

програмувати на текстових мовах JavaScript і Python, що робить його більш доступним для сучасного учня.

Загалом, використання мікроконтролерів набуло популярності у вбудованих комп'ютерних системах [1], які вирішують певні специфічні завдання, будуються на основі різноманітного асортименту процесорів та архітектур, чутливі до обмежень за вартістю, габаритами, відмовами живлення, тепловиділенню, а також за іншими параметрами, в більшості своїй є системами реального часу, вимагають менше ресурсів для свого функціонування, ніж звичайний комп'ютер.

До наведених переваг можна також віднести використання специфічних інструментів і методів у проектуванні елементів робототехніки і швидке втілення у практичну реалізацію як віртуально у середовищах програмування, так і на практичних матеріальних об'єктах, що надзвичайно стимулює учнів до засвоєння знань з основ робототехніки у процесі програмування micro:bit.

### Список використаних джерел

1. Цідило І. М. Переваги застосування вбудованих комп'ютерних систем над персональними комп'ютерами в АПК/Цідило І. М., Замора Я. П. // «Наука і освіта в інтелектуально-інноваційному розвитку суспільства». Матеріали Міжнародної науково-практичної конференції, присвяченій 60-річчю навчального закладу ВП НУБіП України «Бережанський агротехнічний інститут» – «Бережанський агротехнічний коледж» (16–17 травня 2019 р., Бережани). Тернопіль : ФОП Паляниця В. А., 2019. С. 323–325.
2. Simon Monk. Programming the BBC micro:bit. Getting Started with MicroPython/Simon Monk. McGraw-Hill Education. 2018. 188 p.

## ДЕЯКІ АСПЕКТИ ВИКОРИСТАННЯ ІГРОВОГО ДВИГУНА ДЛЯ РОЗРОБКИ ІГРОВИХ ДОДАТКІВ

### Шмигер Галина Петрівна

кандидат біологічних наук, доцент кафедри інформатики та методики її навчання,  
Тернопільський національний педагогічний університет імені Володимира Гнатюка,  
shmyger@fizmat.tnpu.edu.ua

### Красовський Ігор Володимирович

магістрант спеціальності 014.09 Середня освіта (Інформатика),  
Тернопільський національний педагогічний університет імені Володимира Гнатюка,  
krasovsky @ fizmat.tnpu.edu.ua

Нині стрімко розвивається індустрія комп'ютерних ігор. Розробкою теорії гри, її методологічних основ, з'ясуванням її соціальної природи займаються чимало учених [2; 3].

Зазначимо, що ігри пройшли довгий шлях становлення від найпростіших двоколірних 2D ігор до сучасних 3D ігор з високим рівнем графіки. Центральним програмним компонентом комп'ютерних ігор є ігровий двигун. Він забезпечує основні технології, спрощує розробку і часто дає грі можливість запускатися на декількох платформах, таких як ігрові консолі та настільні операційні системи, наприклад, GNU/Linux, Mac OS X і Microsoft Windows. У розробників ігор з'явилося більше можливостей, актуальною стає неформальна освіта, яку можна

отримати, використовуючи ігрові платформи. Це призводить до нового розуміння ролі університетської освіти і свободи для творчості [2].

Ігровий двигун (англ. Game engine) – це програмна платформа для створення і розробки комп'ютерних ігор або будь-яких інших додатків з графікою, оброблюваною в реальному часі [1].

Ігровий двигун, як правило, складається з таких компонентів:

1. Графічний двигун (англ. Graphics engine) – програмний компонент, основним завданням якого є візуалізація (рендеринг) двомірної або тривимірної комп'ютерної графіки.

2. Фізичний двигун (англ. Physics engine) – програмний компонент, який дозволяє моделювати фізичні закони руху у віртуальному оточенні.

3. Звуковий двигун (англ. Sound/audio engine) – програмний компонент, що відповідає за відтворення звуку (шумове та музичне оформлення, голосів персонажів) в комп'ютерній грі або іншому додатку.

4. Ігровий штучний інтелект (англ. Game artificial intelligence) – набір програмних методик, які використовуються в комп'ютерних іграх для створення ілюзії інтелекту в поведінці персонажів, керованих комп'ютером.

До появи ігрових двигуни нові ігри створювалися з нуля, будучи оптимізовані під цільову платформу. Необхідність використання подібного підходу обумовлювалася обмеженими можливостями апаратного забезпечення того часу, а також тим фактором, що архітектура апаратного забезпечення часто змінювалася.

Зазначимо, що сучасні ігрові двигуни є одними з найскладніших додатків, які коли-небудь створювалися, і включають в себе багато злагоджено працюючих підсистем, що дозволяють ефективно використовувати доступні апаратні засоби для досягнення найбільш повного досвіду взаємодії користувача з системою (user experience) [3].

Безперервний розвиток ігрових двигуни призвів до чіткого поділу між створенням графіки, логіки, дизайну персонажів і рівнів. Для кожного етапу застосовуються окремі засоби розробки.

У своєму розвитку ігрові двигуни, з одного боку, стали набагато більш дружніми до користувача, деякі навіть дозволяють обходитися без навичок програмування. З іншого боку, стало більш широким їх застосування. Тепер вони застосовуються не тільки в індустрії ігор, але і для створення додатків візуалізації технічних процесів, всіляких симуляторів або просто для створення інтерфейсу користувача [2].

Сучасний ігровий двигун є комплексною програмною платформою, що складається з кількох основних компонентів, кожен з яких може бути як безпосередньою частиною рушія, так і поставлятися окремо. До прикладу, розглянемо кожен з використовуваних компонентів.

*Графічний двигун* (англ. Graphics engine) – проміжне програмне забезпечення (англ. Middleware), основним завданням якого є візуалізація комп'ютерної графіки. Графічний двигун визначає можливості по відображенню комп'ютерної графіки, які включають в себе:

1. Доступні режими рендерингу (2D/3D).

2. Підтримувані програмні інтерфейси (OpenGL/DirectX).

3. Різні графічні характеристики: доступну роздільну здатність (кількість точок на одиницю площі), роздільну здатність і фільтрацію текстур, параметри освітлення і тіней.

*Фізичний двигун* (англ. Physics engine) – програмний компонент, який дозволяє моделювати фізичні закони руху реального світу у віртуальному світі, з тим або іншим ступенем наближення до реальності. Як і графічний двигун, ігровий фізичний двигун повинен працювати в режимі реального часу (на відміну, наприклад, від наукового фізичного двигуна). Однак, на відміну від графічних двигунів, фізичний двигун далеко не завжди є частиною ігрового двигуна і часто поставляється окремо. Наприклад, найбільш популярні двигуни є повністю незалежними програмними продуктами [3]:

- NvidiaPhysX (26,8 % ринку); Havok (22,7 % ринку);
- Bullet Physics Library (10,3 % ринку);
- Open Dynamics Engine (4,1 % ринку).

*Звуковий двигун* (англ. Sound/audio engine) – програмний компонент ігрового двигуна, який відповідає за відтворення звуку (шумове та музичне оформлення, голоси персонажів) у комп'ютерній грі або іншому додатку.

Найчастіше є частиною ігрового двигуна, може використовувати такі відомі програмні інтерфейси як OpenAL; DirectSound3d; Environmental Audio Extensions (EAX) та FMOD.

*Ігровий штучний інтелект (ДІВ)* (англ. Game artificial intelligence) – програмний компонент або набір програмних методик, що використовуються в комп'ютерних іграх для створення ілюзії інтелекту в поведінці віртуальних персонажів.

Варто зазначити, що наявність ігрового штучного інтелекту цілковито необхідна в одиночних іграх таких жанрів [4]:

- стратегії як покрокові, так і реального часу;
- шутери (англ. Shooter – «стрілялка»);
- у рольових іграх (RPG).

У той же час багато ігор можуть обійтися і без ДІВ, або він може бути замінений відносно простою емуляцією, серед них:

- багато ігр для декількох гравців;
- аркадні ігри;
- більшість платформерів;
- головоломки;
- музичні ігри.

*Цільові платформи.* Ігровий двигун визначає доступні цільові платформи, для яких з його допомогою можна створити ігри, наприклад, CryEngine 2, за допомогою якого був створений бестселер Crysis, міг бути використаний тільки для створення ігор під платформу Microsoft Windows. Unity і Unreal Engine 4 підтримують майже всі існуючі платформи.

Останнім часом найбільшою популярністю користується ігровий двигун Unity. Він повністю задовольняє нас за своїми базовими характеристиками, до того ж має потужну підтримку товариства, що значно знижує поріг входження.

Unity – ігровий двигун, який зібрав у собі фізичний, графічний двигун, великий набір інструментів та широкий функціонал, гарний баланс між кількістю написаного коду та використанням вбудованого інструментарію для налаштування елементів гри. Зручна візуалізація продукту та чудові можливості відладки як коду так і поведінки гри в цілому.

Отже, Unity на даний час є найпотужнішим засобом для розробки гри серед розглянутих і найкраще його використовувати в розробці великих та середніх проєктів, а також може бути використаний як засіб ознайомлення із специфічними особливостями розробки ігрових додатків у вищих навчальних закладах.

### Список використаних джерел

1. Балик Н. Р., Буяк Б. Б., Габрусев В. Ю. Реалізація game-based learning засобом розробки ігрових додатків Godot. *Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації*: матеріали I Всеукр. наук.-техн. конф. молодих вчених, аспірантів та студентів, Одеса, 25–26 берез. 2021 р. Одес. нац. акад. харч. технологій. Навч.-наук. ін-т комп'ют. систем і технологій «Індустрія 4.0» ім. П. М. Платонова; ред. С. В. Котлик; орг. ком.: Б. В. Єгоров (голова) та ін. Одеса : ОНАХТ, 2021. С. 46–49.

2. Габрусев В. Ю., Вельгач А. В., Кулянда О. О. Дослідження функціональних особливостей рушія UNITY 3D на прикладі реалізації 3D міні-гри. *Науковий часопис національного педагогічного університету імені М. П. Драгоманова*. Серія 2. Комп'ютерноорієнтовані системи навчання. Київ : Вид-во НПУ імені М. П. Драгоманова, 2020, № 22 (29), С. 153–160.

3. A. Pathfinding Project. URL: <http://arongranberg.com/astar/> (дата звернення: 05.11.2022).

4. Helgason David. Another million unity developers in the house. URL: <http://blogs.unity3d.com/2013/07/09/anothermillion-unity-developers-in-the-house> (дата звернення: 02.11.2022).