Список використаних джерел

1. Ruby Annette J, Aisha Banu W, Subash Chandran P. Rendering-as-a-Service: Taxonomy and Comparison. Procedia Computer Science. 2015 [cited 2024 Feb. 28]. Volume 50. Pages 276-281. ISSN 1877-0509. Available from: https://www.sciencedirect.com/science/article/pii/S1877050915005499 doi: https://doi.org/10.1016/j.procs.2015.04.048.
2. Хорольська К. Аналіз основних підходів до вирішення задачі конвертації двовимірних зображень в тривимірну модель. Вісник Кременчуцького національного університету імені Михайла Остроградського. Кременчук: КрНУ, 2022 [цитовано 2024 Лют. 28]. Випуск 3(134). 105 с.
3. Weifeng L, Gong B, Hu Y. A large-scale rendering system based on hadoop. 6th International Conference on Pervasive Computing and Applications; 2011; Port Elizabeth; [cited 2024 Feb. 28] pp. 470-475. Available from: https://ieeexplore.ieee.org/document/6106549 doi: 10.1109/ICPCA.2011.6106549.

УДК 004.4

ОПТИМІЗАЦІЯ РЕСУРСІВ ПРОЄКТУ В UNITY

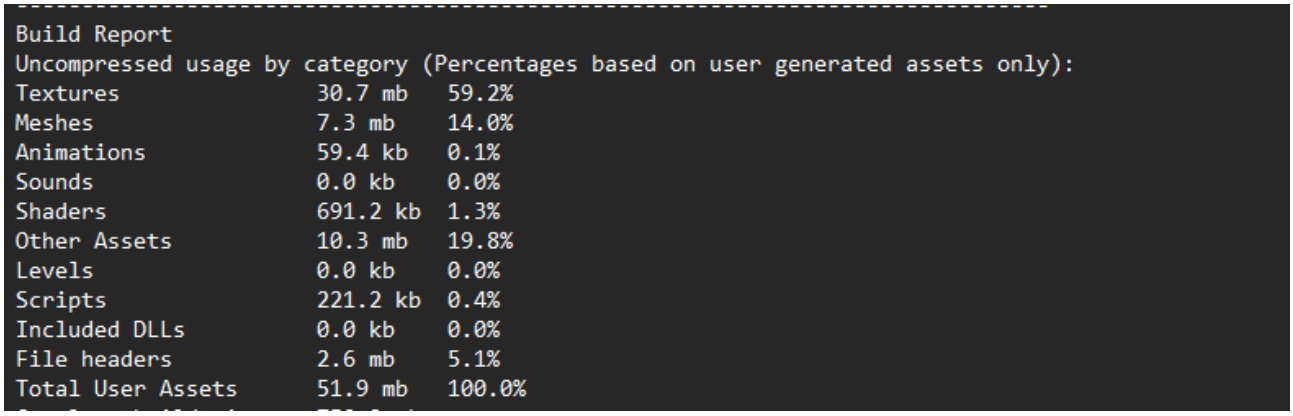
Довда Н.О., студент, zoomguru1@gmail.com, НДУ ім. М. Гоголя
Лисенко І.М., к.ф.-м.н., доцент, glushkoim@gmail.com НДУ ім. М. Гоголя

У наші дні Unity є найпопулярнішим ігровим рушієм для розробки мобільних застосунків для Google Play та App Store [1].

Оптимізація є одним з важливих критеріїв для успіху проєкту. Якщо велике споживання пам'яті у комп'ютерних проєктах зазвичай не створює проблем для споживачів, то при розробці мобільних ігор потрібно приділяти особливо увагу загальному кінцевому розміру самого застосунку.

Зазвичай, споживачі мають обмежені ресурси внутрішньої пам'яті. Користувачі часто вважають, що великі застосунки вимагатимуть зависокі системні потреби для свого використання. Надмірне споживання пам'яті може спонукати до швидкого видалення такого застосунку.

Після успішного компілювання застосунку, в логах Unity можна дослідити інформацію про займану пам'ять готового продукту (див. рис. 1). Дослідження проводилося на основі розробленого проєкту Castle Demolish [2].



```
Build Report
Uncompressed usage by category (Percentages based on user generated assets only):
Textures                30.7 mb   59.2%
Meshes                  7.3 mb    14.0%
Animations              59.4 kb   0.1%
Sounds                  0.0 kb    0.0%
Shaders                 691.2 kb  1.3%
Other Assets            10.3 mb   19.8%
Levels                  0.0 kb    0.0%
Scripts                 221.2 kb  0.4%
Included DLLs           0.0 kb    0.0%
File headers            2.6 mb    5.1%
Total User Assets       51.9 mb   100.0%
```

Рисунок 1 – Загальна інформація займаного обсягу скомпільованого застосунку

Розглянувши дані можна побачити, що велика частина обсягу кінцевої версії складається з текстур та мешів (анг. mesh). Після вдалого імпорту будь-яких ресурсів в Unity завжди потрібно їх правильно налаштовувати [3]. Не налаштовані ресурси в проєкті, можуть мати неочікуваний вплив на розмір кінцевої версії гри. Оскільки, дуже часто бувають випадки коли, наприклад, імпортоване зображення може за замовчуванням мати найвищу якість налаштування та при цьому, в самій грі гравець може зовсім не звертати уваги на цей об’єкт.

Текстура – це звичайне зображення, яке наноситься поверх мешу, тобто змодельованої моделі. Вона додає до неї колір, рельєф та інші властивості, які роблять об’єкт більш реалістичним та деталізованим. В Unity можна імпортувати текстури з найбільш популярних форматів зображення, наприклад .jpg, .png та інші.

До того ж, текстури застосовуються до об’єктів використовуючи матеріали, тобто компонент, який, в свою чергу, визначає як повинна відображатися поверхня на 3D-об’єкті. Матеріали використовують шейдери, тобто спеціалізовані графічні програми, які можуть відобразити текстури на поверхні 3D-об’єкту. Unity має декілька параметрів, які є зручними у налаштуванні та можуть здійснювати сильний вплив на оптимізацію імпортованих текстур.

Найвпливовішим з усіх є Max Size. За замовчуванням, будь-яка імпортована текстура має значення 2048×2048 пікселів та може бути налаштована від 32×32 до 16384×16384 пікселів. Варто зауважити, що потрібно бути уважним підбираючи оптимальне значення для Max Size, щоб випадково не знизити якість відображення об’єктів на сцені, які використовують цю текстуру.

Наступний параметр це формат стиснення Format. Параметр Format може мати різні значення в залежності від платформи під яку розробляється проєкт. Загалом, потрібно використовувати стиснені формати DXT або PVRTC, там де це можливо.

Останні дві опції, які впливають на оптимізацію це Read/Write Enabled та Mip Maps. Перший параметр створює копію текстури для обробки як центральним, так і графічним процесором, тим самим подвоюючи обсяг пам'яті самої текстури. У більшості випадків його можна вимкнути, оскільки для створення текстури під час гри можна скористатися функцією Texture2D.Apply. Другий параметр не потрібний, якщо ця текстура буде використовуватися при оформлені користувацького інтерфейсу, оскільки він, як правило, ніяк не змінює свою відстань від камери.

Додатково Unity має можливість створення «атласів текстур» – розміщення кількох текстур в одній. Використання атласів текстур може зменшити кількість викликів «draw calls» та прискорити процес рендерингу.

Меш – основний графічний примітив в Unity, який складає значну частини 3D-світів. Unity підтримує трикутні, чотирикутні та багатокутні меші. Зазвичай, Unity створює меш під час імпорту його моделі, але його також можна створювати безпосередньо в Unity. Так само, як і текстури, меші можуть споживати багато надмірної пам'яті, якщо вони будуть неправильно налаштовані.

Меші також можна стискати. На відміну від текстур, вони мають спеціальний параметр Mesh Comprassion. Mesh Comprassion є найвпливовішим параметром для оптимізації мешу. Важливо зауважити, що результат стискання мешу впливає на його відображення в ігровій сцені, тому потрібно експериментувати з рівнями компресії та знайти оптимальний. Також, тут присутній параметр Read/Write Enabled, який виконує такі ж самі функції як і з текстурами.

Крім того, у вікні налаштуваннях мешу присутні ще три параметра, на які варто звернути увагу, а саме Rigs, Normals та Tangents. Rigs відповідає за можливість мешу до скелетної або BlendShapes анімації. Інші два вимикати лише тоді, якщо матеріал самого мешу не буде їх використовувати.

Загалом, це всі перелічені параметри на які варто звертати увагу при імпорті мешу. Хочемо зазначити, що основний компонент, який відповідає за потенційний обсяг займаної пам'яті мешу – це його роздільна здатність. Варто взяти це до уваги, оскільки, навіть геометрія заднього фону у грі може займати навіть півмільйона полігонів.

Попередньо застосувавши зазначені оптимізаційні методи, проєкт було скомпільовано знову, щоб порівняти результати займаного обсягу пам'яті (див. рис. 2).

```
Build Report
Uncompressed usage by category (Percentages based on user generated assets only):
Textures                11.7 mb   39.8%
Meshes                  3.8 mb    13.0%
Animations              59.4 kb   0.2%
Sounds                  0.0 kb    0.0%
Shaders                 691.2 kb  2.3%
Other Assets            10.3 mb   35.0%
Levels                  0.0 kb    0.0%
Scripts                 221.2 kb  0.7%
Included DLLs           0.0 kb    0.0%
File headers            2.6 mb    9.0%
Total User Assets       29.4 mb   100.0%
```

Рисунок 2 – Загальна інформація займаного обсягу попередньо оптимізованого застосунку

Порівнюючи дані на рис. 1 та рис. 2 бачимо, що загальний розмір всіх ресурсів проєкту тепер сягає 29.4 МБ, замість 51.2 МБ.

Висновки. У результаті проведеного дослідження підтверджено вплив оптимізації ресурсів проєкту на займаний обсяг пам’яті кінцевої версії застосунку. Доведено, що основними чинниками споживання пам’яті проєкту являються текстури та меші, тому їх оптимізація може значно скоротити споживання пам’яті застосунком.

Список використаних джерел

1. Офіційний сайт Unity [цитовано 09 лют. 2024]. Доступно на: https://unity.com/
2. Проєкт Castle Demolish [цитовано 09 лют. 2024]. Доступно на: https://play.google.com/store/apps/details?id=com.LuB.CastleDemolish&hl=ru&gl=US
3. Unity’s expert team Optimize your game performance for mobile №10207; 2023: 24-27 [цитовано 08 лют. 2024]. Доступно на: unity-e-book-optimize-your-mobile-game-performance