

задача зводиться до знаходження найбільшої площі прямокутного шматка, який можна врятувати.

Ця задача має різні варіації, залежно від обмежень та умов, наприклад, можуть бути задані обмеження на розмір прямокутника, можуть бути враховані діагональні займані клітини, або можуть бути додаткові обмеження на форму або взаємне розташування прямокутників.

Задача про найбільшу клітину на полі має застосування в різних областях, включаючи обробку зображень, комп'ютерне бачення, графіку, проектування схем, маршрутизацію, розкладання планування, логістику та інші області, де важливо знайти найбільший вільний простір або область на заданому полі.

Висновок:

Комбінаторна оптимізація має велику цінність, оскільки дозволяє знаходити оптимальні рішення в складних ситуаціях з великою кількістю можливих варіантів. Вона застосовується в різних галузях, таких як логістика, транспорт, електроніка, виробництво та багато інших, допомагаючи вирішувати проблеми оптимального розподілу ресурсів та планування. Комбінаторна оптимізація допомагає зекономити час, гроші та ресурси, забезпечуючи ефективне використання можливостей і досягнення оптимальних результатів. На разі існує велика кількість задач на комбінаторну оптимізацію та немала кількість з них не мають остаточного ефективного алгоритму вирішення через свою складність і варіативність

Список джерел:

1. *What is combinatorial optimization?* URL: <https://www.engati.com/glossary/combinatorial-optimization>
2. *Knapsack problem*, URL: <https://xlinux.nist.gov/dads/HTML/knapsackProblem.html>
3. *Deterministic Greedy Algorithm for Maximum Independent Set Problem in Graph Theory* - Joshua C. Ballard-Myer
4. *Computing the largest empty rectangle* - B. Chazelle, R. L. Drysdale and D. T. Lee

УДК 004.41

*Менделюк К.В.,
студенти I курсу спеціальності 122
«Комп'ютерні науки»
Ніколюк П. К., професор, доктор
фізико-математичних наук.*

ІГРОВОЙ ДОДАТОК З ВИКОРИСТАННЯМ UNITY

Донецький національний університет імені Василя Стуса

Unity – це більше ніж ігровий движок, це середовище для розробки ігор на ПК та мобільних ігор, в якому об'єднанні різні програмні компоненти, які використовуються при створенні програмного забезпечення. Перевагою Unity є

те, що ти можна почати розробляти власний програмний продукт не маючи за плечами великого досвіду розробки. Не потрібні фундаментальні знання для того, щоб почати творити свій продукт. В Unity використовується компонентно-орієнтований підхід, в рамках якого ви створюєте якийсь об'єкт, а потім в порядку необхідності можна його кастомізувати. Другим великим плюсом Unity є великий вибір плагінів та асетів за допомогою яких можна сильно пришвидшити розробку. Третім плюсом Unity є кросплатформеність – це підтримка великої кількості платформ, технологій та інтерфейсів користувача. Великим недоліком при розробці ігор за допомогою Unity є те, що при розробці якомось масштабного проекту, команді знадобиться досвідчений програміст на мові C#, який зможе написати скрипти до компонентів та налаштувати їх правильне функціонування.

Панель інструментів забезпечує доступ до найважливіших робочих функцій. Зліва він містить основні інструменти для керування видом сцени та об'єктами GameObjects у ньому. У центрі розташовані елементи управління відтворенням, паузою та кроком. Кнопки праворуч дають вам доступ до Unity Collaborate, Unity Cloud Services та вашого облікового запису Unity, за яким слідує меню видимості шарів і, нарешті, меню макета редактора (яке надає деякі альтернативні макети для вікон редактора та дозволяє зберігати ваші власні макети).

Вікно ієрархії - це ієрархічне текстове представлення кожного GameObject у Сцені. Кожен елемент у Сцені має запис в ієрархії, тому два вікна невід'ємно пов'язані між собою. Ієрархія розкриває структуру того, як GameObjects приєднуються один до одного.

Перегляд гри імітує, як буде виглядати ваша остаточна відтворена гра через ваші камери сцен. Після натискання кнопки «Відтворити» починається моделювання.

Перегляд сцени дозволяє вам візуально орієнтуватися та редагувати вашу сцену. Вигляд сцени може показувати тривимірну або 2D-перспективу, залежно від типу проекту, над яким ви працюєте.

Вікно інспектора дозволяє переглядати та редагувати всі властивості вибраного на даний момент GameObject. Оскільки різні типи GameObjects мають різні набори властивостей, макет та вміст вікна інспектора змінюються кожного разу, коли ви вибираєте інший GameObject.

У вікні проекту відображається ваша бібліотека активів, доступних для використання у вашому проекті. Коли ви імпортуєте активи у свій проект, вони з'являються тут. [1]

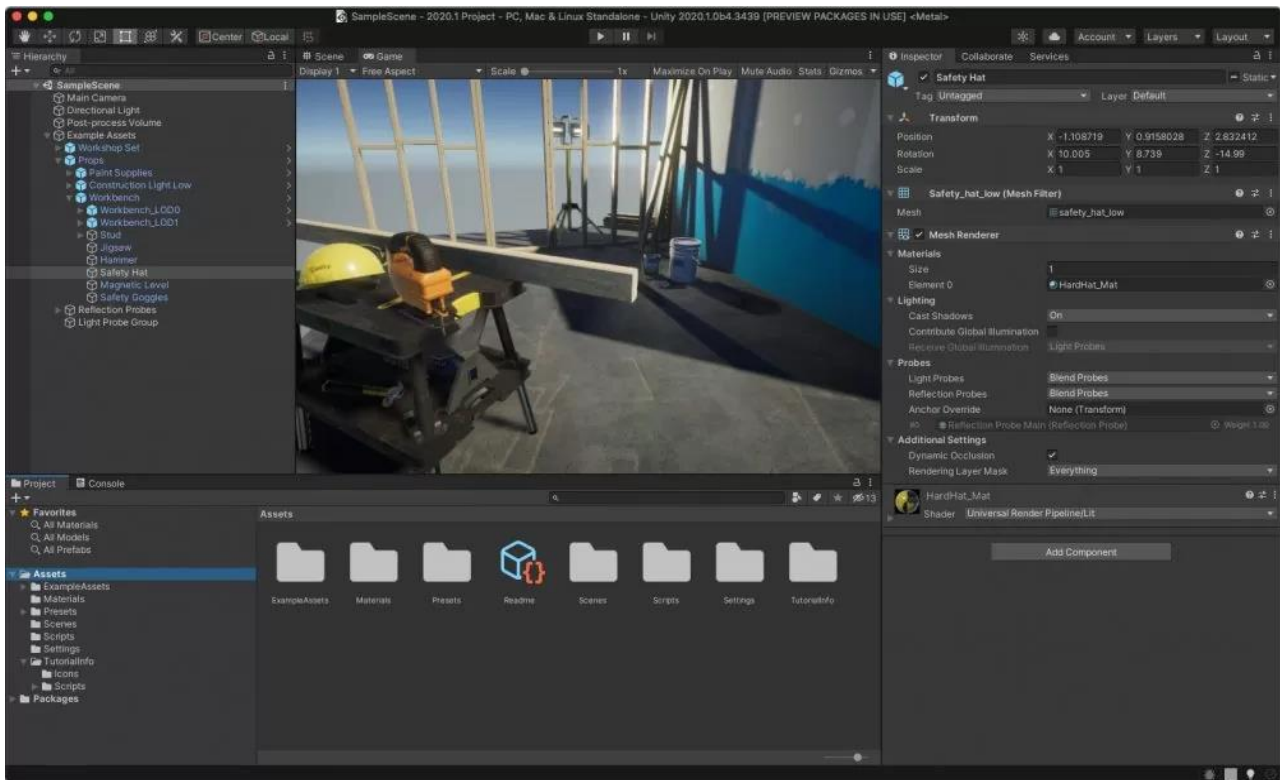


Рисунок 1 - Інтерфейс

Розглянемо більш детально доступні для використання параметри:

1) Scriptable Render Pipeline Settings (Налаштування параметрів візуалізації та скриптів) – даний параметр дозволяє визначити ряд команд, для того щоб точніше керувати чергою відображення об’єктів в сцені.

2) Camera Settings (налаштування камери) – група налаштувань, яка відповідає різним видам рендерінга.

2.1) Transparency Sort Mode – визначає порядок відображення об’єктів по 29 їх відстані відносно певної осі. Рендерінг в Unity сортується по певним критеріям, таким як номер слоя або відстань до камери.

Default – сортування об’єктів в залежності від положення камери.

Perspective – сортування об’єктів в залежності від види перспективи.

Orthographic – сортування об’єктів на основі ортографічного виду.

Transparency Sort Axis – сортування прозорості за бажанням користувача.

3) Tier Settings (Налаштування рівня)

Ці налаштування дозволяють вносити, специфічні для платформи коригування в рендерінг і компіляції шейдерів, шляхом налаштування 30 вбудованих визначень. Рівні (Tiers) визначені в бібліотеці Rendering.GraphicsTier.

- Standart Shader Quality – встановлення якості стандартного шейдера на високий, середній або низький рівенью

- Reflection Probes Box Projection – включає проекцію для відображення UV карт на Reflection Probes.

- Reflection Probes Blending – включає змішування Reflection Probes.
- Detail Normal Map – включає дану функцію якщо вона призначена і необхідна.
- Enable Semitransparent Shadows – включає напівпрозорі тіні. Ця функція додає або видаляє визначення для компілятора шейдерів.
- Enable Light Probe Proxy Volume – включає рендерінг 3D сітки з інтерпольованих Light Probes.
- Cascaded Shadows – включає використання каскадних тіней.
- Prefer 32 bit shadow maps – дає можливість роботи з 32 бітною картою тіней. Більшість платформ мають фіксований формат тіней, які ви не можете налаштувати. Вони відрізняються за форматами і можуть бути 12, 24 або 32 бітні.
- Use HDR – включає рендерінг з високим динамічним діапазоном.
- HDR Mode – дає можливість вибору формату, який буде використовуватися для буфера HDR. За базовим налаштуванням використовується формат FP16.
- FP16 – формат який використовує 16 біт з плаваючою точкою на канал.
- Rendering Path - виберіть як Unity повинен відображати графіку. Різні шляхи рендеринга впливають на продуктивність вашої гри, а також на те, як розраховуються освітлення і затінення. Деякі шляхи більше підходять для різних платформ і устаткування, ніж інші.

Відкладений (Deferred) рендеринг не підтримує при використанні ортогональної проекції. Якщо режим проекції камери встановлено на 31 ортогональний, то ці значення перевизначаються, і камера завжди використовує прямий (Forward) рендеринг.



Рисунок 2 - Графіка в Unity

При створенні проекту є можливість вибору між двома режимами роботи редактора це 2D і 3D. Але в будь-який момент часу ви можете переключатись між цими режимами. Ці два режими визначають налаштування редактора з якими вони будуть працювати. Розглянемо ці два режими роботи більш детально.

В режимі 2D редактор має наступні властивості:

- Будь-яке імпортоване зображення, рахується як 2D об'єкт (спрайт) і виставляється в режимі Sprite;
- Активованій режим Sprite Packer;
- Вікно відображення сцени працює в режимі 2D;
- При створенні нової сцени створюється камера яка знаходиться в положенні по координатам 0;0;-10 і має ортографічну проекцію;

В режимі 3D редактор має наступні властивості:

- Будь-які імпортовані зображення в проект не рахуються спрайтами;
- Режим Sprite Packer деактивується;
- Вікно сцени працює в режимі 3D;
- При створенні нової сцени створюється камера яка знаходиться в положенні по координатам 0;1;-10 і має перспективну проекцію;

Зазвичай 3D ігри використовують трьохвимірну геометрію, а матеріали і текстури візуалізуються на поверхності цих ігрових об'єктів, для того щоб вони виглядали як оточення цих ігрових об'єктів. Камери може вільно переміщуватися разом з світлом і тінями, які є досить реалістичними. 3D ігри зазвичай візуалізують за допомогою камер в режимі перспективи, через це об'єкти які знаходяться на екрані візуально мають більший розмір чим мають в реальності. Іноді в іграх використовується трьохвимірна геометрія, але замість перспективної використовують ортографічну камеру. Це розповсюджена техніка де камера має вид зверху, ці ігри називають 2.5D іграми.

Більшість 2D ігор використовують плоску графіку (спрайти), яка взагалі не мають трьохвимірної геометрії. Сцена будується у виді плоских зображень які накладуються один на одне. Деякі 2D ігри використовують трьохвимірну геометрію для створення персонажів, но при цьому обмежують ігровий процес двома вимірами.

Ще один із доступних можливостей при розробці 2D ігор, використовується 2D графіка разом з перспетивною камерою, для того щоб отримати ефект параллакса (об'єкти які знаходяться ближче до камери рухаються швидше ніж об'єкти які знаходяться далі).

Список літератури:

1. *Unity Manual* [Електронний ресурс] / режим доступу: <https://docs.unity3d.com/Manual/UsingTheEditor.html>