УДК 004.8

Т. О. Савчук
В. Л. Паламарчук

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ 3D ВІДЕОГРИ

Вінницький національний технічний університет

Анотація

Аналізуючи існуюче програмне забезпечення 3D відеоігр було встановлено необхідність створення простої у використанні та універсальної системи генерації рівнів будь-яких розмірів та властивостей.

Ключові слова: Програмне забезпечення, алгоритм, гра, генерація, оптимізація, система.

Abstract

Analyzing existing 3D videogame software, established the need of easy to use and versatile system of generating levels any size and properties

Keywords: Software, algorithm, game, generation, optimization, system.

Створення програмного забезпечення 3D відеогри із системою генерації рівнів буде успішним лише у тому випадку, якщо швидкість генерації рівня, навантаження на апаратне забезпечення є найменшою та згенерований рівень є адекватним.

Розрізняють три підходи до створення системи генерації рівнів [1]:

- оптимізація швидкості генерації;
- зменшення навантаження на апаратне забезпечення;
- адекватність згенерованого рівня.

Розглянемо можливість використання цих підходів при створенні програмного забезпечення для 3D відеогри.

Оптимізація швидкості генерації

Створюючи програмне забезпечення для 3D відеогри доцільно оптимізувати швидкість генерації рівня. Для цього важливо врахувати таке [1]:

- визначення розміру рівня,
- кількість варіантів текстур/моделей/об’єктів/частин рівня,
- можливість збільшення ресурсів, що виділяються на генерацію рівня.

Визначення приблизного розміру рівня відбувається при розробці концепту рівня.

При концептуальній розробці ігрових рівнів враховуються низка чинників таких, як рівень складності гри, приблизний час, який витратить гравець на проходження даного рівня, кількість неігрових персонажів(NPC), нагорода що здобуде гравець після проходження рівня.

Оптимізація кількості текстур/моделей/об’єктів Оптимізація кількості текстур/моделей/об’єктів досягається шляхом застосування задачі про пакування у ємності [2]. Вона полягає у пакуванні об'єктів визначеної завідомо форми в скінченне число ємностей (контейнерів) також завідомо відомої форми у такий спосіб, аби число використаних ємностей було найменшим або кількість чи об'єм предметів (які розміщують) були якнайбільшими.

Нехай дана множина ємностей **V** і множина розмірів предметів $\mathbf{a_1,\ldots,a_n}$. Необхідно знайти ціле число ємностей **B** і розбиття множин{**1,…,n**} на **B** таких підмножин $\mathbf{S_1}$ U...U $\mathbf{S_B}$, що для всіх **k = 1,…,B** . Очевидно, що чим менше **B** , тим вигідніший розв'язок вдалося знайти. $\sum_{i \in S_k} a_i \le V$

Тоді, задача пакування в ємності як задача лінійного програмування визначиться так:

$$B = \sum_{i=1}^{n} y_i$$

при обмеженнях

$$\sum_{j=1}^{n} a_j x_{ij} \le V y_i, \ \forall i \in \{1,\ldots,n\}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad \forall j \in \{1,\ldots,n\}$$

$$y_i \in \{0,1\}, \qquad \forall i \in \{1,\ldots,n\}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \{1,\ldots,n\} \forall j \in \{1,\ldots,n\}$$

де $y_i = 1$, якщо ємність i використовується, й $x_{ij} = 1$ якщо предмет j поміщено в ємність i.

Керування ресурсами Використання екрану завантаження та заставки вирішить задачу використання максимуму
доступних ресурсів, не руйнуючи враження від ігрового процесу.

Зменшення навантаження на апаратне забезпечення

Для генерації рівнів у стороненні 3D відеогри доцільно використовувати так називане зерно рандому(random seed), що дозволяє зменшити навантаження на процессор, не викликаючи кожного разу функцію псевдо-рандому, а зберігати попередньо згенеровані випадові числа [3].

Адекватність згенерованого рівня

Згенерований рівень повинен відповідати ряду вимог, які залежать від розробника системи генерації рівнів. До даних вимог входять:

- кількість кімнат в згенерованому рівні,
- кількість проходів між кімнатами згенерованого рівня,
- кількість глухих кутів,
- розміщення входу та виходу з ігрового рівня.

Поширеним методом генерації рівня є метод кривих Гільберта [4], використання якого забезпечить побудову унікального звивистого маршруту. Удосконалення алгоритму, що лежить в основі методу, дасть можливість будувати маршрут з певною кількістю глухих кутів, заданою кількістю кімнат сполучених між собою коридорами, а також коротким шляхами між ними.

Висновки

Використання запропонованих підходів оптимізації програмного забезпечення 3D відеоігр підвищить швидкість генерації рівнів, варіативність ігрового процесу та зручність ігрового управління.

Список використаних джерел:

1. Procedural dungeon generation. [Електронний ресурс] / A. Adonac // Gamasutra, – Режим доступу:www.gamasutra.com/blogs/AAdonaac/20150903/252889/Procedural_Dungeon_Generation_Algoritm.php
2. Савчук Т.О. Ідентифікація проблемних ситуацій та їх станів у складних технічних системах з використанням модифікованого алгоритму ФОРЕЛ / Т. О. Савчук, С. І. Петришин // Вісник НУ «Львівська політехніка», Інформаційні системи та мережі.-2014-№ 783.- С. 187-193
3. Random seed [Електронний ресурс] / David Eppstein // Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/Random_seed.
4. Hilbert Curves [Електронний ресурс] / Nick Berry // DataGenetics. – Режим доступу: http://www.datagenetics.com/blog/march22013/index.html.

Савчук Тамара Олександрівна — к.т.н, доцент кафедри комп'ютерних наук, професор кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця

Паламарчук Владислав Леонідович – *студент кафедри комп'ютерних наук ВНТУ, Він¬ницький національний технічний університет, м. Вінниця, e-mail:* palamarchuk.gg@gmail.com

Tamara O. Savchuk *— Cand. Sc. (Eng), Assistant Professor, Professor of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia*

Vladyslav L. Palamarchuk *— student of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail:* palamarchuk.gg@gmail.com