



PAWEŁ DYMORA ¹, MIROSŁAW MAZUREK ²,
BARTOSZ KOWAL ³

Dydaktyczne aspekty projektowania aplikacji w środowisku Unity 3D

Educational Aspects of Application Design in Unity 3D Environment

¹ Doktor inżynier, Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki, Zakład Systemów Złożonych, Polska

² Doktor inżynier, Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki, Zakład Systemów Złożonych, Polska

³ Magister inżynier, Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki, Zakład Systemów Złożonych, Polska

Streszczenie

W artykule przedstawiono zagadnienia związane z etapami tworzenia aplikacji i gier z przeznaczeniem do wykorzystania w edukacji w oparciu o silnik gier Unity. Skupiono się na możliwościach, jakie daje ten silnik, opisano proces tworzenia prostych gier oraz wykorzystania technologii VR. Zagadnienia te mogą mieć szerokie zastosowanie w procesie dydaktyki przedmiotów informatycznych, szkoleniach itp. W szczególności tematyka odpowiada aktualnym wyzwaniom jak np. Przemysł 4.0.

Słowa kluczowe: Unity, Unreal, silnik gier, grafika generowana komputerowo, Przemysł 4.0

Abstract

The paper presents issues related to the stages of creating applications and games in the Unity game engine for educational purposes. The focus was on the possibilities offered by this engine in the process of creating games and VR environment that could be used in the education process. In particular, the subject corresponds to current challenges such as Industry 4.0.

Keywords: Unity, Unreal, silnik gier, computer generated graphics, Industry 4.0

Wstęp

Już od początku XX w. ludzkość myślała o stworzeniu wirtualnego świata. Na przestrzeni lat 60. XX w. powstały pierwsze wyświetlacze 3D zakładane na głowę, które były prekursorami dzisiejszych gogli VR. Kluczową datą dla prze-

mysłu wirtualnej rzeczywistości był 2016 r. Powstały wtedy pierwsze gogle wysokiej rozdzielczości dedykowane dla konsumentów. Co roku wydawane są nowsze i mocniejsze podzespoły komputerowe mogące obsłużyć aplikacje wymagające dużego stopnia szczegółowości.

Artykuł pokazuje możliwości, jakie dają silniki gier w kontekście wykorzystania w edukacji. Silniki gier to podstawa tworzenia aplikacji i gier eliminująca nakłady pracy związane z programowaniem struktury aplikacji od zera. W opracowaniu porównano najpopularniejsze obecne silniki gier dla przemysłu VR oraz przedstawiono wykorzystanie silnika gier Unity w tworzeniu środowiska 3D, które między innymi może być wykorzystane w procesach technologicznych i produkcyjnych pod ogólnie pojętą koncepcją Przemysłu 4.0

Silniki gier

Silnik gier to w skrócie prosty framework do tworzenia aplikacji i gier. Zapewnia on funkcje, począwszy od fizyki świata aż po sztuczną inteligencję. Silniki gier są odpowiedzialne za renderowanie grafiki, określanie i wykrywanie kolizji, zarządzanie wydajnością, wybór docelowej platformy sprzętowej i wiele innych opcji. Silniki te są bardzo często wykorzystywane ze względu na łatwość tworzenia aplikacji oraz wygodę używania. Zazwyczaj zawierają w sobie pięć głównych elementów (Jason, 2014):

- główne środowisko zawierające logikę,
- silnik graficzny,
- silnik audio,
- silnik fizyki do implementacji „symulowanych” praw fizyki,
- moduł sztucznej inteligencji.

Obecnie dostępne narzędzia i zaangażowanie społeczności programistów w znaczący sposób uprościły proces tworzenie gier. Największym problemem jest jednak wybór odpowiedniego silnika pod odpowiedni projekt. Każdy z silników ma swoje mocne i słabe strony. Poniżej krótko przedstawiono popularne silniki gier 3D.

Unity

Dużą część programistów uważa środowisko Unity za jeden z najłatwiejszych silników gier ze względu na prosty i intuicyjny interfejs. Jedną z głównych funkcji, którą oferuje, jest tworzenie gier dla wielu platform. Korzystając z silnika Unity, można tworzyć gry dla systemów Android, iOS i innych systemów operacyjnych telefonów, w tym PC. Unity ma aktywną społeczność programistów, twórców wtyczek i zasobów, którzy oferują wiele bezpłatnych i niedrogich zasobów do wykorzystania w silniku gry. Kolejną zaletą Unity są darmowe licencje dla szkół, które mogą być wykorzystane do nauki progra-

mowania, szkoleń i prac badawczo-rozwojowych. Na rysunku 1 przedstawia interfejs główny środowiska Unity (Geig, 2013; Jackson, 2015).

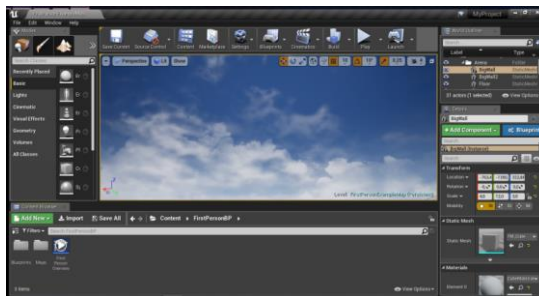


Rysunek 1. Interfejs główny w Unity

Źródło: opracowanie własne.

Unreal

Silnik gier Unreal jest drugim z najbardziej popularnych silników gier pozwalający na tworzenie aplikacji i gier dla osób mających niewielkie doświadczenie z programowaniem. Główną zaletą silnika Unreal są wirtualne plany pozwalające na szybkie tworzenie prototypów, a nawet całych gier i aplikacji za pomocą gotowych planów. Zawiera on w sobie duży zestaw narzędzi i dostępne zasoby tworzone przez społeczność programistów Unreal, zwiększające możliwości samego silnika. To środowisko udostępnia licencje typu *education* dla szkół, pozwalając na rozwój uczniów w dziedzinie programowania gier i tworzenia obiektów 3D. Ponadto szkoły i uczniowie otrzymują regularne aktualizacje, dzięki czemu mogą łatwo śledzić najnowsze osiągnięcia w zakresie gier, zasobów, wizualizacji i produkcji w czasie rzeczywistym. Jedyną wadą tego silnika jest nieprzystosowanie do pracy dla jednej osoby. Duża liczba modułów może przytłoczyć początkującego programistę (Sanders, 2016; Cookson, Dowling Soka, Crumpler, 2016).



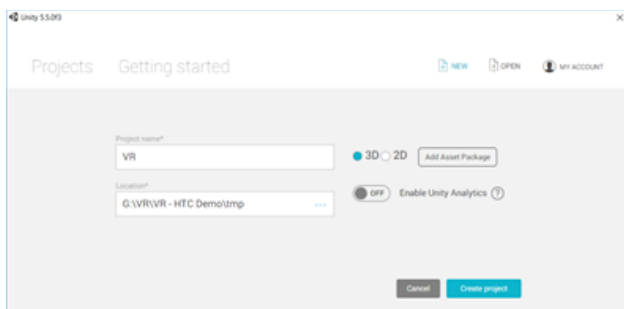
Rysunek 2. Interfejs główny w Unreal Engine

Źródło: opracowanie własne.

Największą różnicą między dwoma silnikami gier jest model licencjonowania gier, które mają zostać opublikowane (nie na cele dydaktyczne). Na rysunku 2 przedstawiono wygląd interfejsu głównego w silniku Unreal. Ceny poszczególnych licencji zależą przede wszystkim od zarobków gier oraz od wybranych funkcji (<https://www.pluralsight.com/blog/film-games/unreal-engine-4-vs-unity-game-engine-best>).

Podstawy tworzenia świata 3D w Unity

Pierwszym etapem w stworzeniu aplikacji z użyciem silnika gier Unity jest stworzenie nowego projektu. Podczas tworzenia należy wybrać odpowiedni tryb aplikacji 2D lub 3D. Dodatkowo jest możliwość automatycznego dodania przypisanych zasobów z własnego konta. Na rysunku 3 przedstawiono etap tworzenia nowego projektu w Unity 3D.



Rysunek 3. Nowy projekt 3D

Źródło: opracowanie własne.

Program zawiera w sobie niezbędne elementy potrzebne do stworzenia dowolnego świata 3D. Głównymi elementami są:

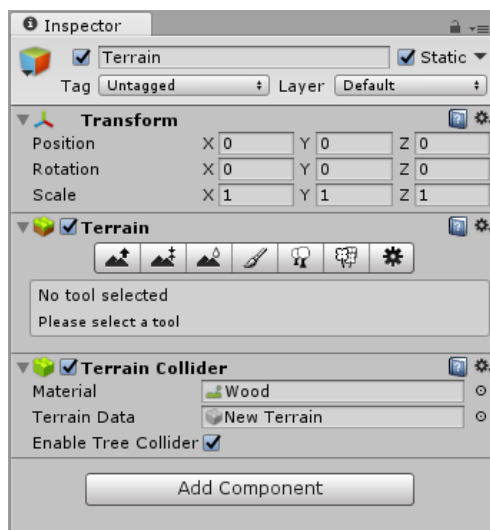
1. **Scene** (Scena). Jest to tzw. *sandbox*. Tutaj umieszcza się wszystkie elementy gry, które tworzą spójną całość.

2. **Game** (Gra). Ten element jest podobny do panelu sceny. Różnicą w tym okienku jest widok – przedstawia tylko to, co widzi kamera gracza w momencie grania. Dokładniej mówiąc – jest to widok z kamery ustawionej jako kamera główna.

3. **Project** (Projekt). Są to wszystkie pliki przypisane do projektu, znajdujące się na dysku. Można je przeglądać, filtrować (do tego służy pasek wyszukiwania u góry) oraz dodawać nowe (przycisk *Create* po lewej).

4. **Hierarchy** (Hierarchia). Tutaj oznaczone są wszystkie elementy składowe będące na scenie.

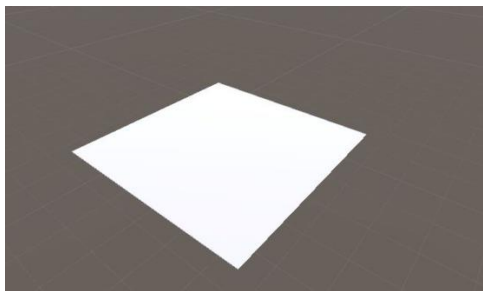
5. **Inspector** (Inspektor). Najważniejszy z paneli (rys. 4). W nim zawierają się wszystkie informacje o danym obiekcie: nazwa, tag, warstwa, na której się znajduje, oraz komponenty ze swoimi parametrami.



Rysunek 4. Panel inspekcji

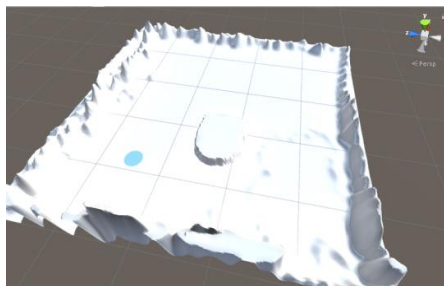
Źródło: opracowanie własne.

Budowę aplikacji najlepiej zacząć od stworzenia terenu, czyli podstawowego miejsca, po którym można się poruszać w świecie. Do tego celu można wykorzystać panel hierarchii, w którym tworzy się nowy obiekt 3D **teren**. Domyślnie stworzy się płaska powierzchnia, którą należy wymodelować. Dokonuje się tego w panelu inspekcji (rys. 5 i 6).



Rysunek 5. Domyślny teren

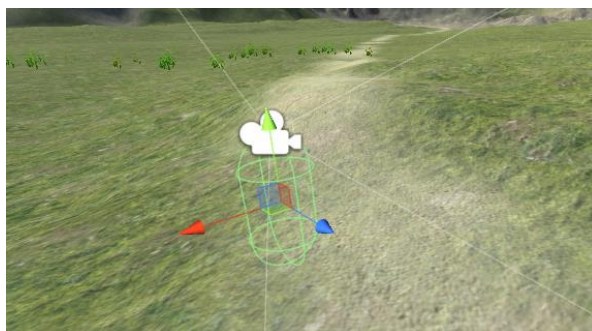
Źródło: opracowanie własne.



Rysunek 6. Teren po modyfikacji

Źródło: opracowanie własne.

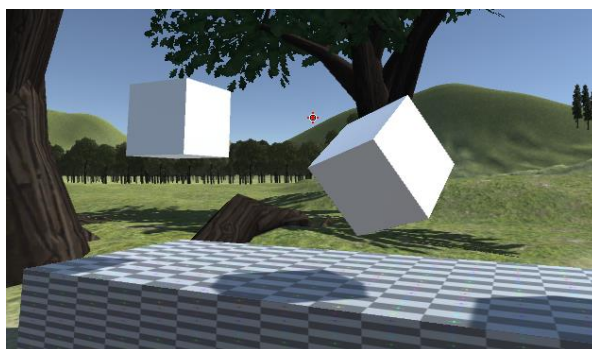
Po utworzeniu i wymodelowaniu terenu następnym krokiem jest dodanie poruszającego się obiektu, np. animacji człowieka – za pomocą przycisków na klawiaturze. Silnik Unity zawiera w sobie już gotowy prefabrykat zawierający całą fizykę oraz skrypty pozwalające na poruszanie się obiektów. Projektantowi gry pozostaje tylko dodać kamerę do tego obiektu (rys. 7).



Rysunek 7. Dodanie kamery do prefabrykatu gracza

Źródło: opracowanie własne.

Kolejnym ważnym etapem podczas projektowania sceny są kolizje z obiektami. Przechodzenie przez takie obiekty, jak drzewa, krzewy, mury, głązy itp., jest niedopuszczalne. Łatwym rozwiązaniem takiego problemu jest dodanie komponentu kolizji do tych obiektów, np. *Capsule Collider* dla drzew, pozwalającego np. na ustawienie promienia kolizji dla tego obiektu. Następnym ze szczególnie istotnych elementów, bez którego żadna z aplikacji i gier nie może się obejść, jest animacja obiektów. Silnik ma już wbudowane narzędzie. Od programisty wymagane jest tylko odpowiednie zmodyfikowanie elementów oraz określenie, jak długo będzie taka animacja działać. Przykładowe dodanie elementów animacji zostało przedstawione na rysunku 8.



Rysunek 8. Podstawowe animacje obiektów – cykliczne poruszanie się obiektów

Źródło: opracowanie własne.

Nie można zapominać o nowej technologii wirtualnej rzeczywistości. Po pobraniu z menedżera zasobów Unity dodatku *SteamVR* można w bardzo łatwy sposób zmodyfikować działający program pod VR. Od użytkownika wymagane jest skopiowanie zasobów *CameraRig* oraz *SteamVR* do własnego pola hie-

rarchii. Dzięki temu na goglach wirtualnej rzeczywistości będzie wyświetlane to samo, co widzi użytkownik na monitorze. Na rysunku 9 przedstawiono implementację technologii VR w projekcie 3D.



Rysunek 9. Obsługa wirtualnej rzeczywistości w projekcie 3D

Źródło: opracowanie własne.

Systemy wirtualnej rzeczywistości dają ogromną liczbę możliwości w edukacji, pozwolą one ożywić program nauczania. Najważniejszą z zalet, jakich dostarcza taki system, jest zachęcenie uczniów do samodzielnego kreowania własnego świata. Nowy sposób przedstawiania informacji stanowi więc uzupełnienie tradycyjnych metod nauczania, zanurzając uczniów w świecie, w którym mogą tworzyć i łączyć się z programem nauczania na zupełnie nowym poziomie.

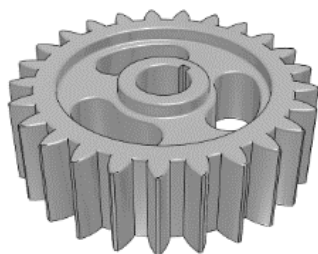
Przed przystąpieniem do tworzenia aplikacji VR należy zadać sobie kluczowe pytania: Na jaką platformę należy ją wydać, jaki silnik gier użyć oraz na jaki sprzęt VR zostanie ona wdrożona – smartfony czy na komputery? Google VR dla smartfonów czy gogle VR dedykowane na komputery klasy PC? Tworząc aplikacje VR, trzeba sprawdzić licencje, otwartość kodu, dostępność bibliotek i zasobów. To właśnie od tych potrzeb zależy, jaki silnik zostanie wybrany. Niezależnie od wyboru platformy VR podstawy aplikacji czy gier tworzy się na tej samej zasadzie, tj. dodając obiekty, tekstury, teren. Dopiero później następuje faktyczne wdrożenie sterowania systemem VR, tj. położenie gogli i wskaźników, interakcja z obiektami, poruszanie się itp. W celach dydaktycznych najlepiej sprawdzają się dedykowane gogle wirtualnej rzeczywistości na komputery klasy PC. Zapewniają one dosyć dobre pole widzenia (110°), wysoką rozdzielczość obrazu (2160×1200 px) oraz dedykowane kontrolery pozwalające na ich szybką integrację z silnikiem gier. Na rysunku 10 przedstawiono przykład przygotowanej aplikacji dla Androida na gogle VR.



Rysunek 10. Stworzona przeglądarka zdjęć 360 przeznaczona na platformę Android

Źródło: opracowanie własne.

System VR można wprowadzić do zajęć związanych z architekturą, np. do tworzenia modeli CAD. Importując modele do systemów wirtualnej rzeczywistości, można w skali rzeczywistej sprawdzić z każdej strony, jak w rzeczywistości wygląda ten obiekt, jak on się zachowuje, jakie siły fizyczne mogą mieć na niego wpływ (rys. 11 i 12).



Rysunek 11. Obiekt stworzony za pomocą CAD

Źródło: opracowanie własne.



Rysunek 12. Zaimportowane obiekty w środowisku VR

Źródło: opracowanie własne.

Podsumowanie

Głównym celem artykułu było przedstawienie możliwości, jakie dają współczesne silniki gier. Dostarczają one kompletne środowisko potrzebne do tworzenia gier i środowiska 3D na wielu platformach. Silniki te posiadają bardzo duże możliwości, w tym implementują aktualne rozwiązania umożliwiające tworzenie wirtualnych i rozszerzonych rzeczywistości, które mogą znaleźć szerokie zastosowanie w Przemysle 4.0 oraz z powodzeniem mogą być wykorzystywane w edukacji. Wadą wykorzystania tych silników jest stopień zaawansowania poszczególnych silników, trudność obsługi oraz typ licencji.

Niemniej silnik graficzny Unity jest dobrym początkiem dla każdej osoby mającej chęć na stworzenie różnych projektów, symulacji, wirtualnego świata, a nie tylko gier.

Literatura

- Cookson, A., Dowling Soka, R., Crumpler, C. (2016). *Unreal Engine 4 Game Development in 24 Hours*. Sams Publishing.
- Geig, M. (2013). *Unity Game Development in 24 Hours*. Sams Publishing.
- <https://www.pluralsight.com/blog/film-games/unreal-engine-4-vs-unity-game-engine-best> (12.10.2018).
- Jackson, S. (2015). *Unity 3D UI Essentials*. Packt Publishing.
- Jason, G. (2014). *Game Engine Architecture*. Taylor & Francis Ltd.
- Sanders, A. (2016). *An Introduction to Unreal Engine 4*. A K Peters/CRC Press.

**Projekt finansowany w ramach programu Ministra Nauki i Szkolnictwa Wyższego pod nazwą „Regionalna Inicjatywa Doskonałości” w latach 2019-2022.
Nr projektu 027/RID/2018/19 kwota finansowania 11 999 900 zł.**

**This work is financed by Polish Ministry of Science and Higher Education under the program “Regional Initiative of Excellence” in 2019–2022.
Project number 027/RID/2018/19, funding amount 11 999 900 PLN.**