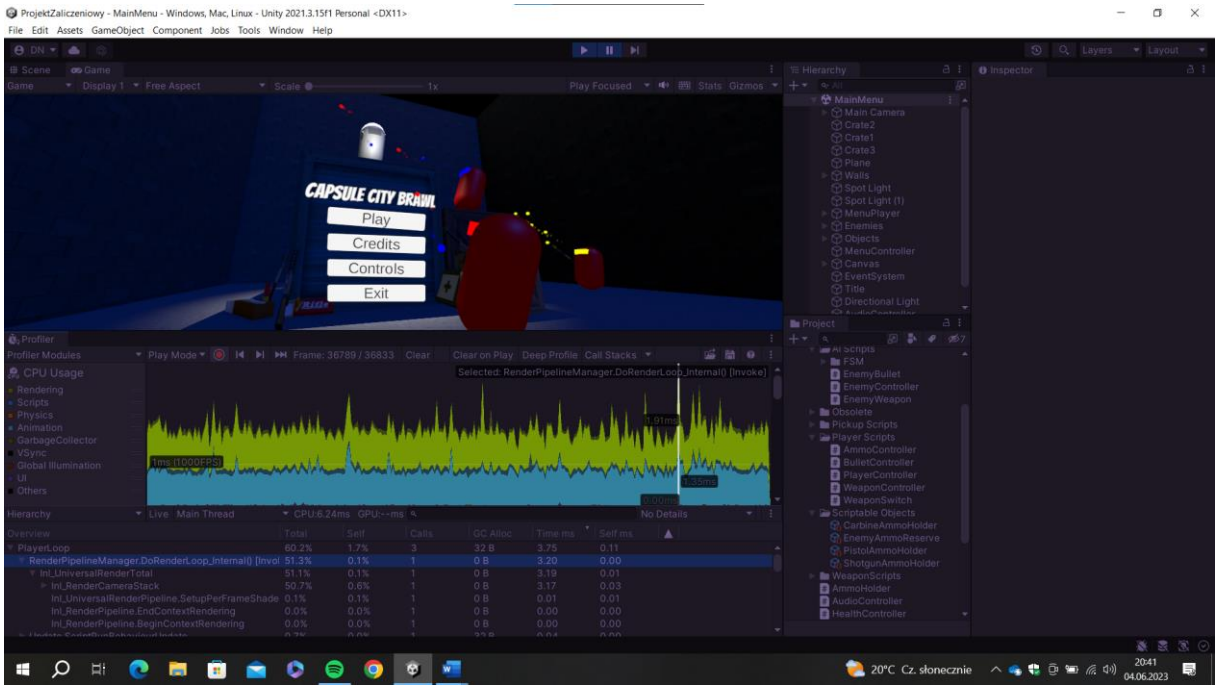


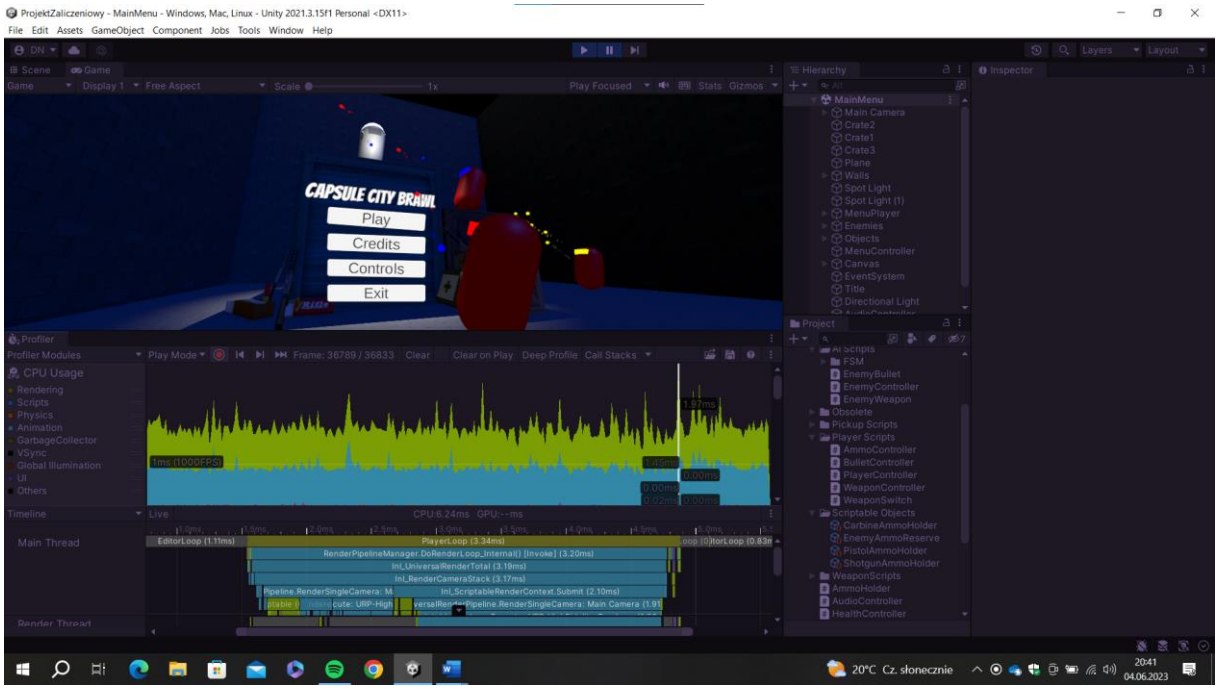




Z powyższych zrzutów ekranu można zauważyć, że w tej scenie najwięcej czasu przeznaczone jest na URP.

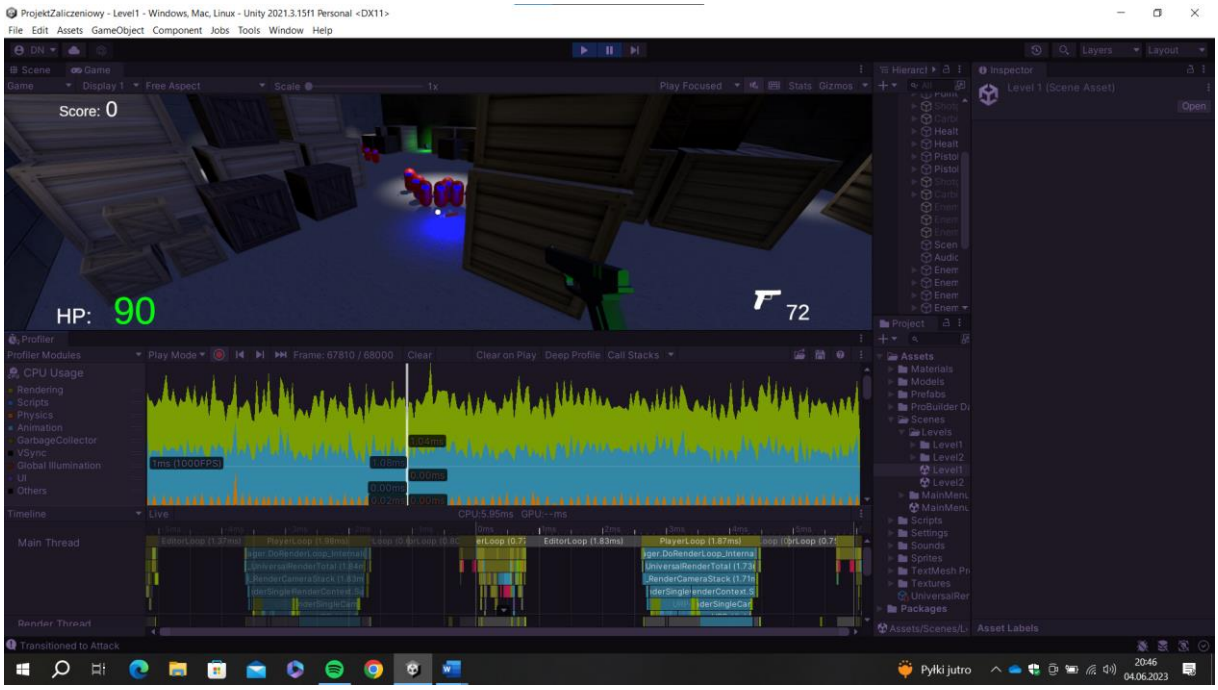
Podobny wynik można zaobserwować w przypadku największego spadku wydajności:

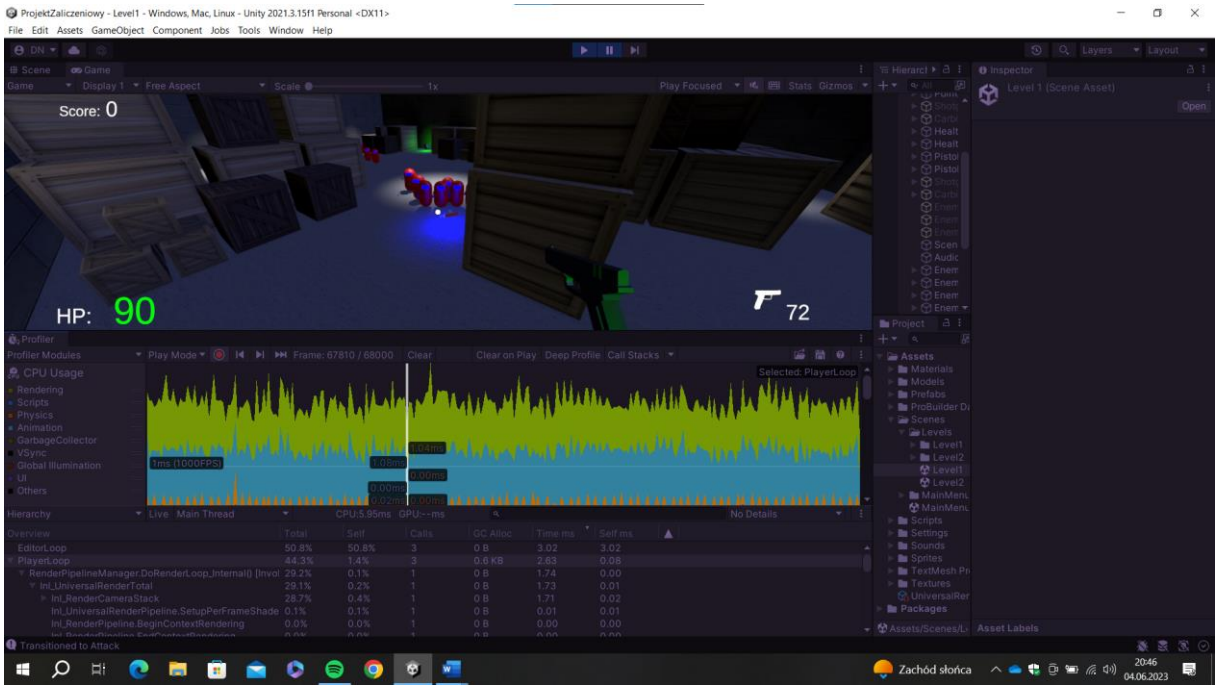

ProjektZaliczeniowy - MainMenu - Windows, Mac, Linux - Unity 2021.3.15f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
CAPSULE CITY BRAWL
Play
Credits
Controls
Exit
Hierarchy
Inspector
Project
Profiler
CPU Usage
20°C Cz. słonecznie
20:41
04.06.2023

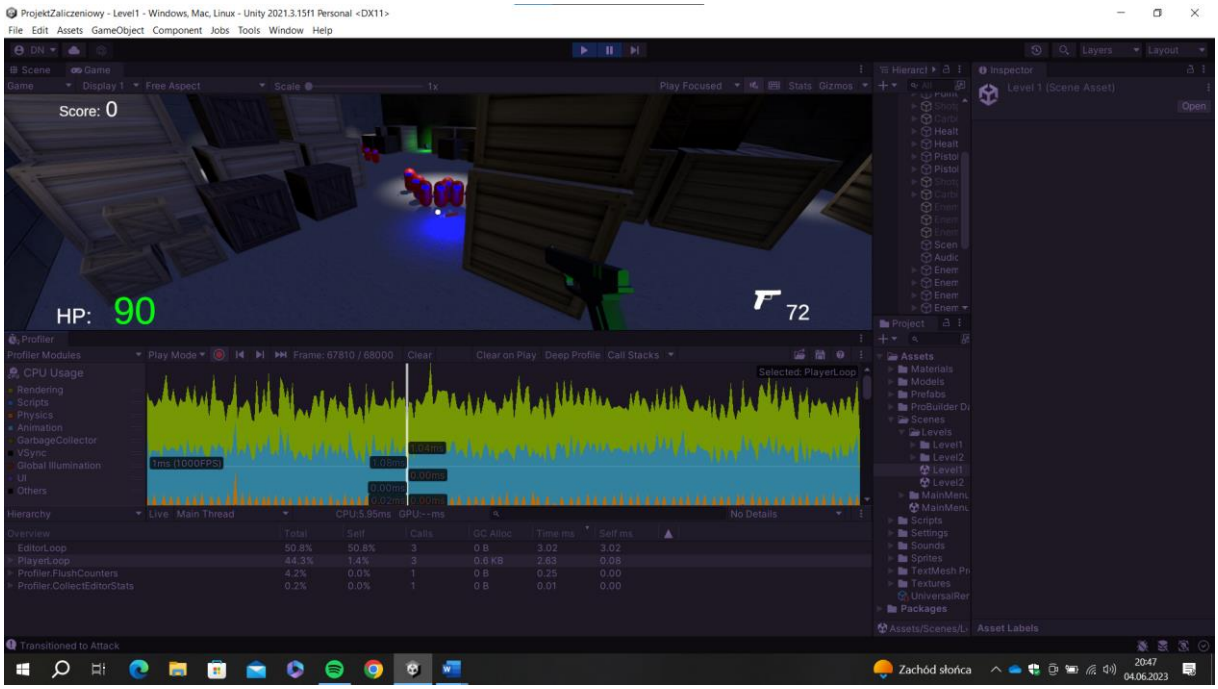

ProjektZaliczeniowy - MainMenu - Windows, Mac, Linux - Unity 2021.3.15f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
CAPSULE CITY BRAWL
Play
Credits
Controls
Exit
Hierarchy
Inspector
Project
Profiler
CPU Usage
Timeline
Main Thread
20°C Cz. słonecznie
20:41
04.06.2023

b) Level1

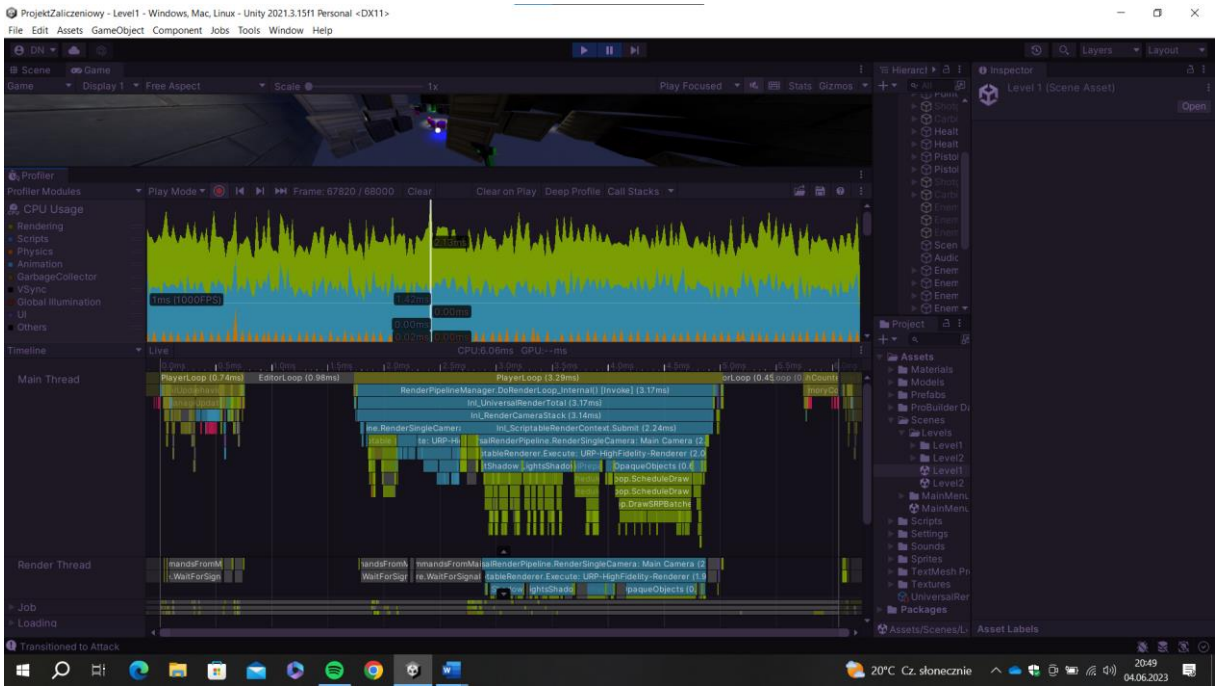
Profilowanie dla sceny Level1 wygląda następująco:

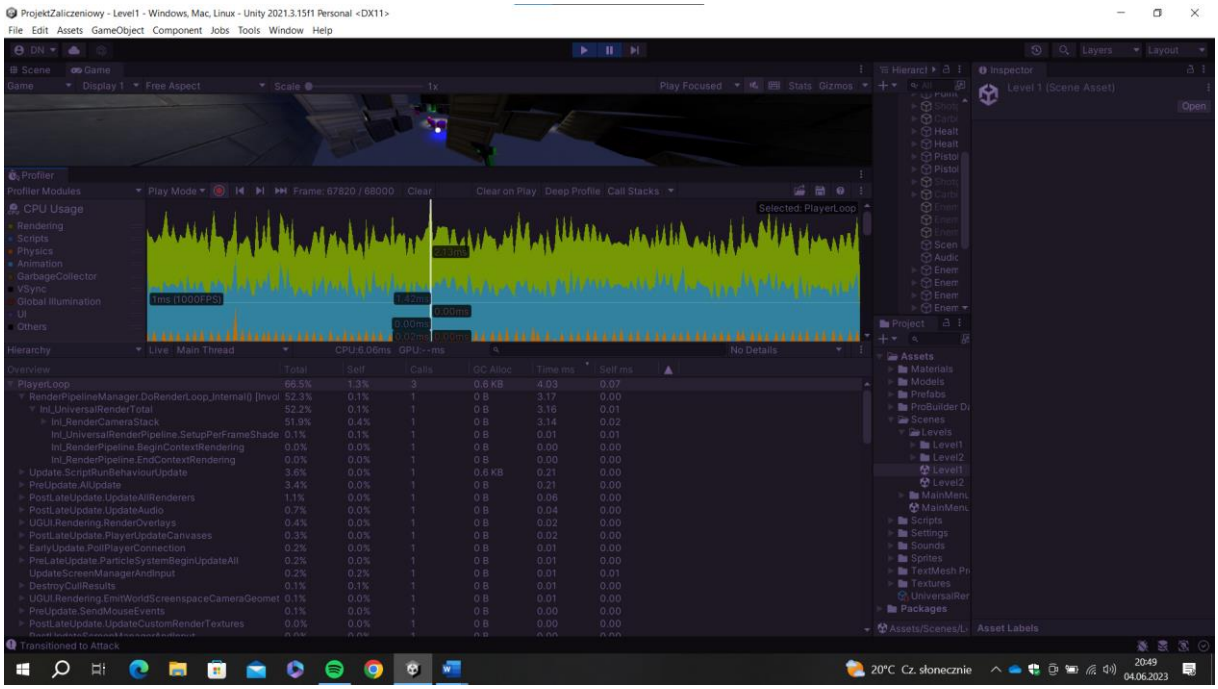


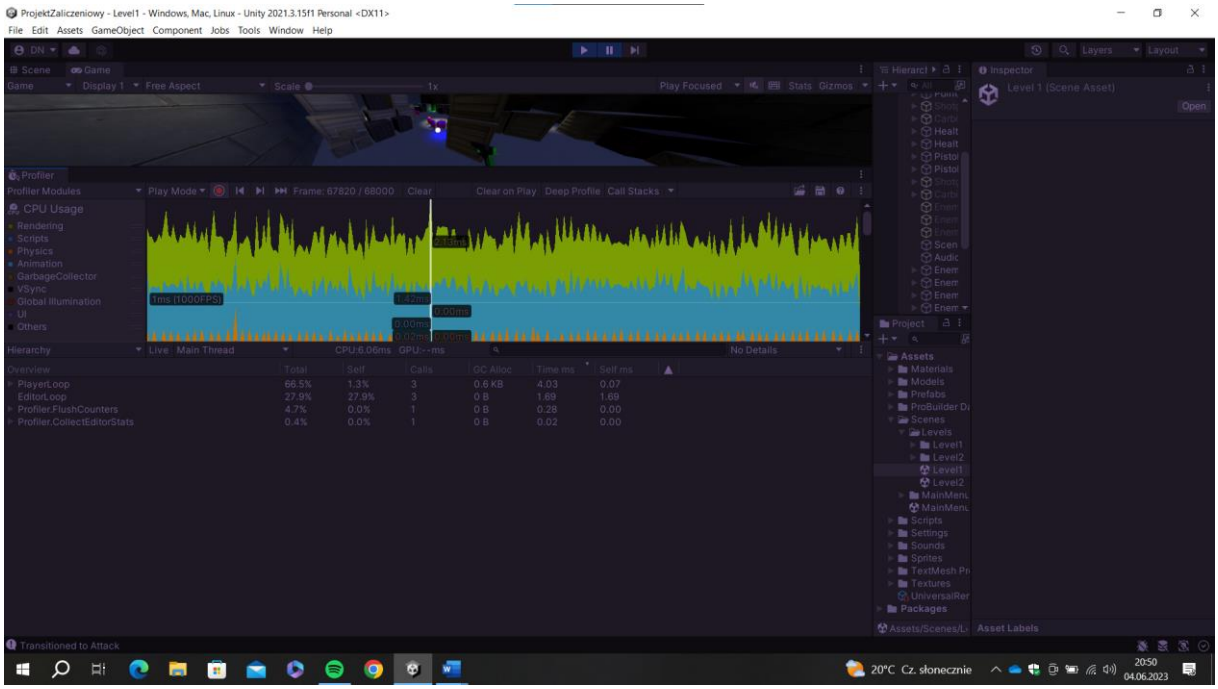




Ponownie winowajcą jest wspomniany wcześniej URP, choć w tym przykładzie więcej zasobów idzie na EditorLoop, niż na PlayerLoop. W najwyższym spike'u sytuacja przedstawia się następująco:



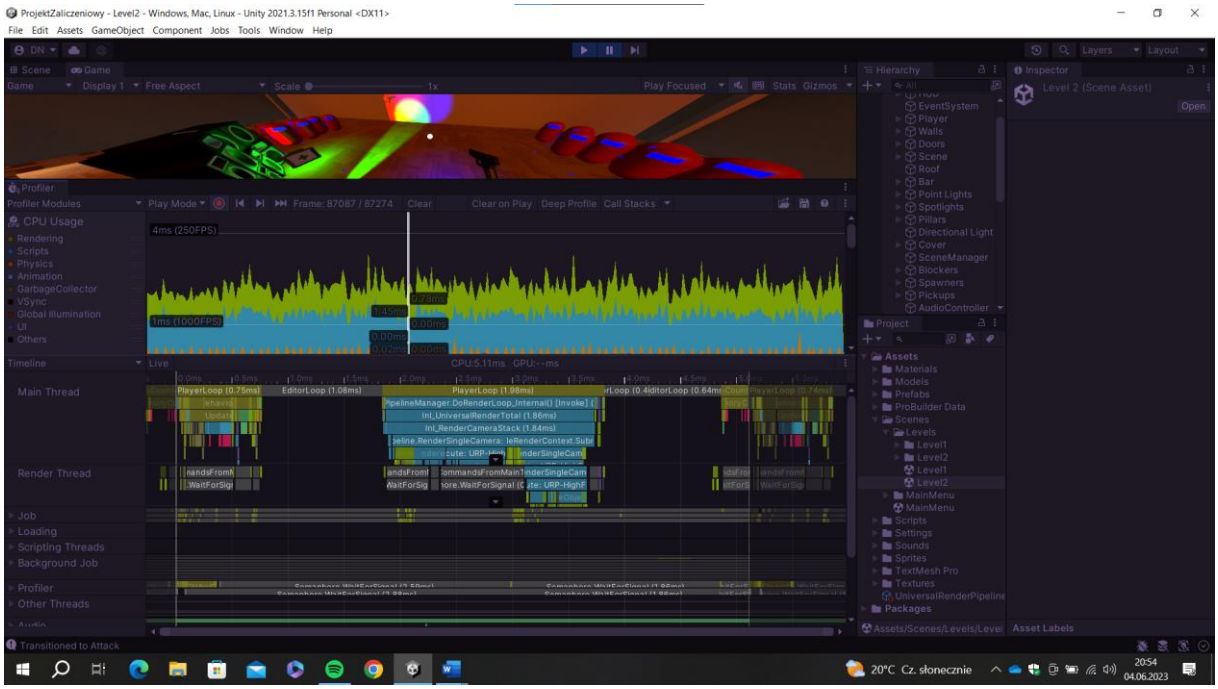


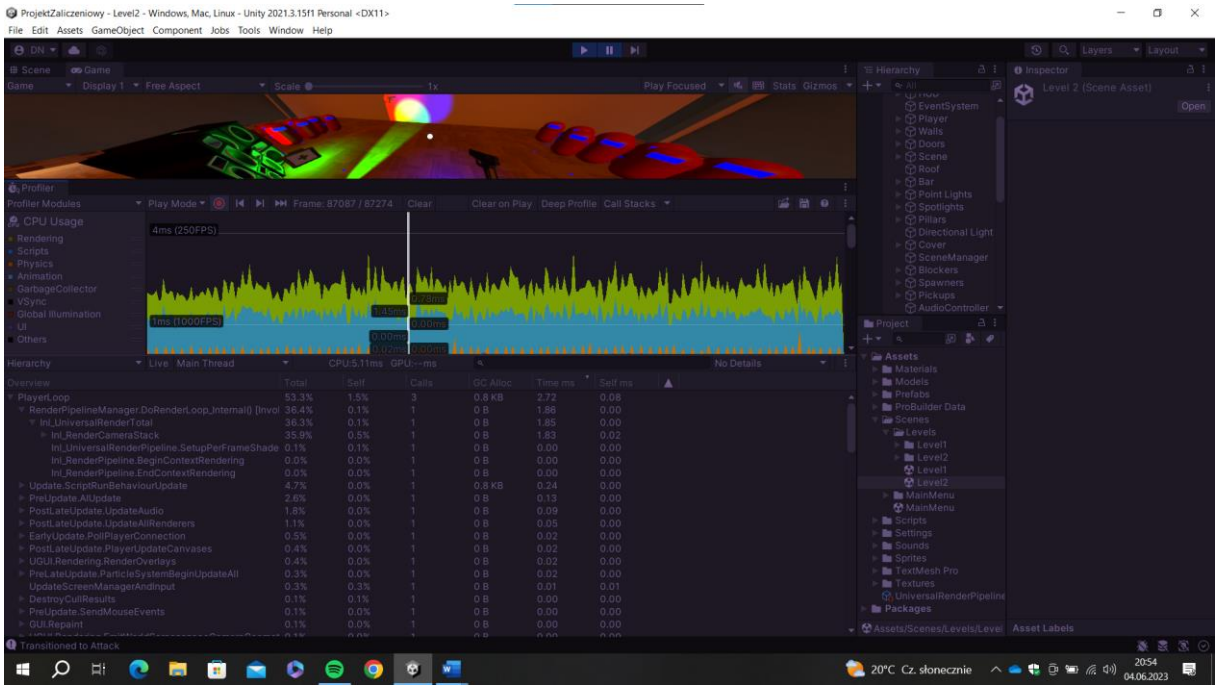


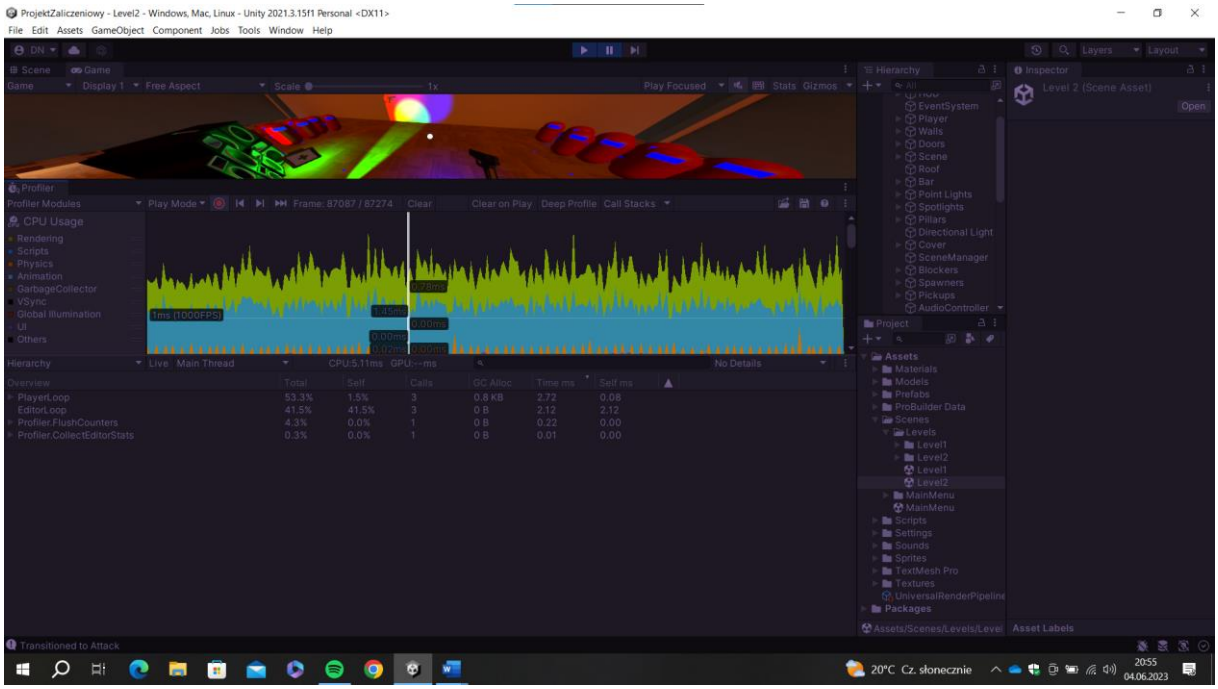
Ponownie, winowajca jest ten sam.

c) Level2

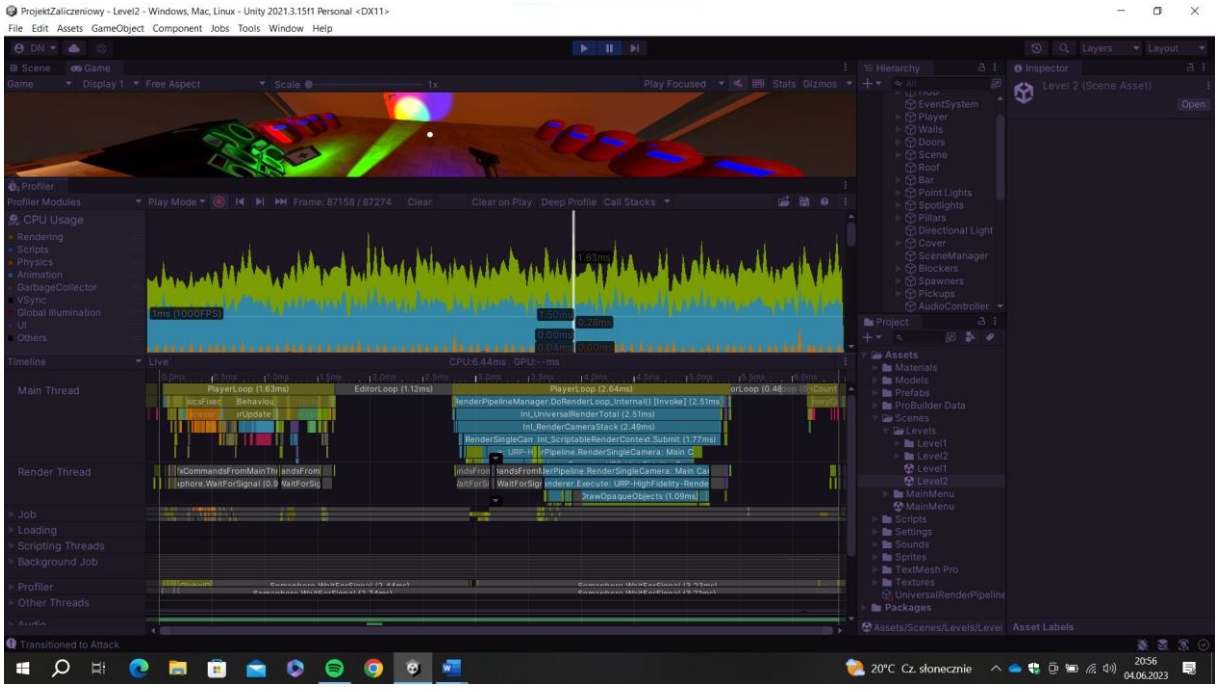
Wyniki profilowania trzeciej sceny wyglądają następująco:

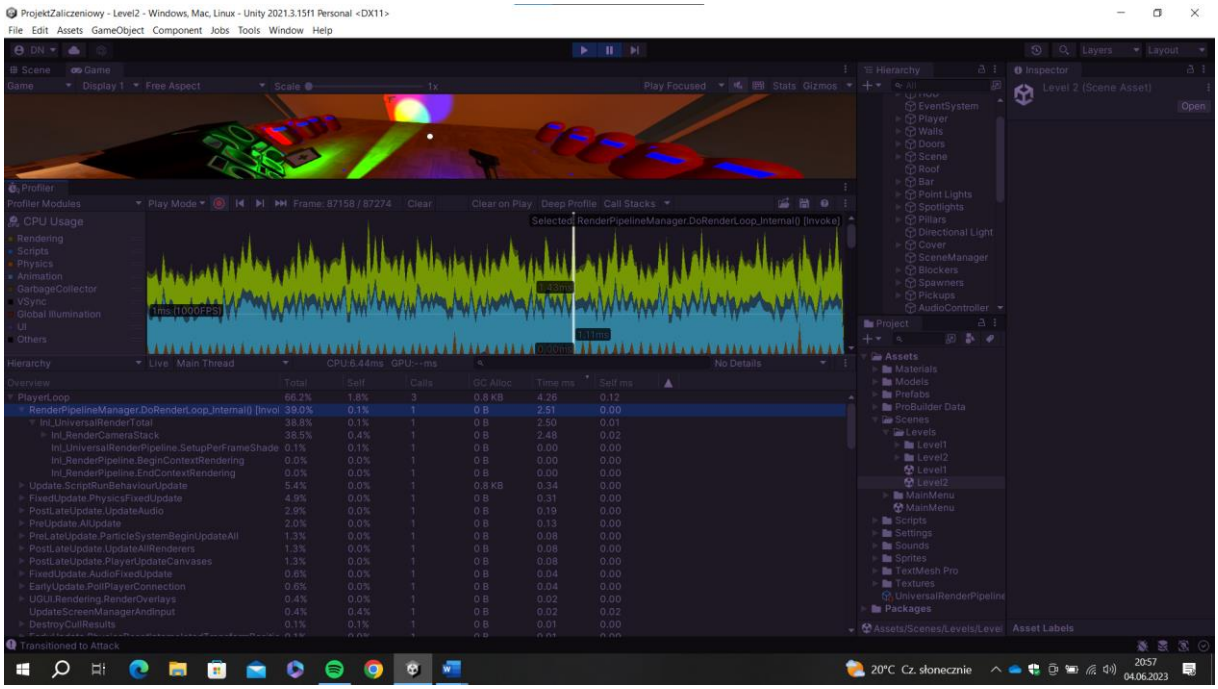

ProjektZaliczeniowy - Level2 - Windows, Mac, Linux - Unity 2021.3.15f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Scene
Game
Profiler
CPU Usage
Main Thread
Render Thread
Hierarchy
Inspector
Project
Level 2 (Scene Asset)
Open
Transitioned to Attack
20°C Cz. słonecznie
20:54
04.06.2023

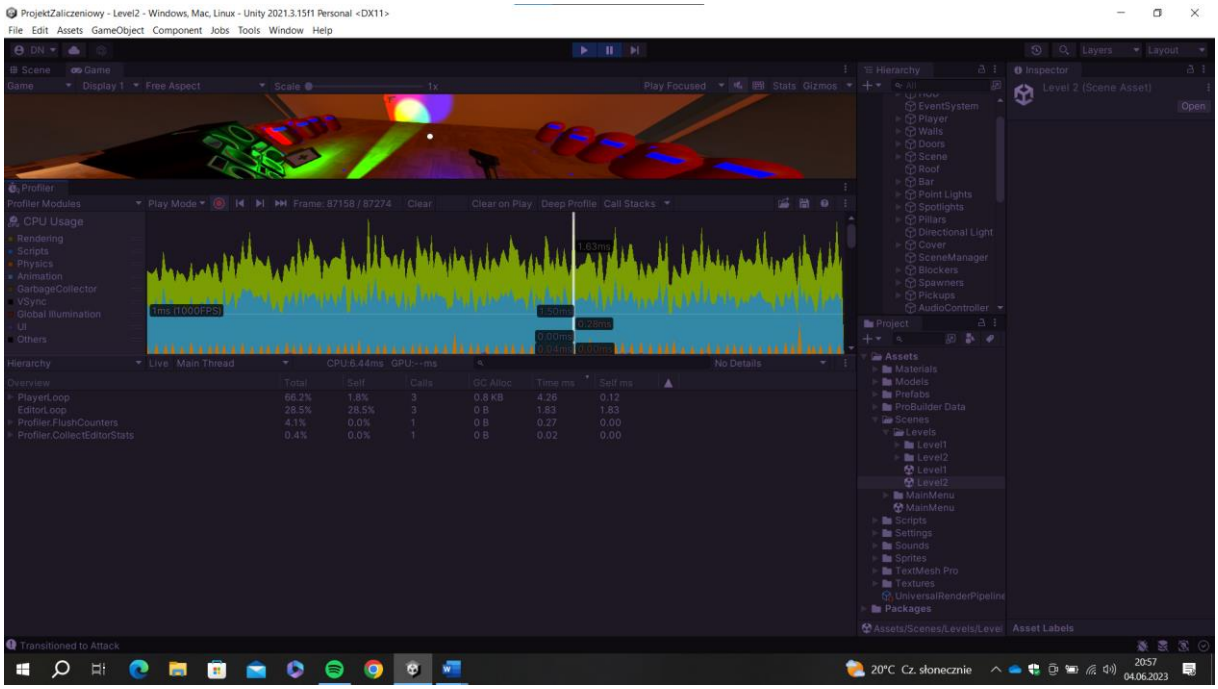

ProjektZaliczeniowy - Level2 - Windows, Mac, Linux - Unity 2021.3.15f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Scene
Game
Profiler
CPU Usage
Hierarchy
Inspector
Project
Level 2 (Scene Asset)
Open
Transitioned to Attack
20°C Cz. słonecznie
20:54
04.06.2023



Wniosek jest ten sam – największym zjadaczem wydajności w grze jest URP. Podobne rezultaty znajdą się w największym spike’u



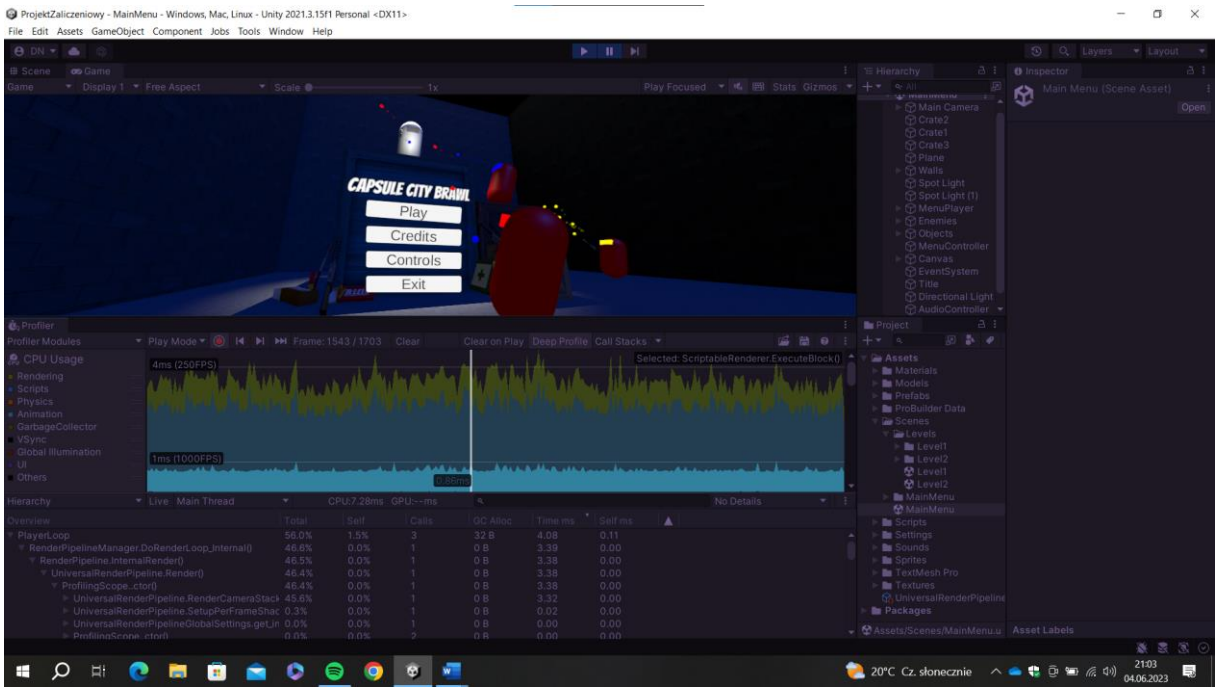


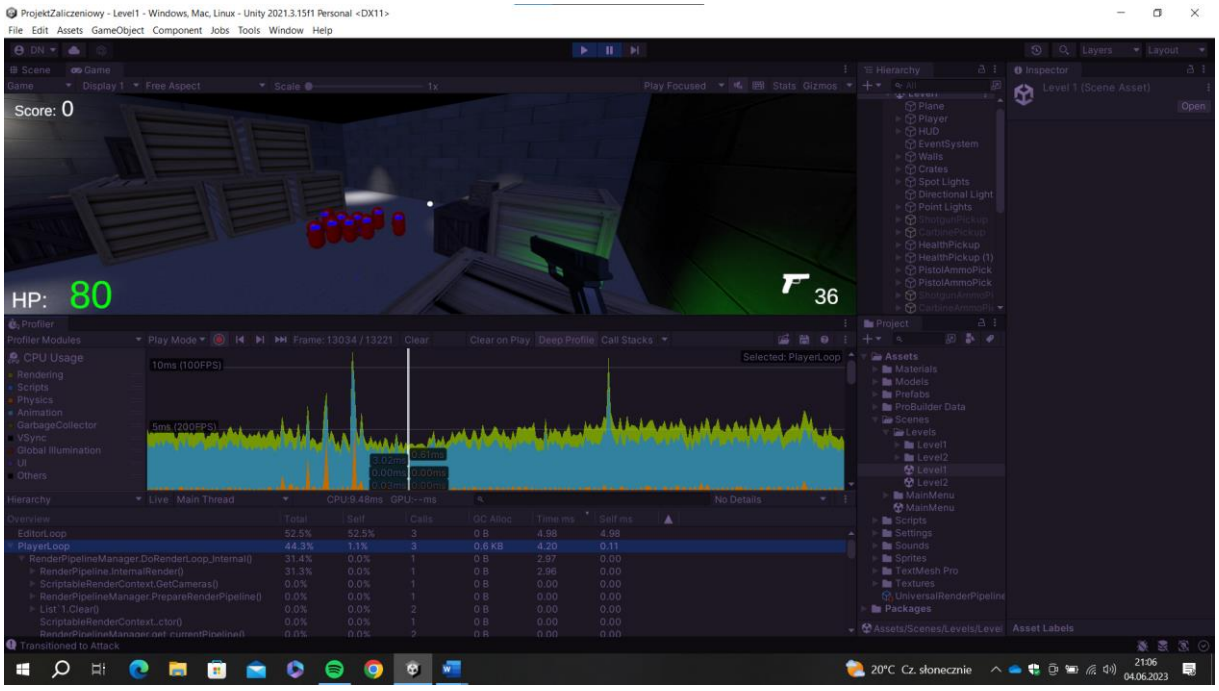


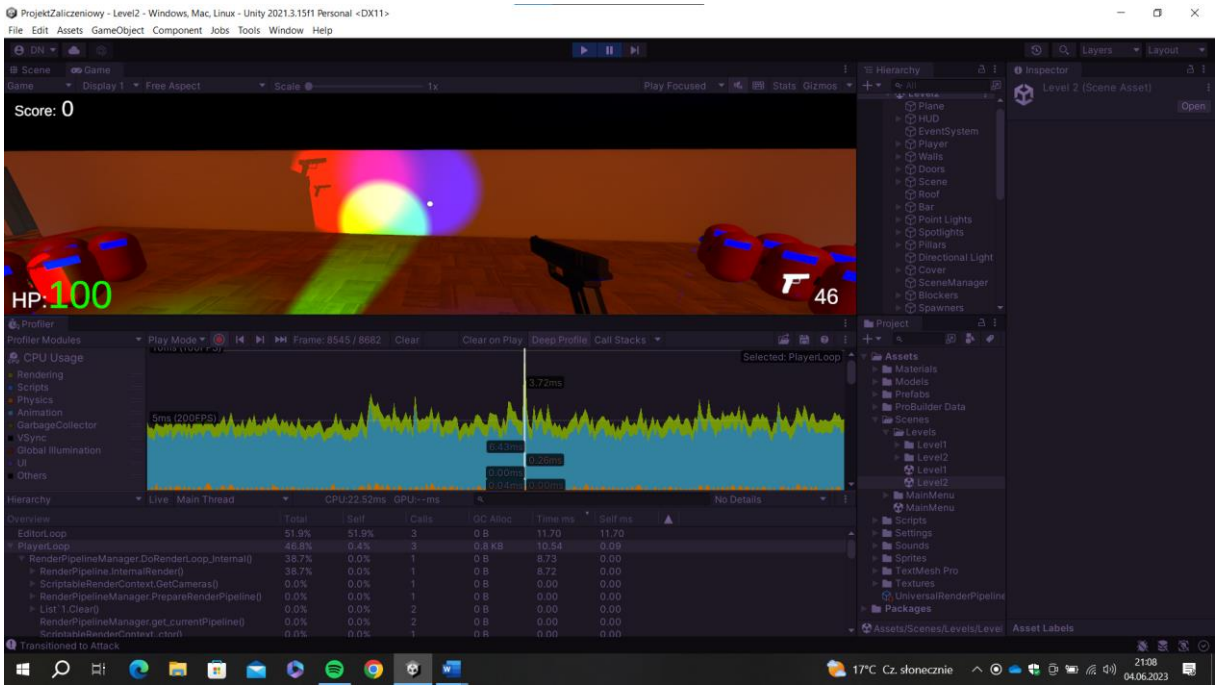
Jakie więc można wyciągnąć wnioski nt. optymalizacji gry? Otóż aby zwiększyć potencjalną liczbę klatek na sekundę, musielibyśmy zmienić ustawienia używanego render pipline. Innym sposobem na zwiększenie ilości klatek na sekundę mogłoby być ograniczenie ilości dynamicznego oświetlenia w scenach, gdzie jest to możliwe. Drugim najwięcej zużywającym zasoby mogłaby być funkcja Update – możliwym rozwiązaniem może być usunięcie pustych updatów lub przeniesienie kodu, który nie musi być wykonywany co klatkę w inne miejsce.

4. Deep Profile

Postanowiłem jeszcze testowo włączyć Deep Profile dla każdej ze scen, kolejność taka sama, jak poprzednio:



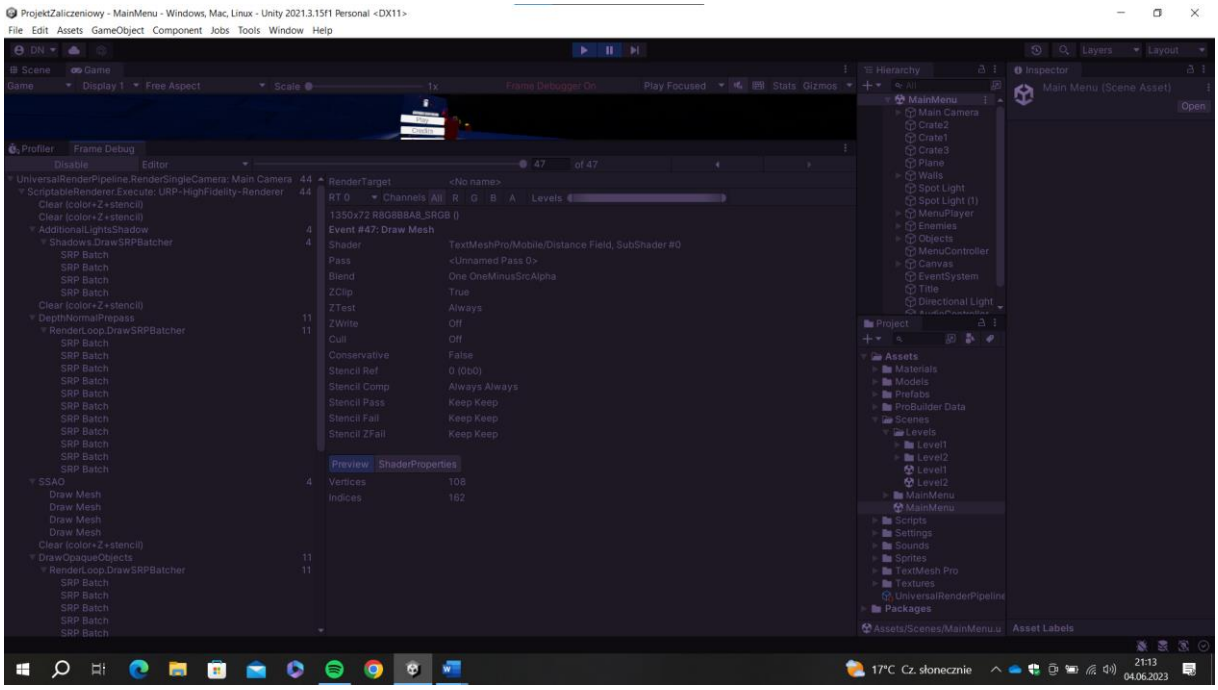


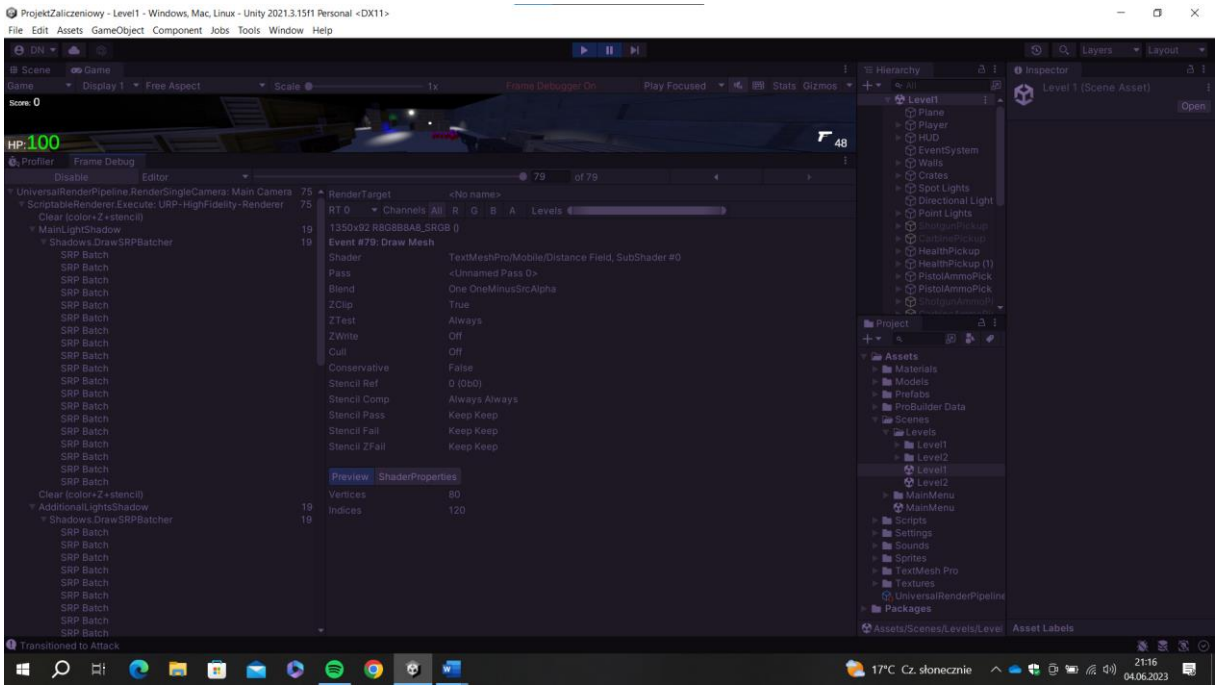


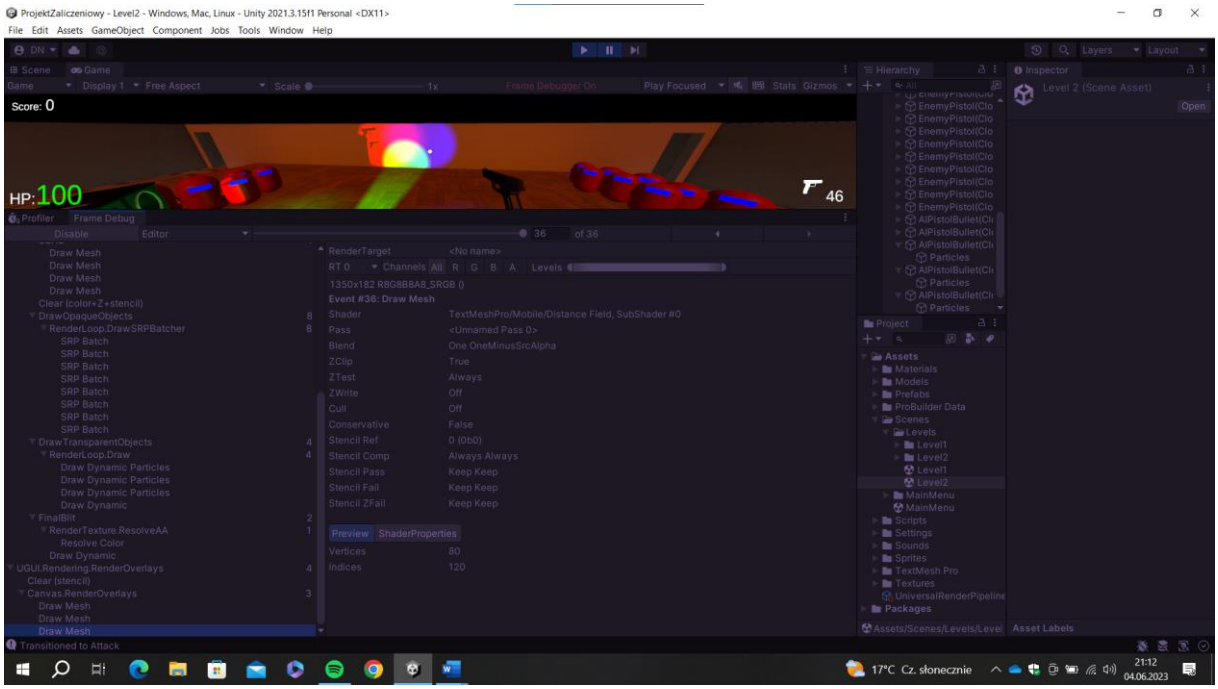
Są identyczne.

5. Frame Debugger

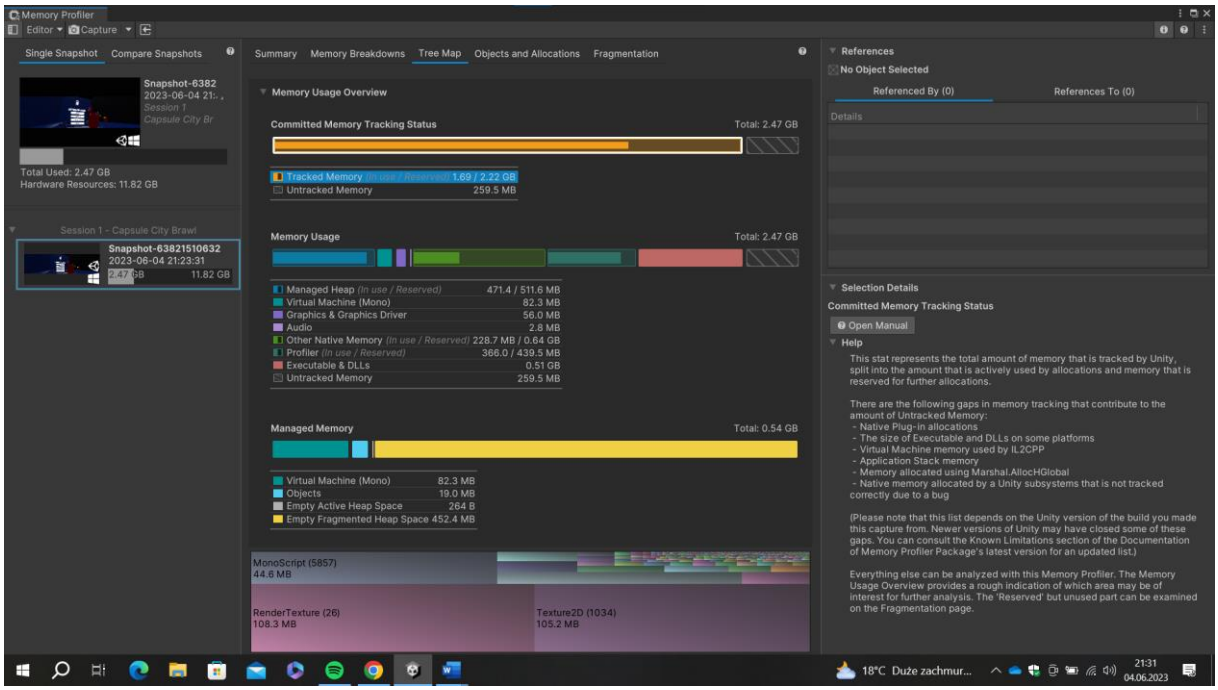
Kolejność jest taka sama:

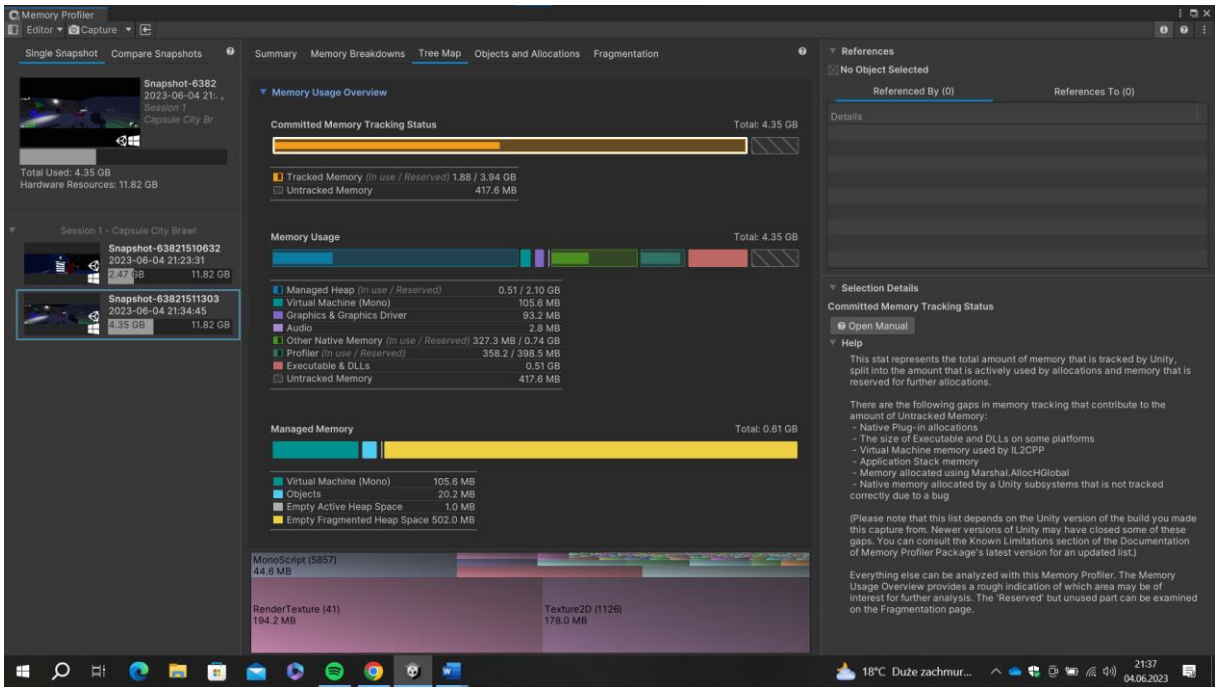


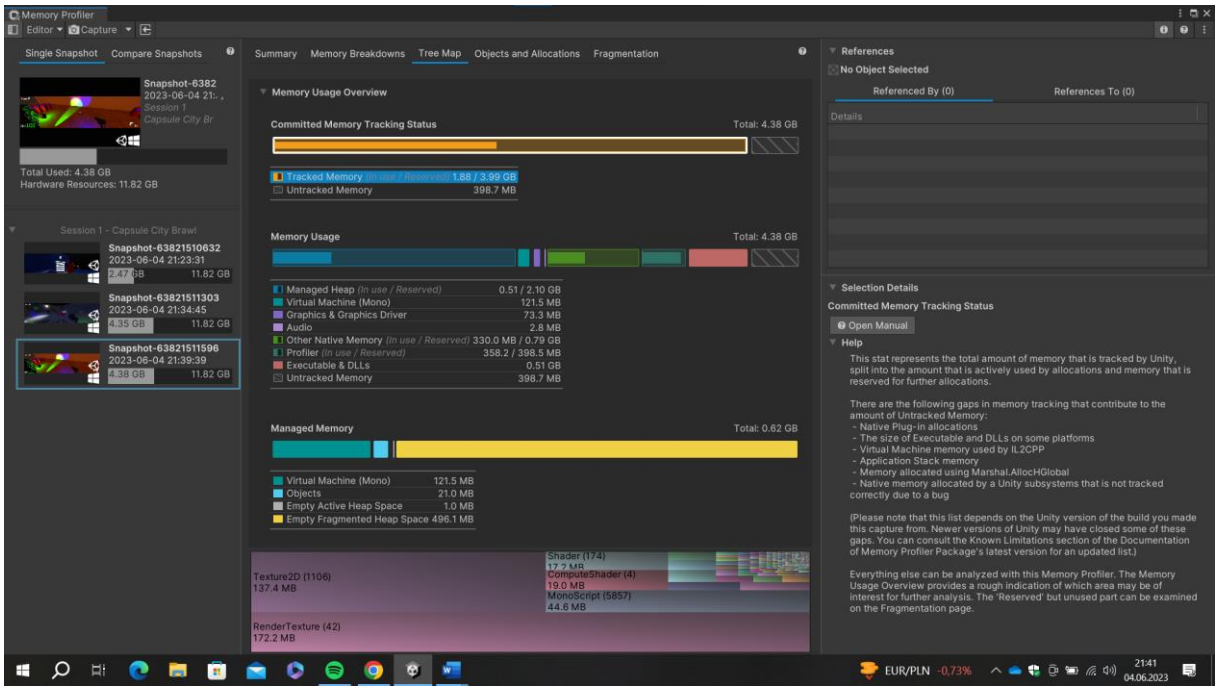

ProjektZaliczeniowy - Level1 - Windows, Mac, Linux - Unity 2021.3.15f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Score: 0
HP:100
48
Transitioned to Attack
17°C Cz. słonecznie
21:16
04.06.2023


ProjektZaliczeniowy - Level2 - Windows, Mac, Linux - Unity 2021.3.15f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Score: 0
HP:100
46
Transitioned to Attack
17°C Cz. słonecznie
21:12
04.06.2023

6. Zrzuty pamięci







Wyniki przedstawiają, że w menu gra alokuje około 2,5 GB pamięci RAM, z kolei w trakcie rozgrywki – ok. 4,5 GB. Przy czym sama pamięć używana to 1,89GB. Może to wynikać z sposobu, w jaki pozbywam się pocisków oraz wrogów z mapy – są oni po prostu usuwani za pomocą funkcji Destroy(). Mógłbym to zoptymalizować korzystając np. z object pooling'u dla spawnujących się wrogów oraz wystrzeliwanych pocisków.