

КОМП'ЮТЕРНІ НАУКИ

УДК 004.4

DOI: <https://doi.org/10.32515/2414-3820.2021.51.227-238>**О.М. Дреєв**, доц., канд. техн. наук*Центральноукраїнський національний технічний університет, м.Кропивницький, Україна**e-mail: dreu.sanya@gmail.com***Б.Ю. Железняк**, студ.*Центральноукраїнський національний технічний університет, м.Кропивницький, Україна**e-mail: bzeleznak@gmail.com*

Аналіз комп'ютерних систем візуалізації з метою алгоритмізації обґрунтування щодо їх використання

Проектування програмного забезпечення передбачає використання кількох фреймворків, які хоч і скорочують час розробки програмного забезпечення, але мають свої можливості, недоліки та переваги. Використання фреймворка також накладає на проект певні архітектурні особливості, які залежать від обраного каркаса. Тому перед інженером-програмістом постає інтелектуальне завдання створення архітектури програмного забезпечення з вибором із ряду доступних фреймворків та з урахуванням їх впливу на функціонування системи в цілому. Результат проектування є важливим кроком, який може вплинути на успіх проекту. Тому в даній роботі розглядається проблема посилення об'єктивної складової у процесі вибору фреймворку, зокрема системи комп'ютерної візуалізації при створенні архітектури програмного забезпечення. В результаті запропоновано метод вибору системи візуалізації, який базується на аналізі завдань до системи візуалізації з виведенням вимог до побудованого зображення. Показано застосовність багатокритеріальної оптимізації для поділу застосовуваних систем візуалізації та вибору найкращої з них.

проекування, програмне забезпечення, прийняття рішення, комп'ютерна система візуалізації

Постановка проблеми. В роботі прийнято за термін «комп'ютерні системи візуалізації» приймати набір алгоритмів та програмних засобів в результаті використання яких ми отримуємо зображення. Введення даного терміну, який є близьким до терміну «комп'ютерна графіка», використано для відокремлення засобів отримання зображення від методів його зберігання, обробки та інших задач комп'ютерної графіки. Тому, тут і далі, «комп'ютерна система візуалізації» є частиною засобів «комп'ютерної графіки», яка відповідає за частину створення зображення на моніторі (проекторі або іншому пристрою виведення динамічної картини). В разі прийняття означення «комп'ютерні системи візуалізації», до комп'ютерної системи візуалізації будуть входити апаратні засоби побудови зображення та їх відображення, а також програмні засоби побудови цифрового зображення.

Аналіз останніх досліджень і публікацій. На сьогодні існує багато методів побудови цифрового зображення (растерізація) з геометричного завдання об'єктів сцени, властивостей поверхонь цих об'єктів та параметрів освітлення в сцені [1, 2, 3, 4]. Також сюди входять і засоби створення сцен з врахуванням растрових шаблонів цифрового зображення (текстур) [18]. Всі ці методи є відмінними по якості створення зображення по критеріям стабільності, реалістичності, правильності а також мають

колосальні відмінності в часі створення окремого кадру цифрової сцени. Більш того, існують відмінності в розрізі одного програмного забезпечення, з однаковим набором алгоритмів побудови зображення, але з різними налаштуваннями, або при використанні іншої апаратної бази. Все це безумовно впливає на швидкість та якість побудованого зображення, і різноманіття засобів та налаштувань систем візуалізації призводить до ускладнення процесу визначення засобів і налаштування системи візуалізації для вирішення конкретних задач.

Постановка завдання. В результаті аналізування стану проектування програмного забезпечення, яке використовує системи комп'ютерної візуалізації, маємо об'єктивне протиріччя, коли з одного боку маємо велику кількість комп'ютерних систем візуалізації, а з іншого є потреба в швидкому та правильному виборі конкретної системи візуалізації з врахуванням вимог що до візуалізації для вирішення конкретних задач. Для вирішення визначеного протиріччя було поставлено наступну мету:

Мета статті полягає в створенні методу обрання системи комп'ютерної візуалізації з врахуванням вимог що до розв'язуваних задач.

Об'єктом дослідження є процес проектування систем, які використовують комп'ютерну візуалізацію.

Предметом дослідження є методи визначення систем комп'ютерної візуалізації, які б відповідали поставленим задачам.

Поставлена мета містить ряд невизначеностей, які потрібно вирішити розв'язавши наступні задачі:

- 1) Провести класифікацію вимог до систем візуалізації та пов'язати ці вимоги з задачею, яку поставлено до системи візуалізації.
- 2) Провести класифікацію систем комп'ютерної візуалізації з врахуванням визначених вимог.
- 3) Побудувати метод визначення комп'ютерних систем візуалізації, які задовольнятимуть визначеним вимогам.

Виклад основного матеріалу. Відповідно до визначення вимог до задач, які повинна вирішувати система комп'ютерної візуалізації, потрібно провести насамперед класифікацію задач, які вирішує система візуалізації. В наш час комп'ютерні системи використовуються не лише в технічних областях, але вирішують задачі і в побуті, мають значну вагу в мультимедіа та використовуються для розваг та ігор. Відповідно, для кожної із задач, вимоги до системи візуалізації є суттєво різними. Наприклад, для мікроконтролерного пристрою, який надає кілька вимірних значень, може мати світлодіодний індикатор на кілька цифр, а система проектування повинна розгортати креслення з високою роздільною здатністю. Як показало дослідження, в більшості джерел [1, с.6; 2, с.20; 3, с.5; 4, с.3] використовують поділ комп'ютерної графіки не за класифікацією до вимог, що стоять до процесу утворення зображення, а по сферам людської діяльності, наприклад в [2] наведено класифікацію, яку відображає наступний список:

- 1) інженерна графіка;
- 2) автоматизовані системи наукових досліджень;
- 3) інформаційні системи;
- 4) системи ілюстративної та ділової графіки;
- 5) системи машинної геометрії;
- 6) анімаційні задачі;
- 7) комп'ютерні ігри;
- 8) відео тренажери;
- 9) мистецтво та видавнича діяльність.

Однак, в розрізі ділової графіки, необхідно враховувати засоби демонстрації графічних матеріалів. Наприклад, графік для зображення динаміки змін даних в часі для презентації матиме інші вимоги до системи візуалізації ніж фотографічне зображення планування жилого комплексу як і по роздільній здатності, так і в розрізі правильності передачі кольору. Більш детальна класифікація [1, с.6-7] містить опис специфічних вимог до систем візуалізації, але ці вимоги в одному пункті можуть мати протиріччя. Наприклад, за наведеною класифікацією інженерна графіка може вимагати моделювання в реальному часі та реалістичність візуалізації, що може привести до протиріччя вимог. Тому стає актуальним розв'язання задачі побудови класифікації вимог до систем візуалізації та розробки методики проєкції поставлених задач діяльності на ці вимоги до систем візуалізації.

Пропонується використовувати для пошуку системи візуалізації, систему яка базується на вимогах до результату візуалізації. Тому за основу можна запропонувати наступні вимоги до комп'ютерних систем візуалізації:

- 1) швидкодія;
- 2) правильність побудови освітлення (накладання тіней повинно бути правильним, але їх зображення може бути схематичним);
- 3) реалістичність побудови освітлення (допускає порушення законів оптики, пропущення тіней, але буде якісні для ока зображення);
- 4) підтримка динамічного освітлення;
- 5) точність відображення кольорів;
- 6) врахування властивостей поверхні при розрахунку кольору пікселя (відблиски, розсіювання, поглинання, світіння, колір поверхні);
- 7) використання текстур;
- 8) використання мап рельєфу;
- 9) зображення синтезоване;
- 10) зображення натуральне;
- 11) допуск втрат інформації при стисненні або відтворенні зображення;
- 12) вимоги до складності сцени;
- 13) можливість змінювати метод зображення (показати лише вершини, ребра, ігнорування текстур або матеріалу або інші опції, які можна змінити в довільний момент);
- 14) можливість будувати стереоскопічні пари зображення сцени;
- 15) інші вимоги.

З точки зору виконавця технічного проєкту, в якому використовуються системи візуалізації, формулювання вимог не є першим етапом, а в першу чергу формується задача, яку потрібно розв'язати за допомогою системи візуалізації. Приклади таких задач показано в наступному переліку:

- 1) плакати;
- 2) ділова графіка;
- 3) публікація в Інтернеті;
- 4) 3D графіка, художня;
- 5) графіка для динамічних ігор;
- 6) графіка для повільних ігор;
- 7) графіка інженерна;
- 8) графіка конструкторська;
- 9) технічні інтерактивні схеми для устаткування;
- 10) відображення графіків та діаграм;
- 11) відеомонтаж;

- 12) відеоспостереження;
- 13) медичне зображення;
- 14) наукове зображення;
- 15) відображення текстів для редагування;
- 16) системи віртуальної реальності;
- 17) інші.

Створені списки є суб'єктивно-орієнтовані і можуть відрізнятися в залежності від кола інтересів робочих груп. Тому метод може трансформуватися до конкретних алгоритмів, які відповідатимуть потребам конкретних груп розробників або замовників програмного забезпечення.

На основі наведених переліків вимог та задач пропонується метод обрання систем візуалізації на основі наступної послідовності дій (термінологія наукових методів, які використані в блок-схемі, взято з [5]; рис. 1):

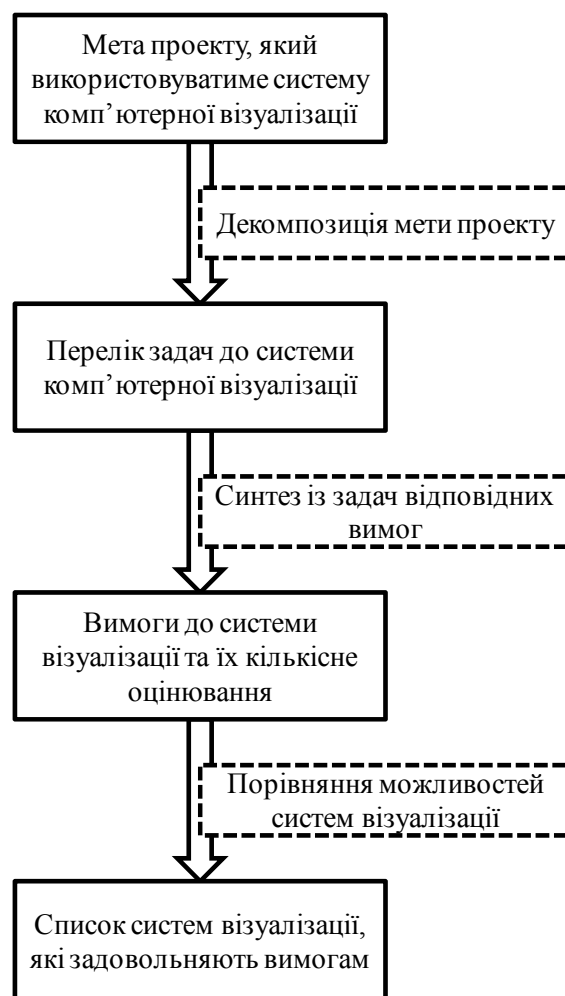


Рисунок 1 – Метод визначення сумісних комп'ютерних систем візуалізації

Джерело: розроблено автором

Побудований метод не є детермінованим, навіть при умові фіксованих списків задач та вимог, тому що метод містить дії, які мають інтелектуальні складові при виконанні дій декомпозицій та синтезу. Процес порівняння та конкурс комп'ютерних систем візуалізації є можливість формалізувати при умові фіксації списку вимог до систем візуалізації та засобів їх кількісного оцінювання, наприклад, використавши на

основі теорії множин та кортежів кількісних оцінок правила багатокритеріальної оптимізації Парето [6].

Експериментальна частина. Авторами розглянуто приклад застосування методу обрання системи комп'ютерної візуалізації на прикладі проектування програмного забезпечення, яке реалізує покрокову стратегію. Однак, формулювання вимог до системи комп'ютерної візуалізації можливий лише після аналізу вимог до отриманого зображення з порівнянням вимог до інших програмних засобів де використовується графіка. Тому першим етапом обрання системи візуалізації є формулювання мети проекту програмного забезпечення – побудова зображення для комфортної гри в жанрі покрокової стратегії.

1) Формування переліку задач на основі декомпозиції мети проекту.

Швидкість побудови зображення є однією із головних особливостей динамічної (ігрової) візуалізації, оскільки від швидкості побудови зображення залежить сприйняття користувачем розробленої гри. Це зобов'язання виникло внаслідок необхідності враховувати швидкість сприйняття інформації людиною. У середньому, людський зір здатний реагувати лише на 20-25 кадрів у секунду. Вимоги до швидкості побудови зображення різняться від динаміки жанру гри, наприклад, у шутерах та гонках спостерігається часта зміна зображення і для забезпечення вчасного реагування на зміни в оточенні часто вимагається можливість побудови 120 та більше кадрів за секунду. У стратегічних іграх, графічних редакторів та інженерних рішеннях, зокрема і в покроковій стратегії, швидкість зміни кадрів повинна бути лише достатньою для забезпечення видимості плавності анімацій та відсутності мерехтіння – 30 кадрів за секунду. Більш детально описано про дотримання частоти зміни кадрів наведено в [7]. Покрові стратегічні ігри насамперед базуються на принципах неспішності, тому динамічна зміна зображення в таких іграх створюється для подоби живої картини. Застосування такої динаміки вдало спостерігається на прикладі гри з серії Civilization. Починаючи з перших ігор у серії, пересування ігрових персонажів, побудова внутрішніх та зовнішніх об'єктів у містах, використання зброї масового ураження та природні лиха супроводжувалися динамічною зміною на карті. Такий виборчий підхід до динаміки зміни зображення дозволив підтримувати плавну та захоплюючу атмосферу, а також задовольнити потребу користувача відстежити прогрес в грі.

Велику швидкість зміни зображення можна простежити на прикладі гри Quake та її модифікації DeFRaG. Гра Quake є класичним шутером від першої особи, в якому швидкість зміни кадрів середньо-швидка. В її модифікації DeFRaG, при активному проходженні мап, на певний час можлива швидка зміна кадрів. Швидкість руху в гоночних іграх може досягати більш 250 кілометрів за годину, тому, з огляду на те, що під час проходження траси потрібно часто змінювати положення камери, зміна кадрів також повинна відбуватися з достатньою частотою.

Користувач повинен чітко розуміти особливості поверхні та рельєфу в ігрових локаціях. Наприклад, якщо розглядати культову гру Heroes Might of Magic 3, то в ній спостерігається розподілення на глобальну карту та на карти битв. На глобальній карті рельєф, перш за все, впливає на швидкість переміщення героя на тій чи іншій місцевості. В битвах ігрові одиниці, що приписані місцевості на якій йде битва, отримують прибавку у 1 одиницю наступних показників: атака, захист, швидкість (як переміщення, так і ініціатива). Крім того, в битвах має певне значення рельєф карти в вигляді перешкод, що інколи кардинально змінює результат битви. Місцевість впливає не лише на бійців, але і на самого героя, забороняючи використовувати магію вище 1-го рівня, або дозволяючи чаклувати усі заклинання на рівні «експерт», що має найвищий рівень [8].

При створенні рельєфу варто звертати увагу не тільки на технічну складову, а і на візуально динамічну. Динамічні сцени покрокової стратегії використовують найчастіше для створення живого ефекту та поділяються на «пасивні» та «активні», що реалізується за допомогою об'єктів на карті (наприклад, курка, яка дзьобає щось на землі, працюючий млин, дим з будинків і т.д.), пересування хмар та води, дій ігрового персонажу на місці (наприклад, виконує анімацію перевірки свого клинку на гостроту). Спочатку проминають такі моменти, адже найчастіше покрокові комп'ютерні стратегії розробляються як аналог настільній гри. Але якщо приділити анімаціям увагу, то гра отримає більш живу атмосферу, а гравець можливість більш зануритися в атмосферу гри.

Однією із складових якісної гри є оптимізація. Спрощення візуалізації об'єкта є найлегшим способом досягти підвищення швидкості формування зображення. Під спрощенням візуалізації передбачаються дії, які приводять до зменшення потреб в обробці зображення. До описаних дій належить прив'язка статичної тіні об'єкта до власне моделі об'єкта. Проблема згаданої прив'язки може бути лише в динаміці тіней, тому такий підхід до оптимізації гри є рентабельним. Якщо розглядати точність побудови тіні об'єкта, то для початку потрібно знати як саме тінь була створена, чи є вона частиною моделі об'єкта, або динамічно формується при зміні кута та сили освітлення. Якщо аналізувати ігри від першого особи, то можна спостерігати динамічну зміну тіней або її відсутність при взаємодії об'єкта з навколишнім середовищем, тобто зміни освітлення за допомогою умовно «факела», що активується і підводиться до об'єкта дослідження. Якщо досліджувати більш складну техніку освітлення "Muzzle Flash Shading", у якій створюється ефект реалістичних тіней від об'єктів при стрільбі з вогнепальної зброї, то варто відзначити, що такий ефект спостерігається не в кожному шутері, а його реалізації в іграх від третьої особи і зовсім немає і по цей час. Згаданий ефект мав би бути у Gears of War 1, так як це було демонстровано в трейлері гри, але був вирізаний з релізної версії, тому "Muzzle Flash Shading" можна зустріти лише в іграх Doom 3, Killzone 2, STALKER [9].

Зазвичай саме в іграх з віддаленою камерою та при наявності умовно 3D об'єктів застосовуються статичні тіні, так в грі Heroes Might of Magic 3, можна спостерігати прив'язану тінь об'єкта, але навіть така тінь є необхідною для створення об'ємності та правильності об'єкта.

Ще однією особливістю динамічної візуалізації є технологія трасування променів. Згадана технологія дозволяє придати грі більшій реалістичності, завдяки обчислюванню траєкторії кожного променя світла від об'єкта до камери, і виводить на екран картинку з урахуванням проведених розрахунків. Перш ніж використовувати технологію трасування променів потрібно ретельно ознайомитися з матеріалами та кольорами, які будуть на сцені, так як одні з них краще відображають світло, інші гірше, а скло і зовсім переломлює промені, що створює інший ефект. Докладний підхід дає можливість точно передати освітленість предметів, їхні тіні, заломлення, розсіювання світла й, взагалі, надавати зображенню на екрані реалістичніший вигляд. Окрім цього, трасування променів дозволяє спростити створення рівнів та їхнього оточення. Так при створення рівня, на якому буде горіти світильник, здебільшого, потрібно налагодити тіні та промені вручну. А за допомогою трасування променів, алгоритм, що відповідає за цю роботу, обробить складові рівня в автоматичному режимі. Окрім цього, трасування може працювати й зі звуком, створюючи набагато реалістичніше аудіо для ігор. Проблемою технології трасування є її витрати на ресурси комп'ютера, тому що обробка одного кадру за допомогою технології трасування може займати від декількох годин до декількох днів. Отже, цілковито застосовувати досліджувану технологію для створення ігор на теперішній час не є раціональним

рішенням, потрібно її використовувати за потреби та лише для деяких об'єктів [10, 11]. Відповідно, технологія трасування променів хоча і є бажаною частиною гри, але не є складовою, особливо якщо розглядати такий жанр як стратегія, де саме ця технологія майже не застосовується.

В багатьох випадках вимагається ілюзія реалізму або навмисна стилізація до рисунків. Ілюзія реалізму або стилізація потрібна при створюванні ігор наближеної реальності: історичних, фантастичних, мультиплікаційних та інших історій [12, 13].

Існує велика кількість елементів, які неможливо спростити. Вдалим прикладом системи, в якій неможливо спростити елементи, може бути візуалізація результатів медичного обстеження, або креслення та схеми приладів або схем.

В більшості інженерних та художніх програмах є можливість переключатися між методами візуалізації. Відображення лише ребер об'єкту дозволяє чітко побачити утворення форми, наявність внутрішніх елементів та інше. Спрощений рендер без урахування особливостей поверхні, тобто матеріалу та кольору, тобто текстури, дозволить дуже швидко будувати складні зображення з більшим вмістом вершин та граней без спрощення форми. Спрощений рендер, хоча і враховує матеріал та текстуру, призначений лише для оцінювання отриманої картини або для художнього сприйняття. Остаточне зображення будується найбільш якісними рендерами, за допомогою яких можна досягти фотореалістичності, але при цьому побудова кадру, в залежності від складності сцени, може продовжуватися від кількох секунд до декількох годин.

Технічне зображення повинне бути зрозумілими, а не гарним. Для технічної візуалізації необхідна насамперед якість виконання роботи, зображення повинно містити лише необхідну інформацію. Непотрібно гнатися за гарним зображенням, головне щоб все було зрозуміло і не відволікало від роботи із програмою. Тому такі програми як AutoCAD, для створення технічних креслень, не має можливості створювати фото реалістичні сцени, проте має можливість швидко будувати складні об'єкти та їх перерізи із точним врахуванням освітленості. Це досягнуто за рахунок зниження реалістичності та зниження частоти кадрів [14]. Також тут підвищені вимоги правильного переходу від полігонального зображення об'єктів до його креслення в заданій проекції. Вдалим прикладом може слугувати будинок із його переходом від цілого будинку до окремих поверхів, квартир, або можна уявити двигун та його складові.

В результаті можна відокремити наступні задачі візуалізації комп'ютерної покрокової стратегічної гри: забезпечити частоту кадрів; забезпечити зображення складних сцен; забезпечити зображення тіней від об'єктів сцени (правильність, реалістичність, динамічні/статичні), забезпечити врахування властивостей матеріалу; забезпечити використання текстур; забезпечити точність відтворення форм; врахувати відбиття та заломлення світла.

Інші вимоги та задачі виходять за рамки статті і освітлені в основній роботі.

В результаті огляду технічних особливостей комп'ютерної візуалізації є можливість переходу до їх систематизації та аналізування вимог по кожній з відокремлених задач, що відповідає наступному пункту за схемою на рис. 1.

2) Формування переліку вимог.

Визначення вимог до системи візуалізації вимагає на етапі проектування знання та уявлення про запланований готовий програмний продукт. Для цього використовується робота художників та сценаристів, які дадуть уявлення про те, що повинен побачити гравець, кількість елементів та рухомих об'єктів в кадрі, та інші не менш важливі характеристики. Тут є вагомими розкадровки майбутнього програмного забезпечення, типових етапів роботи та приклади робочих сцен (рис. 2)



Рисунок 2 – Проект ігрової сцени програмного продукту, який розробляється
Джерело: розроблено автором

На основі зазначених міркувань, можна зробити витяг основних задач, які полягатимуть на систему візуалізації. За мету роботи прийнято обрання системи візуалізації покрової стратегії, але для порівняння та підкреслення особливостей додамо задачі візуалізації шутеру та системи попередньої демонстрації моделей для 3Д-друку (табл. 1):

Таблиця 1 – Задачі, які поставлені до системи візуалізації

	Покровава стратегія	Шутер	Демонстрація моделей 3Д-друку
Швидке генерування кадру	Близько 30	Більше 30, бажано 120 та більше	До 1..3 с.
Складність сцени	Висока	Середня	Дуже складні
Правильність тіней	Можливі схематичні тіні	Можливі схематичні тіні	Високі вимоги до правильності побудови тіней
Реалістичність тіней	Бажаний реалізм	Бажаний реалізм	Реалізм не вимагається
Динамічні тіні	Рельєф незмінний, можливо використати статичні тіні	Бажано використати динамічні тіні	Тіні розраховуються для кожного кадру динамічно
Використання властивостей матеріалу	Високі вимоги до врахування матеріалу	Високі вимоги до врахування матеріалу	Достатньо загального кольору
Якість текстур	Висока	Висока	Текстури відсутні
Можливість спрощення моделей	Так	Так	Ні
Врахування відбиття та заломлення світла	Бажано, приблизно	Бажано, приблизно	Необхідно, точно

Джерело: розроблено автором

3) Кількісне оцінювання вимог до системи візуалізації.

Наступний крок також містить суб'єктивну складову, яку виконують експерти, які проводять проектування програмного продукту. Для цього відокремлені задачі з приблизними вимогами зводять в таблицю (табл. 2). Тут проводиться кількісна оцінка вимоги та експертна оцінка важливості дотримання окремої вимоги:

Таблиця 2 – Кількісна оцінка вимог та їх важливості

	Кадри за сек.	Складність сцени	Тіні			Матеріал	Мультиполігональність	Використання RTX	Вартість
			Правильність	Реалізм	Динамічність				
Показник	30	1000 об'єктів	Ні	Так	Так	Так	Так	Так	0 грн.
Важливість, 0..1	1	1	0,3	0,8	0,5	0,75	0,9	0,2	0,95

Джерело: розроблено автором

Після формування таблиці вимог, їх важливості, потрібно визначити характеристики доступних систем візуалізації, та обрати з них більш оптимальні за принципом Парето.

4) Вибір систем візуалізації, які задовольняють вимогам.

Обрання системи візуалізації проведено за характеристиками, які наведено в [15-18] (табл. 3):

Таблиця 3 – Кількісна оцінка вимог та їх важливості

	Кадри за сек.	Складність сцени	Тіні			Матеріал	Мультиполігональність	Використання RTX	Вартість, грн.	Зважена оцінка
			Правильність	Реалізм	Динамічність					
Важливість, 0..1	1	1	0,3	0,8	0,5	0,75	0,9	0,2	0,95	
Цільові показники	30	1000 об'єктів	-	+	+	+	+	+	0	6,40
Unreal Engine [15]	+	+	-/+	+	+	+	+	+	0	6,25
Unity3D [16]	+	+	-	+	+	+	+	-	0	6,20
Blender 3D [17]	+	-/+	+	+	+	-/+	-	+	0	4,90
GoDot [18]	+	+	-	-	-	-	-	-	0	3,25

Джерело: розроблено автором

В результаті співставлення вимог, з врахуванням їх важливості, виділено перевагу систем створення ігрового програмного забезпечення Unreal Engine та

Unity3D. Обидві системи мають високу відповідність вимогам. Тому для уточнення вибору потрібно врахувати вже не інженерні показники. Наприклад, важливою частиною аргументів що до вибору, є наявність підготовлених кадрів та програмістів, для роботи з тією або іншою системою.

Висновки. За поставленою метою, створення методу обрання системи комп'ютерної візуалізації з врахуванням вимог що до розв'язуваних задач, проведено обрання системи комп'ютерної візуалізації гри в жанрі покрокової стратегії. В результаті:

Проведено класифікацію вимог до систем візуалізації та пов'язано ці вимоги з задачею, яку поставлено до системи візуалізації. За результатами класифікації вимог, сформовано класифікацію систем комп'ютерної візуалізації з врахуванням визначених вимог.

На основі введених класифікацій побудовано метод визначення комп'ютерних систем візуалізації, який використано для проектування ігрового програмного забезпечення покрокової стратегії, і в результаті отримано рекомендації що до використання фреймворків Unreal Engine або Unity3D.

Створений метод обрання системи комп'ютерної візуалізації є універсальним, і його можна використати для проектування програмного забезпечення. Метод дає системний підхід до розв'язання алгоритмічно невизначеної задачі обрання системи комп'ютерної візуалізації за допомогою експертних оцінок та багатокритеріальних порівнянь.

Список літератури

1. Комп'ютерна графіка : конспект лекцій для студентів усіх форм навчання спеціальностей 122 «Комп'ютерні науки» та 123 «Комп'ютерна інженерія» з курсу «Комп'ютерна графіка» / Укладач: Скиба О.П. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2019. 88 с
2. Маценко В.Г. Комп'ютерна графіка: навч. посіб. Чернівці: Рута, 2009. 343 с. ISBN 966-568-846-4
3. Новожилова М.В., Мироненко В.В. Комп'ютерна графіка. Ч.1: навч.-метод. посіб. Х.: ХНУБА, 2015. 60 с.
4. Білуха М. Т. Методологія наукових досліджень: підруч. К.: АБУ, 2002. 480 с.
5. M. Ehrgott and X. Gandibleux. Approximative Solution Methods for Multiobjective Combinatorial Optimization . TOP. Sociedad de Estadística e Investigación Operativa. 2004. Т. 12, вип. 1.
6. Michael Klappenbach Understanding and Optimizing Video Game Frame Rates // December 16, 2020. URL: <https://www.lifewire.com/optimizing-video-game-frame-rates-811784>
7. Heroes Might of Magic 3. URL: <https://www.ubisoft.com/ru-ru/game/heroes-of-might-and-magic-3-hd>
8. Крис Фади́на Большое исследование про свет в играх // 10 августа 2020. URL: <https://cyber.sports.ru/tribuna/blogs/betterthansex/2812994.html>
9. Улучшенная графика с трассировкой лучей и DLSS. URL: <https://www.nvidia.com/ru-ru/geforce/rtx/>
10. Владислав Миронович. Что такое трассировка лучей и нужна ли она нам в играх . URL: https://blog.allo.ua/chto-takoe-trassirovka-luchej-i-nuzhna-li-ona-nam-v-igrakh_2020-05-51/
11. Unity Toon Shader мультяшний шейдер (1): мультипликационная раскраска. URL: <https://russianblogs.com/article/4312217817/>
12. Toony Colors Pro 2. URL: <https://assetstore.unity.com/packages/vfx/shaders/toony-colors-pro-2-8105>
13. AutoCAD overview. URL: <https://www.autodesk.ru/products/autocad/overview>
14. Make something Unreal . URL: <https://www.unrealengine.com/en-US/unreal>
15. Встречайте Unity Gaming Services. URL: <https://unity.com/ru>
16. The Freedom to Create. URL: <https://www.blender.org/about/>
17. GODOT showcase. URL: <https://godotengine.org/showcase>
18. Романюк О.Н., Романюк С.О., Піддубецька М.П. Аналіз методів анізотропної фільтрації. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2013. № 2. С.123-128

References

1. Skyba, O.P. (2019). *Komp'uterna hrafika [Computer Graphics]*. Ternopil' : Ternopil's'kyj natsional'nyj tekhnichnyj universytet imeni Ivana Puliuia [in Ukrainian].
2. Matsenko, V.H. (2009). *Komp'uterna hrafika [Computer Graphics]* . Chernivtsi: Ruta[in Ukrainian].
3. Novozhylova, M.V. & Myronenko, V.V. (2015). *Komp'uterna hrafika [Computer Graphics]*. Part 1. Kh.: KhNUBA [in Ukrainian].
4. Bilukha, M. T. (2002). *Metodolohiia naukovykh doslidzen [Research methodology]*. Kyiv: ABU [in Ukrainian].
5. Ehr Gott, M. & Gandibleux, X. (2004). Approximative Solution Methods for Multiobjective Combinatorial Optimization . TOP. Sociedad de Estadística e Investigación Operativa. *Part. 12. Vol. 1.* [in English].
6. Michael Klappenbach Understanding and Optimizing Video Game Frame Rates // December 16, 2020. *www.lifewire.com*. Retrieved from <https://www.lifewire.com/optimizing-video-game-frame-rates-811784> [in English].
7. Heroes Might of Magic 3; *www.ubisoft.com*. Retrieved from <https://www.ubisoft.com/ru-ru/game/heroes-of-might-and-magic-3-hd> [in English].
8. Kris Fadina Bol'shoe issledovanie pro svet v igrakh [Chris Fadina A great study on lighting in games]. 10 avgusta 2020. *cyber.sports.ru*. Retrieved from <https://cyber.sports.ru/tribuna/blogs/betterthansex/2812994.html> [in Russian].
9. Ulutshennaja grafika s trassirovkoj lucej i DLSS [Improved graphics with ray tracing and DLSS]. *www.nvidia.com*. Retrieved from <https://www.nvidia.com/ru-ru/geforce/rtx/> [in Russian].
10. Vladislav Mironovich. Chto takoe trassirovka lucej i nuzhna li ona nam v igrakh [What is ray tracing and do we need it in games]. 01.05.2020. *blog.allo.ua*. Retrieved from https://blog.allo.ua/chto-takoe-trassirovka-lucej-i-nuzhna-li-ona-nam-v-igrakh_2020-05-51/ [in Russian].
11. Unity Toon Shader mul'tjashnyj shejder (1): mul'tplikacionnaja raskraska [Unity Toon Shader cartoon shader (1): cartoon coloring]. *russianblogs.com*. Retrieved from <https://russianblogs.com/article/4312217817/> [in Russian].
12. Toony Colors Pro 2. *assetstore.unity.com*. Retrieved from <https://assetstore.unity.com/packages/vfx/shaders/toony-colors-pro-2-8105> [in English].
13. AutoCAD overview. *www.autodesk.ru*. Retrieved from <https://www.autodesk.ru/products/autocad/overview> [in English].
14. Make something Unreal. *www.unrealengine.com*. Retrieved from <https://www.unrealengine.com/en-US/unreal> [in English].
15. Vstrechajte Unity Gaming Services [Meet Unity Gaming Services]. *unity.com*. Retrieved from <https://unity.com/ru> [in Russian].
16. The Freedom to Create . *www.blender.org*. Retrieved from <https://www.blender.org/about/> [in English].
17. GODOT showcase . *godotengine.org*. Retrieved from <https://godotengine.org/showcase> [in English].
18. Romaniuk, O.N. & Romaniuk, S.O. & Piddubets'ka, M.P. (2013). Analiz metodiv anizotropnoi fil'tratsii [Analysis of anisotropic filtration methods]. *Vymiriuval'na ta obchysliuval'na tekhnika v tekhnolohichnykh protsesakh – Measuring and computing devices in technological processes*, 2, 123-128 [in Ukrainian].

Olexandr Drieiev, Assoc. Prof., PhD tech. sci., **Bohdan Zhelesnyak**, student

Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine

Analysis of Computer Visualization Systems in Order to Algorithmize the Rationale for their Use

Designing modern software involves the use of several frameworks. The framework reduces software development time. But frameworks have their own capabilities, disadvantages and advantages. The use of the framework also imposes certain architectural features on the project. Therefore, the engineer faces the intellectual task of creating a software architecture when you need to choose from an available set of frameworks and take into account the impact of these frameworks on the functioning of the system as a whole. The design result is an important step that has an impact on the success of the project. Therefore, this paper considers the problem of strengthening the objective component in the process of choosing a framework, namely the computer visualization system when creating a software architecture.

As a result, a method for selecting a visualization system is proposed. This method is based on the analysis of tasks that are put forward in the visualization system, taking into account the requirements for the image to be built. On the basis of the formed tasks to the visualization system, the article synthesizes the requirements for the visualization systems. Also, according to experts, the importance of compliance with certain requirements is assessed. According to the characteristics of computer visualization systems and quantitative

values, the evaluations provided by experts show the applicability of multicriterial optimization to separate the applicable visualization systems and choose the best one. In order to illustrate the choice of computer visualization system, the article uses an example of applying the method to a turn-based strategy game.

The result is a method for selecting computer visualization systems, which is based on the created classifications of tasks facing the visualization system. The relationship between the purpose of the visualization system and the list of tasks that the visualization system must perform is shown. The paper identifies the stages that depend on subjective and objective factors. The results allow a structured approach to computer visualization systems in the software architecture design process.

design, software, decision making, computer visualization system

Одержано (Received) 02.11.2021

Прорецензовано (Reviewed) 24.11.2021

Прийнято до друку (Approved) 29.11.2021