

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеська національна академія харчових технологій
Університет Інформатики і прикладних знань, м.Лодзь, Польща
Національний технічний університет України «Київський
політехнічний інститут»
Навчально-науковий інститут комп'ютерних систем і технологій
«Індустрія 4.0» ім. П.М. Платонова

XXI Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

22-23 квітня 2021 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22-23 квітня 2021 р. - Одеса, Видавництво ОНАХТ, 2021 р. – 229 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова - д.т.н., проф., **Єгоров Б.В.**, ректор ОНАХТ.

Співголови:

Поварова Н.М. – к.т.н., доц., проректор з наукової роботи ОНАХТ,
Котлик С.В. – к.т.н., доц., директор ННІКСІТ "Індустрія 4.0" ОНАХТ,
Даріуш Долива, д.математичн.наук, уповноважений декана факультету Інформатики УІтаПЗ, м.Лодзь, Польща,
Ковалюк Т.В. - к.т.н., доц. кафедри АСОІтаУ НТУУ «Київський політехнічний інститут»

Члени оргкомітету:

Плотніков В. М. – д.т.н., проф., завідувач кафедри ІТтаКБ ОНАХТ,
Артеменко С.В. – д.т.н., проф., завідувач кафедри КІ ОНАХТ,
Хобін В.А. – д.т.н., проф., завідувач кафедри АТПтаРС ОНАХТ,
Тарасенко В.П. – д.т.н., проф., завідувач кафедри СКС НТУУ «Київський політехнічний інститут»,
Невлюдов І.Ш. – д.т.н., проф., завідувач кафедри КІТАМ ХНУРЕ,
Мельник А.О. – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,
Жуков І.А. – д.т.н., проф., завідувач кафедри КСтаМ НАУ.

Матеріали подано українською, російською та англійською мовами.
Редактор збірника Котлик С.В.

ВИКОРИСТАННЯ ГРАФІЧНОГО ФРЕЙМВОРКУ LIBGDX ДЛЯ РОЗРОБКИ КРОСПЛАТФОРМНИХ ІГОР

РОМАНЮК О.Н., ВЕРЕНЬКО А. І., МИРГОРОДСЬКИЙ А. В.
Вінницький національний технічний університет

Розглянуто особливості використання графічного фреймворку LIBGDX для розробки кросплатформних ігор

Графічні API дозволяють напряму використовувати новий функціонал, але сучасний ринок мультимедійних додатків та ігор вимагає швидкого процесу розробки нових продуктів, а для максимізації прибутку додатково необхідна підтримка великої кількості платформ з різним апаратним забезпеченням. Кросплатформні бібліотеки та фреймворки, такі як libGDX, покликані спростити ці задачі для розробників.

Метою дослідження є вивчення варіантів використання та можливих переваг графічного фреймворку libGDX для розробки кросплатформних ігор.

Графічний фреймворк libGDX – це високопродуктивний, кросплатформний ігровий Java фреймворк з відкритим кодом, що базується на графічному API OpenGL ES та має підтримку WebGL для роботи в веб-браузерах [1]. На відміну від багатьох популярних платформ, заснованих на редакторах, libGDX не обмежує розробника певною методологією, окремим дизайном чи стилем кодування. Він повністю орієнтований на код і дозволяє керувати кожним аспектом створюваного ПЗ. Не зважаючи на це, libGDX пропонує велику кількість додаткових інструментів та бібліотек, які значно розширюють можливості розробників.

Код LibGDX ліцензований під Apache 2.0 – популярною ліцензією для вільного програмного забезпечення від Apache Software Foundation, тому фреймворк активно підтримується спільнотою розробників [2]. Використання Java та JVM в якості платформи для розробки відкриває доступ до цілої екосистеми потужних інструментів: різноманітні середовища розробки, повна підтримка багатьох систем контролю версій, відлагоджувачі з широким спектром функцій для відслідковування недоліків та зручного профілювання додатків, а також велика кількість надійних бібліотек з обширною документацією.

LibGDX використовується для розробки ігор і додатків, що працюють у режимі реального часу. Такі додатки називають RTA. Їхньою особливістю є те, що вони виконують потрібні дії у момент отримання завдання. Додатки реального часу мають ряд переваг, серед яких можна виділити: гнучкість, адаптивність, швидкість доступу, зменшена фрагментація даних тощо.

LibGDX має власний інструмент налаштування проекту. Це застосунок, що дозволяє сформувати проект за параметрами, які задає розробник. Згенерований проект доступний для редагування в будь-якій IDE для мови програмування Java або для розробки ПЗ для конкретної платформи, наприклад, Android Studio.

Під час першого відкриття згенерованого проекту система автоматичного збирання Gradle закінчить підготовку даних і файлів з вихідним кодом для роботи. Отриманий проект поділений на кілька модулів: основний, що містить платформно-незалежний код логіки гри або мультимедійного додатку, та платформно-специфічні: Android, Desktop, Web і iOS. Взаємодія між ними організована таким чином, щоб код для певної платформи залежав та використовував основний код логіки. Тобто, для більшості розробникам необхідно буде написати код один раз і він відразу запрацює на більшості пристроїв, лише за певної потреби або використання специфічних можливості з'явиться необхідність редагувати додатковий код для конкретної платформи. Результуючий початковий стан проекту залежить від обраних в конфігураторі налаштувань.

Архітектура libGDX та його взаємодія з апаратним і програмним забезпеченням показана на рисунку 1.

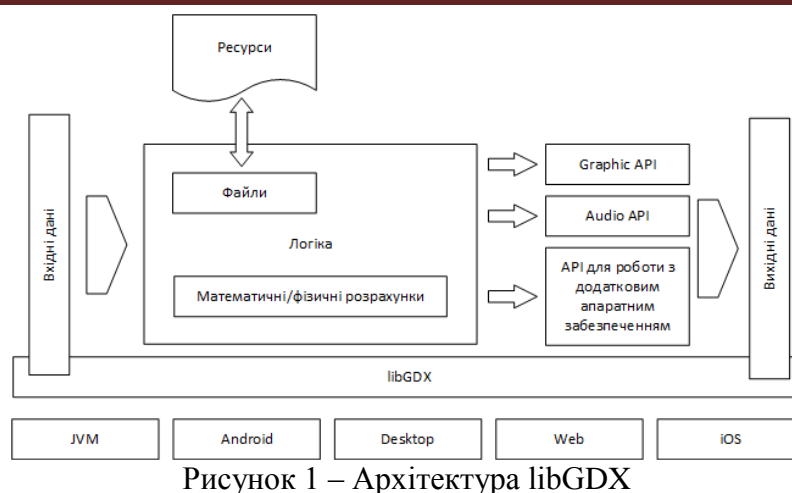


Рисунок 1 – Архітектура libGDX

За замовчуванням усі візуалізації в libGDX виконуються з віссю у, спрямованою вгору. Точка відліку координат для осей x та y знаходиться в лівому нижньому куті екрана. Для рендеру зображення 3D-геометрія, шейдери та текстери передаються в GPU за допомогою графічного API OpenGL, задачі по підготовці цих даних та роботі з низькорівневими запитами до API виконує libGDX.

Типовим підходом для покращення швидкодії графічних додатків є зменшення кількості OpenGL-запитів, що дозволяє більш оптимально використовувати ресурси GPU. Замість того, щоб для виведення кожної графічної фігури або текстури робити окремий виклик відповідних низькорівневих функцій, можна підготувати та надіслати всі необхідні дані декількома великими порціями. В libGDX для цього використовується клас SpriteBatch, що дозволяє значно спростити частину задач.

Клас SpriteBatch має конструктор, який встановлює максимальну кількість спрайтів, які можуть бути поміщені в буфер перед відсиланням до графічного процесора [3]. Якщо число занадто мале, це призведе до додаткових зверненнями до графічного процесора. Якщо воно занадто велике, то SpriteBatch буде використовувати більше пам'яті, ніж це необхідно.

Клас SpriteBatch надсилає текстуру і координати для малювання кожного трикутника. Він робить збір геометрії без передачі її в графічний процесор. Якщо він отримує текстуру, відмінну від останньої, то він прив'язує останньою текстуру, передаючи зібрану геометрію для малювання, і починає збір геометрії для нової текстури. Клас Texture автоматично декодує файл зображення і завантажує його в пам'ять графічного процесора. Клас TextureRegion дозволяє працювати лише з конкретним регіоном всередині текстури і корисний при малюванні тільки її частини.

Зважаючи на описані вище переваги, а саме: широкий набір підтримуваних платформ, високі показники швидкодії, відкритий код, підтримка великої кількості інструментів та бібліотек, доступ до екосистеми JVM, підтримка RTA, інструментарій для конфігурації проекту, а також відносно легкий спосіб вивчення – libGDX є оптимальним варіантом для створення ігор як для новачків, так і для досвідчених розробників.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. libGDX [Електронний ресурс] – Режим доступу до ресурсу: <https://libgdx.com/>.
2. libgdx/libgdx: Desktop/Android/HTML5/iOS Java game development framework [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/libgdx/libgdx>.
3. Class SpriteBatch [Електронний ресурс] – Режим доступу до ресурсу: <https://libgdx.badlogicgames.com/ci/nightlies/docs/api/com/badlogic/gdx/graphics/g2d/SpriteBatch.html>.