

УДК 004.514

ОСОБЛИВОСТІ РОЗРОБКИ ГРИ ДЛЯ МОБІЛЬНОГО ПРИСТРОЮ

*Гаманець А.О., студент, кафедра МСТ, ХНУРЕ,
Кулішова Н.Є., к.т.н., професор, кафедра МСТ, ХНУРЕ*

Анотація. Завдяки активному розвитку смартфонів, ринок мобільних розваг зростає дуже швидко. Розробка ігор для операційних систем Android та iOS може бути дуже прибутковим заняттям, проте при розробці мобільних ігор виникає ряд вимог та обмежень, які пропонується розглянути у цій роботі.

Ключові слова: МОБІЛЬНИЙ ПРИСТРІЙ, МОБІЛЬНА ГРА, ТЕКСТУРА, ЗД МОДЕЛЬ.

Сучасні мобільні пристрої стали невід'ємною складовою нашого життя. Звичайний смартфон або планшет здатний підтримувати складну 3Д-графіку у реальному часі, анімації, доповнену реальність та ін., що забезпечує повноцінні враження від гри. Але, разом з невеликими розмірами екранів та обмеженою продуктивністю пристрою, з'являються і нюанси, які доводиться враховувати під час розробки ігор.

Наприклад, величезна різноманітність пристроїв веде до абсолютно різного співвідношення сторін і роздільної здатності екрана. Це змушує розробників враховувати всі найпопулярніші дозволи та адаптувати гру, інтерфейс та багато іншого відразу під кілька варіантів.

До основних проблем, які виникають при перегляді на різних пристроях можна віднести: зміну розмірів елементів інтерфейсу із зміною роздільної здатності (рис. 1), вихід функціональних елементів гри за межі екрану (рис. 2), зміну швидкості анімації.

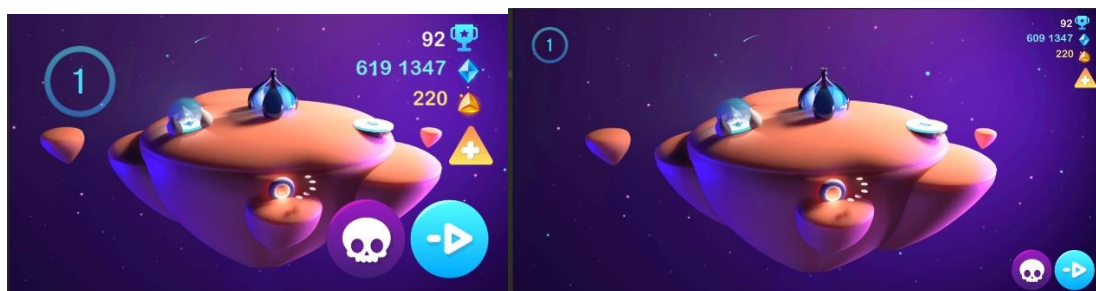


Рисунок 1 – Зміна розмірів елементів інтерфейсу



Рисунок 2 – Некоректне відтворення елементів гри в залежності від співвідношення сторін

Також проблеми виникають унаслідок апаратної складової. Широкі варіації потужностей не дають можливість забезпечити весь діапазон функцій на всіх девайсах. Через це доводиться враховувати і рівнятися на найслабші моделі, або робити кілька версій, що перемикаються в залежності від потужності пристрою, щоб не обділяти власників потужних смартфонів [1].

Різноманітність версій самої системи також ускладнює процес розробки – функції, які можна реалізувати на новіших версіях телефону, не можна відтворити на старих. І якщо стоїть мета охопити велику аудиторію, то доведеться мати справу із застарілими бібліотеками та зменшеною потужністю.

Для того, щоб забезпечити швидкодію пристрою та зменшити навантаження на апаратну складову існує багато підходів, але насамперед варто подбати про підготовку зображень (текстур) та 3Д моделей.

Вимоги до текстур. Найкраще працювати з квадратними текстурами. Якщо текстура прямокутна, вона розтягнеться до квадрата. Через особливості цифрових обчислень, дозвіл текстур завжди намагаються робити кратними ступеня двійки. Сторона текстури може бути 32 пікселя завширшки або 64, 128, 256, 512, 1024, 2048 і навіть 4096 пікселів [2], [4-6].

Текстури кратні ступені двійки заповнюють цілі осередки відеопам'яті. Якщо замість текстури 1024×1024 помістити у середовище розробки текстуру розміром 1000×1000 - вона не заповнить всю комірку відеопам'яті і її обробка вимагатиме більше ресурсів від відеокарти. Через таку дрібницю гра стане менш продуктивною.

Наприклад, у середовищі розробки комп'ютерних ігор Unity передбачена можливість вибору формату для зображення: automatic (сюди входить crunch compression), RGB 16 та 24 bit, RGBA 32 bit. Розглянемо особливості цих форматів на прикладі рис. 3.

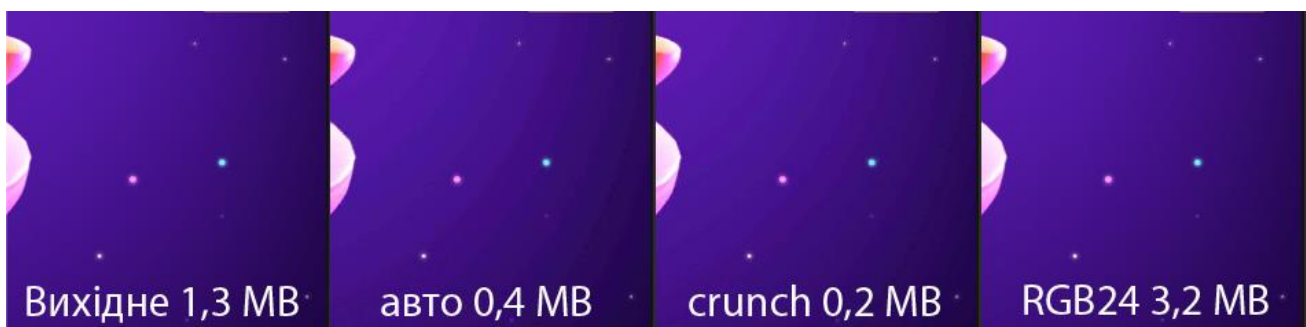


Рисунок 3 – Фрагменти зображення із різними видами стиснення

При використанні автоматичного стиснення значно зменшується розмір зображення, проте відтворюється погана плавність градацій, тому такий метод підходить для невеликих за розміром та «різких» зображень. Використання crunch compression допомагає досягти найменшого розміру, а регулювання сили компресії роблять цей метод стиснення оптимальним для більшості зображень.

Ідеальну плавність градацій передають лише формати RGB24 та 32 bit, проте при цьому збільшуючи розмір вихідного файлу майже втричі, що зумовлює використання цих форматів лише у виняткових випадках.

Вимоги до 3Д моделей. Кожний 3д-об'єкт складається із полігонів (вершин та граней). Враховуючи особливості мобільних пристроїв, кожна точку тривимірного об'єкту прораховують декілька десятків разів у секунду (зазвичай 60). Тож, основне завдання оптимізації полігональної 3Д моделі полягає у максимальному зменшенні кількості вершин та граней без втрати деталізації та плавності форм.

Крім безпосереднього зменшення кількості точок шляхом зміни полігональної сітки (ретопології), можна виділити ще й метод «відкидання» частини моделі, яку не бачить гравець (рис. 4).

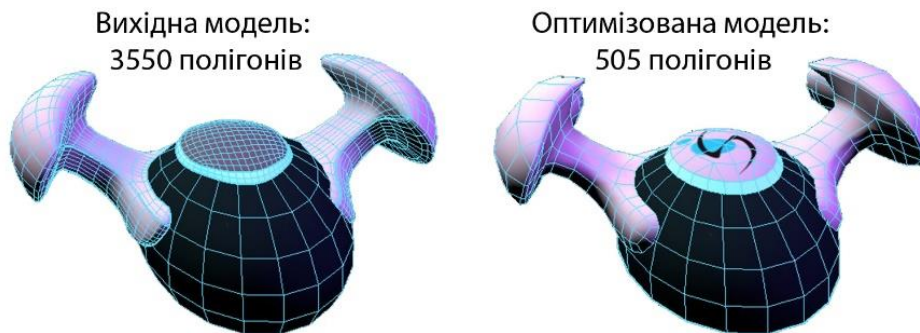


Рисунок 4 – Приклад підготовки 3Д моделі для мобільної гри

Таким чином, було розглянуто особливості розробки мобільної гри з урахуванням вимог до гнучкості інтерфейсу, які обумовлюються різними розмірами екранів, а також досліджено базові прийоми зменшення навантаження на пристрій за рахунок оптимізації текстур та 3Д моделей.

Література.

1. У чому складність розробки додатків для Android. <http://surl.li/bwgsi>.
2. UV розгортка AAA-пайплайн. <http://surl.li/bwgts>.
3. Розробка ігор на Android. <http://surl.li/bwgum>.
4. Kulishova, N., Paramonov, A., & Tkachenko, V. (2020, August). Real-Time Automatic Video Inspection System for Piece Products Marking. In 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) (pp. 169-173). IEEE.
5. Григор'єв О. В. Корекція колірного балансу цифрового зображення на основі статистичних характеристик / О. В. Григор'єв, Т. А. Колесникова, Л. О. Яценко // Поліграфічні, мультимедійні та web-технології: колективна монографія. – Харків: ТОВ «Друкарня Мадрид», 2021. – С. 68-79.
6. Deineko, Z., Zeleniy, O., Lyashenko, V., & Tabakova, I. (2021). Color space image as a factor in the choice of its processing technology. Abstracts of I International scientific-practical conference «Problems of modern science and practice» (September 21-24, 2021). Boston, USA, pp. 389-394.