

Технічні науки

УДК 004.416.2

Кучеренко Алевтина Олександрівна

студентка

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

ОПТИМІЗАЦІЯ ПАМ'ЯТІ ТА ОПТИМІЗАЦІЇ ПРОДУКТИВНОСТІ WEBGL

Анотація. Проведено аналіз методів вирішення поширених проблем пов'язаних з продуктивністю WebGL. Сформульовано основні підходи у використанні пам'яті, роботі з активами та оптимізації розміру побудованого WebGL-додатку.

Ключові слова: WebGL, Unity, JavaScript, програвач, розподіл пам'яті.

Багато розробників Unity почали експортувати версії WebGL своїх ігор. На жаль, багато з них також стикаються з проблемами, які можуть призвести до погіршення продуктивності та нестабільності. Це особливо складна проблема на старих комп'ютерах, які мають менше ресурсів.

У цій статті викладено декілька методів вирішення деяких поширених проблем пов'язаних з продуктивністю WebGL, які можливо побачити в іграх, опублікованих на Kongregate.

Виміряйте двічі, оптимізуйте один раз.

Одна з найпоширеніших помилок, яку розробники роблять, намагаючись оптимізувати вміст - це припущення про те, що займає багато часу або місця. З Unity та WebGL це трапляється часто, коли вирішується, скільки місця виділити для програми WebGL у налаштуваннях програвача.

Не роблячи досліджень з цього питання, можна легко припустити, що збільшення пам'яті, доступної для програми, призведе до кращої продуктивності. Насправді розробники, як правило, хочуть виділити найменший об'єм пам'яті, що може суперечити інтуїції чи здоровому глузду, але часто бути все-таки правдою.

Зменшення розміру пам'яті може значно зменшити чи виключити кількість помилок "Веб-браузер не може виділити достатню кількість пам'яті для вмісту WebGL". На щастя, розробник може отримати досить гарне враження про те, скільки пам'яті використовує ваша програма за допомогою простого фрагмента JavaScript, який можна вставити в консоль браузера, або як постійне доповнення до шаблону index.html. За допомогою TOTAL_MEMORY змінної та функції setInterval, можливо створити додаток який буде виводити обсяг використаної та вільної пам'яті в консоль кожні 5 секунд, наприклад: Статистика пам'яті - використовується: 155М вільно: 37М. Завдяки цьому дуже простому доповненню, тепер є деяке розуміння того, як налаштувати розмір пам'яті, що є фантастичним. Розробник можете дивитися на ці данні під час гри, і розуміти, на що слід обмежити використання пам'яті. Але треба бути обережним, щоб не знижувати її занадто сильно, інакше можна почати бачити помилки в розподілі пам'яті, які підказують збільшити її розмір.

Примітка щодо динамічного розміру купи

Купа в інформатиці та програмуванні — назва структури даних, за допомогою якої реалізована динамічно розподілювана пам'ять програми.

Шукаючи вирішення проблем із пам'яттю, розробник можете натрапити на прапор Emscripten під назвою ALLOW_MEMORY_GROWTH, який звучить як ідеальне рішення ваших негараздів. Увімкнення цього прапора може негативно впливати на продуктивність, а також може викликати проблеми із пам'яттю, оскільки менеджер пам'яті повинен виділити ще більше місця при спробі

збільшення купи. Команда Unity ніколи не рекомендувала вмикати цей прапор.

Згладжування початкового завантаження

При використанні Unity 5.3 і вище, ігрові активи за замовчуванням будуть стискатись gzip і матимуть префікс gz у назві файлів. Як описано в публікації блогу WebGL, якщо завантажити їх у Kongregate, він автоматично встановимо заголовок Content-Encoding: gzip, щоб активи можна було декомпресувати браузером замість JavaScript, що призводить до набагато кращого досвіду завантаження.

Однак якщо ви розміщуєте свою гру зовнішньо, вам потрібно буде вручну переконатися, що активи обслуговуються правильним кодуванням. Ви дізнаєтесь, чи це є проблемою, тому що Unity буде записувати такі повідомлення, як Decompression Release / MyGame-WebGL.jsgz, кожні 500 мілісекунд в консолі, а в браузері може з'явитися помилка під час завантаження, і панель прогресу не буде працювати належним чином.

Якщо ви використовуєте apache або сумісний сервер, ви можете використовувати файл .htaccess, який створює Unity для вирішення цієї проблеми. Однак якщо ви розміщуєте хостинг на S3 або щось подібне, ви можете увімкнути кодування, змінивши імена файлів у вашому index.html(рис.1), щоб додати суфікс gz, наприклад:

```
var Module = {  
  TOTAL_MEMORY: 201326592,  
  errorHandler: null,  
  compatibilitycheck: null,  
  dataUrl: "Release/MyGame-WebGL.datagz",  
  codeUrl: "Release/MyGame-WebGL.jsgz",  
  memUrl: "Release/MyGame-WebGL.memgz",  
};
```

Рис. 1. Приклад найменування файлів[3]

Вам також потрібно буде переконатися, що S3 налаштований для надсилання заголовка Content-Encoding: gzip для кожного з файлів, які

закінчуються на gz, або ви отримаєте помилки декомпресії при завантаженні гри. Ви можете скористатися інспектором мережі браузера, щоб переконатися, що сервер надсилає заголовок належним чином.

Оптимізація завантаження активів

Ще одне несподіване джерело проблем із ефективністю викликано завантаженням пакетів активів. Для версій Unity нижче 5.5 пакети активів за замовчуванням стискаються за допомогою LZMA. Декомпресія вмісту LZMA для побудови WebGL викликає дуже помітну паузу. Ці паузи не лише змушують вашу гру працювати дуже погано, але й можуть викликати діалог помилок "цей ресурс не відповідає". На щастя, Unity 5.3 і вище підтримують швидший формат LZ4 для пакетів активів (а для WebGL це стандарт з версії Unity 5.5), тому ви можете вирішити цю проблему порівняно легко.

Слід зазначити, що активи LZ4 можуть бути трохи більшими, ніж LZMA, але компроміс, як правило, прийнятний. Якщо це не так, ви можете використовувати компресію gzip поверх пакетів активів і подавати їх заголовок Content-Encoding: gzip (як зазначено вище), щоб браузер автоматично розпакував їх.

Будьте в курсі оновлень Unity

Команда Unity часто випускає патчі для підтримуваних версій Unity. Слідкуйте за ними і переконайтесь, що будь-яка з змін, пов'язаних із WebGL, може вирішити проблеми, які виникають у вас.

Наприклад, патч Unity 5.3.6 вирішив проблему, коли для побудови WebGL було виділено набагато більше пам'яті, ніж потрібно. Це, безумовно, помилка, яку ви хочете виправити під час створення гри. Якщо оновлення не є життєздатним варіантом, вони також, як правило, охоплюють обхідні шляхи.

Висновок. В ході останніх тенденцій оптимізація по роботі з пам'яттю відіграє важливу роль у побудові WebGL-додатків. Пам'ять

користувача повинна використовуватися оптимально, а розміри активів повинні бути мінімальними. Все це дає змогу надати користувачу якісний продукт, та забезпечити стабільну роботу додатку.

Література

1. Кантор Д. WebGL для начинающих / Д. Кантор. 2012 С. 158-175.
2. Дикинсон К. Оптимизация игр в Unity 5. Советы и методы оптимизации игровых приложений / К. Дикинсон. 2017. С. 200-215.
3. Релізація найменування файлів. URL:
<https://blog.kongregate.com/tag/game-development/>