



Technologia i rozwiązania

# Unreal Engine

## Nauka pisania gier dla kreatywnych



Joanna Lee

[PACKT] open source\*  
PUBLISHING community experience distilled

Tytuł oryginału: Learning Unreal Engine Game Development

Tłumaczenie: Łukasz Piwko (wstęp, rozdz. 2 – 7), Katarzyna Gotlin (rozdz. 1)

ISBN: 978-83-283-2830-3

Copyright © Packt Publishing 2016.

First published in the English language under the title 'Learning Unreal Engine Game Development – (9781784398156)'.

Polish edition copyright © 2017 by Helion SA  
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:  
<ftp://ftp.helion.pl/przyklady/unreng.zip>

Drogi Czytelniku!  
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres  
<http://helion.pl/user/opinie/unreng>  
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>O autorce</b>	<b>9</b>
<b>O recenzentach</b>	<b>11</b>
<b>Wstęp</b>	<b>13</b>
<b>Rozdział 1. Trochę informacji o silniku Unreal Engine</b>	<b>17</b>
<b>Z czego składa się gra?</b>	<b>17</b>
<b>Co to jest silnik gry?</b>	<b>18</b>
<b>Historia silnika Unreal Engine</b>	<b>18</b>
<b>Wytwarzanie gier</b>	<b>19</b>
Graficy	20
Twórcy filmowi	20
Projektanci dźwięku	20
Scenarzyści	20
Programiści	20
<b>Komponenty Unreal Engine 4</b>	<b>21</b>
Silnik dźwięku	21
Silnik fizyki	21
Silnik grafiki	21
Szkielet rozgrywki	22
Światło i cienie	23
Efekty postprocesowe	23
Sztuczna inteligencja	24
Obsługa różnych systemów i rozgrywek przez internet	24
<b>Niesamowite możliwości edytorów Unreal Engine</b>	<b>24</b>
Unreal Editor	24
Edytor materiałów	25
Edytor dźwięku Sound Cue	27
Edytor Matinee	27
System skryptów Blueprint	28

<b>Programowanie w Unreal Engine</b>	<b>29</b>
Obiekty	29
<b>Przewodnik po edytorze Unreal dla początkujących</b>	<b>30</b>
Menu startowe	30
Przeglądarka projektów	30
Przeglądarka zawartości	31
Pasek narzędzi	32
Widok	32
Lista obiektów	33
Tryb	34
<b>Podsumowanie</b>	<b>34</b>
<b>Rozdział 2. Tworzenie pierwszego poziomu</b>	<b>35</b>
<hr/>	
<b>Przeglądanie gotowych poziomów</b>	<b>36</b>
<b>Tworzenie nowego projektu</b>	<b>36</b>
<b>Poruszanie się po oknie widoku</b>	<b>38</b>
Widoki	38
Klawisze sterowania	39
<b>Tworzenie poziomu z nowej pustej mapy</b>	<b>40</b>
<b>Tworzenie podłoża za pomocą pędzla BSP Box</b>	<b>42</b>
Wskazówka — jak łatwo zaznaczyć obiekt	44
Wskazówka — zmiana trybu widoku dla polepszenia widoczności obiektów	44
<b>Dodawanie światła do poziomu</b>	<b>45</b>
Wskazówka — pozycjonowanie obiektów w poziomie	46
<b>Dodawanie nieba do poziomu</b>	<b>47</b>
<b>Dodawanie punktu początkowego</b>	<b>47</b>
Wskazówka — obracanie obiektów w poziomie	48
<b>Oglądanie utworzonego poziomu</b>	<b>49</b>
<b>Zapisywanie poziomu</b>	<b>50</b>
<b>Ustawianie mapy jako pierwszego poziomu</b>	<b>50</b>
<b>Dodawanie materiału do podłoża</b>	<b>51</b>
<b>Dodawanie ściany</b>	<b>52</b>
<b>Duplikowanie ściany</b>	<b>53</b>
<b>Tworzenie otworu na drzwi</b>	<b>53</b>
<b>Dodawanie materiałów do ścian</b>	<b>56</b>
<b>Zamykanie pomieszczenia</b>	<b>57</b>
<b>Dodawanie własności lub statycznej siatki do pomieszczenia</b>	<b>58</b>
<b>Dodawanie punktu oświetleniowego</b>	<b>59</b>
<b>Wykańczanie pomieszczenia</b>	<b>61</b>
Wskazówka — wyrównywanie do siatki	61
<b>Podsumowanie</b>	<b>63</b>
<b>Rozdział 3. Obiekty gry i ruch</b>	<b>65</b>
<hr/>	
<b>Pędzel BSP</b>	<b>65</b>
Tło	66
Typ pędzla	66
Twardość pędzli	66

<b>Statyczna siatka</b>	<b>67</b>
<b>Pędzel BSP a siatka statyczna</b>	<b>67</b>
<b>Tworzenie ruchomych siatek statycznych</b>	<b>68</b>
<b>Materiały</b>	<b>69</b>
Tworzenie materiału w edytorze Unreal	70
Materiały a tekstury	71
Mapowanie tekstur UV	71
<b>Poziom szczegółowości</b>	<b>73</b>
<b>Kolizje</b>	<b>73</b>
Parametry konfiguracyjne kolizji	74
Interakcje	77
<b>Proces tworzenia siatki statycznej</b>	<b>77</b>
<b>Wprowadzenie do woluminów</b>	<b>78</b>
Wolumin blokujący	78
Wolumin blokujący kamerę	78
Wolumin rozsyłania zdarzeń	79
Wolumin obszaru ruchu postaci	80
Wolumin fizyki	80
<b>Wprowadzenie do Blueprint</b>	<b>83</b>
Level Blueprint	84
<b>Włączanie i wyłączenie światła przy użyciu woluminu rozsyłania zdarzeń</b>	<b>85</b>
<b>Włączanie i wyłączenie światła za pomocą woluminu rozsyłania zdarzeń (opcjonalna technika)</b>	<b>95</b>
<b>Podsumowanie</b>	<b>97</b>
<b>Rozdział 4. Materiał i światło</b>	<b>99</b>
<b>Materiały</b>	<b>99</b>
<b>Edytor materiałów</b>	<b>100</b>
System renderowania	100
Fizyczny model cieniowania	100
Wysokopoziomowy język cieniowania	101
Rozpoczynanie pracy	101
Tworzenie prostego materiału	101
Tworzenie własnego materiału przy użyciu prostych tekstur	107
Przekształcanie poziomu za pomocą własnych materiałów	109
<b>Proces renderowania</b>	<b>111</b>
<b>Shadery</b>	<b>111</b>
<b>Interfejsy API — DirectX i OpenGL</b>	<b>114</b>
DirectX	115
<b>Światła</b>	<b>118</b>
Dodatkowe ustawienia światła punktowego	119
Dodawanie i konfiguracja światła reflektorowego	122
Profile IES	123
Dodawanie i konfigurowanie światła kierunkowego	125
Światła statyczne, nieruchome i ruchome	127
Ćwiczenie — rozbudowa poziomu gry (opcjonalne)	128
<b>Podsumowanie</b>	<b>134</b>

<b>Rozdział 5. Animacja i sztuczna inteligencja</b>	<b>135</b>
<b>Czym jest animacja</b>	<b>136</b>
<b>Animowanie modeli trójwymiarowych</b>	<b>136</b>
Przygotowywanie się do animacji	137
Jak powstaje animacja	138
<b>Co Unreal Engine 4 ma do zaoferowania pod względem animacji w grach</b>	<b>138</b>
Importowanie animacji z programów Maya i 3ds Max	138
Co można zrobić w edytorze Persona	141
Dlaczego należy mieszać animacje	142
<b>Sztuczna inteligencja</b>	<b>155</b>
Drzewa zachowań	156
Ćwiczenie — projektowanie logiki drzewa zachowań	157
Przykład — tworzenie prostego drzewa zachowań	157
Jak zaimplementować drzewo zachowań w Unreal Engine 4	160
Siatka nawigacji	160
Poradnik — tworzenie logiki sztucznej inteligencji	162
Implementacja sztucznej inteligencji w grach	178
<b>Podsumowanie</b>	<b>178</b>
<b>Rozdział 6. System cząsteczkowy i dźwięki</b>	<b>179</b>
<b>Czym jest system cząsteczkowy</b>	<b>179</b>
<b>Badanie istniejącego systemu cząsteczkowego</b>	<b>180</b>
<b>Główne składniki systemu cząsteczkowego</b>	<b>181</b>
Moduły	181
<b>Zasady projektowania systemu cząsteczkowego</b>	<b>182</b>
Badania	182
Iteracyjny proces kreacji pomysłów	183
<b>Przykład — tworzenie systemu cząsteczkowego kominka</b>	<b>183</b>
Dopracowywanie systemu P_Fireplace	187
<b>Odgłosy i muzyka</b>	<b>188</b>
<b>Jak powstają odgłosy i muzyka w grach</b>	<b>189</b>
<b>Jakość dźwięku</b>	<b>189</b>
<b>Jak nagrywa się odgłosy</b>	<b>189</b>
<b>System audio w Unreal</b>	<b>190</b>
<b>Wprowadzanie dźwięków do Unreal</b>	<b>190</b>
Format audio	190
Częstotliwość próbkowania	191
Głębia bitowa	191
Obsługiwane kanały dźwiękowe	192
<b>Formaty dźwięku Unreal i terminologia</b>	<b>192</b>
<b>Edytor Sound Cue</b>	<b>193</b>
Jak uruchomić edytor Sound Cue	193
<b>Ćwiczenie — importowanie dźwięku do edytora Unreal</b>	<b>194</b>
<b>Ćwiczenie — dodawanie własnych dźwięków do poziomu</b>	<b>197</b>
<b>Konfigurowanie edytora Sound Cue</b>	<b>198</b>
<b>Podsumowanie</b>	<b>199</b>

<b>Rozdział 7. Teren i efekty filmowe</b>	<b>201</b>
<b>Podstawowe informacje o budowie terenu</b>	<b>201</b>
Ćwiczenie — tworzenie wzgórz przy użyciu narzędzia Landscape	202
Opcje tworzenia krajobrazu	209
<b>Wprowadzenie do efektów filmowych</b>	<b>210</b>
<b>Dlaczego potrzebujemy scen filmowych</b>	<b>211</b>
<b>Techniki filmowe</b>	<b>211</b>
Funkcje kamery, które można regulować	212
Ruch kamery	213
Ujęcie sceny	215
<b>Edytor Unreal Matinee</b>	<b>217</b>
Ćwiczenie — tworzenie prostej sekwencji matinee	217
Podsumowanie	223
<b>Skorowidz</b>	<b>225</b>





# Trochę informacji o silniku Unreal Engine

Na początku chciałabym podziękować Ci za wybór tej książki. Jestem pewna, że nie możesz się już doczekać, aż stworzysz swoją pierwszą grę. W tym rozdziale opowiem Ci o kluczowych elementach każdej gry. Dowiesz się też, co Unreal Engine 4 ma pod tym względem do zaoferowania.

Tematy, które poruszymy na początek, to:

- Z czego składa się gra?
- Historia silnika Unreal Engine (UE)
- Na czym polega proces tworzenia gier?
- Komponenty i edytory UE

---

## Z czego składa się gra?

Jeśli lubisz grać w gry, to na pewno dobrze wiesz, co się w nich dzieje. Pomyślmy na przykład o prostej strzelance. Kiedy naciskasz lewy przycisk myszy, broń wypala. Widzisz latające wokół naboje, słyszysz dźwięk wystrzału i rozglądasz się, w co trafiłeś. Jeśli strzeliłeś na przykład w ścianę, widzisz, że pojawiły się na niej uszkodzenia.

Jako twórca gier musisz zrozumieć te zdarzenia na bardziej podstawowe czynniki. W najprostszej postaci nasza strzelanka będzie wymagała połączenia przycisku myszy z wyrzuceniem naboju, uruchomieniem pliku dźwiękowego z nagraniem wystrzału, efektem graficznym isker (zwanym **efektem cząsteczek**) wokół lufy i zmianą wyglądu celu.

Kiedy następnym razem będziesz grać w grę, spróbuj rozłożyć ją na czynniki pierwsze. Pomoże Ci to w późniejszej pracy nad własnymi projektami.

Podczas gry mnóstwo rzeczy dzieje się poza ekranem. Za pomocą Unreal Engine będzie Ci dużo łatwiej to opanować, ponieważ interakcje między wieloma komponentami są już wbudowane w silnik. To ogromna oszczędność czasu.

## Co to jest silnik gry?

Silnik zapewnia rozmaite narzędzia i programy, które pomagają w tworzeniu gry. To bardzo duże ułatwienie. Unreal Engine jest jednym z najpopularniejszych obecnie silników na rynku. Jest dostępny za darmo, o ile nie zarabiasz na swojej grze (możesz sprawdzić warunki licencji pod adresem <https://www.unrealengine.com/custom-licensing>). Popularność UE wynika głównie z bardzo dużych możliwości dostosowania każdego elementu do własnych potrzeb, działania w wielu systemach i możliwości stworzenia nawet gry typu AAA. Jeżeli myślisz o rozpoczęciu kariery w tworzeniu gier, UE jest świetnym wyborem do utworzenia własnego portfolio.

## Historia silnika Unreal Engine

Zanim zapoznasz się z funkcjami oferowanymi przez UE, zabiorę Cię w krótką podróż do przeszłości. Opowiem Ci, jak powstał silnik i jak zmieniał się przez lata.

Jeżeli lubisz gry, to na pewno kojarzysz serię gier *Unreal*. Czy wiesz, jak powstała pierwsza gra z tej serii? Inżynierowie z Epic Games stworzyli silnik, który pomógł im w pracy nad nią. Przez lata wraz z powstawaniem kolejnych części silnik rozrastał się o nowe funkcje. Dawało to coraz większe możliwości przy tworzeniu gier.

W 1998 roku pierwsza wersja UE pozwoliła na modyfikację naszej strzelanki. Można było podmienić jej zawartość na własną i dostosować zachowanie **bohaterów niezależnych** (NPC — ang. *non-player characters*, postacie sterowane przez komputer, czasem nazywane *botami*) do swoich potrzeb. Modyfikacje te tworzyło się za pomocą UnrealScriptu. Możliwość gry wieloosobowej została z kolei dodana podczas tworzenia gry sieciowej *Unreal Tournament*. Ta gra pojawiła się w wersjach nie tylko na PC i Maca, ale także na PlayStation 2.

Do 2002 roku UE błyskawicznie się rozwijał. Pojawił się system cząstek do generowania takich efektów jak dym czy mgła, narzędzia do edycji kształtów, silnik fizyki do tworzenia realistycznych interakcji między obiektami (na przykład zderzeń) oraz Matinee, czyli narzędzie do tworzenia krótkich filmów. Zastosowanie tych nowinek można było zobaczyć w *Unreal Championship* oraz *Unreal Tournament 2003*. Pierwsza z wymienionych gier pojawiła się także w wersji na Xboxa, natomiast wieloosobowo można było w nią pograć na Xboxie Live.

Prace nad kolejną grą Epic Games, *Unreal II: The Awakening*, zaowocowały dodaniem możliwości animacji obiektów oraz poprawieniem dotychczasowych funkcji. Łącza internetowe znacząco przyspieszyły, przez co pojawiło się większe zapotrzebowanie na gry sieciowe. Gra *Unreal Tournament 2004* umożliwiała takie rozgrywki. Pojawiła się możliwość tworzenia pojazdów i wielkich pól bitwy, a także rozwinęła się jakość gry przez internet. Prace nad *Unreal Champion 2* w 2005 roku wzmocniły możliwości UE na konsolach Xbox. Została też dodana możliwość ustawienia kamery z perspektywy trzeciej osoby. To otworzyło drzwi do tworzenia nowych typów gier, takich jak dobrze wszystkim znane *Gears of War*. W 2006 roku pojawiła się trzecia wersja silnika, Unreal Engine 3.

Silnik rozwijał się dalej. Zastosowanie bibliotek DirectX 9/10 sprawiło, że obiekty w grach wyglądały dużo bardziej realistycznie. Pojawił się też **Kismet**, system wizualnej obsługi skryptów, który pozwala na tworzenie logiki za pomocą łączenia bloków kodu bez konieczności zagłębiania się w jego pojedyncze linie. W UE3 poprawiono organizację oświetlenia i materiałów, zmieniono moduł fizyki oraz dostosowano silnik do konsoli, w tym Xbox360 i Playstation 3. Efekty zmian można zobaczyć w grze *Gears of War 2* (2008). Kilka lat później, w 2013 roku, pojawiła się natomiast gra *Gears of War Judgement*.

Tymczasem programiści Epic Games intensywnie pracowali nad rozwojem gier sieciowych. W 2009 roku pojawiło się oprogramowanie o nazwie Atlas Technology, które wraz z UE pozwalało na tworzenie wieloosobowych gier sieciowych (**MMO** lub **MMOG** — ang. *Massively Multiplayer Online Games*).

Z biegiem lat zwiększyło się zapotrzebowanie na gry mobilne. UE3 nie pozostawał w tyle i pojawiło się w nim wsparcie dla różnych systemów mobilnych. Szerokie spektrum możliwości tej wersji silnika sprawiło, że stała się najpopularniejsza spośród wszystkich wydań UE. Wielu twórców gier używa tej wersji do dziś.

UE3 zdominował rynek na osiem lat, dopóki w 2014 roku nie pojawił się UE4, a z nim kolejne znaczące zmiany. Przede wszystkim Kismet został zastąpiony przez nowy moduł o nazwie **Blueprint**, a o pozostałych możliwościach UE4 opowiem w dalszej części rozdziału.

## Wytwarzanie gier

Każde studio ma swoją własną ścieżkę rozwoju gier. Zwykle jednak gra przechodzi przez kilka etapów. Te etapy to planowanie, produkcja i postprodukcja. Większość czasu zajmuje oczywiście produkcja.

Tworzenie gry jest procesem iteracyjnym. Rozpoczyna go pojawienie się pomysłu. Trzeba sprawdzić, czy ten pomysł spodoba się odbiorcom. W tym celu tworzy się prototypy. Kolejne iteracje, które zmieniają prototyp w pełnowymiarową grę, mogą ciągnąć się tygodniami, miesiącami, a nawet latami. Zespół produkcyjny intensywnie pracuje w tym czasie nad rozwojem. W skład zespołu wchodzi graficy (2D i 3D, animatorzy), twórcy filmowi, projektanci dźwięku, scenarzyści i programiści.

## Graficy

Zajmują się wszystkim, co jest widoczne na ekranie — od przycisków menu do drzew na mapie. Część z nich tworzy modele trójwymiarowe, część zajmuje się animacją. Dzięki grafikom gra wygląda estetycznie i realistycznie.

Do tworzenia grafiki graficy używają takich programów jak 3DMax, Maya i MODO, więc muszą umieć importować obrazy i modele do UE4. Po umieszczeniu modeli w grze często testują ich zachowanie przy użyciu modułu Blueprint.

## Twórcy filmowi

Używają swoich artystycznych zdolności do tworzenia krótkich scenek filmowych. Zwykle pracują przy użyciu Matinee.

## Projektanci dźwięku

Mają bardzo czuły słuch i zwykle także wykształcenie muzyczne. Pracują nad stworzeniem odpowiednich dźwięków do gry i przyłączeniem ich w odpowiednim momencie. Podczas pracy z UE4 zwykle używają edytora Sound Cue.

## Scenarzyści

Decydują, co będzie tematem gry, jaki będzie miała charakter i jakie będą możliwe wydarzenia. W fazie planowania większość czasu spędzają na dyskusjach, prezentacjach i dokumentacji. W fazie produkcji kontrolują prototypy poziomów, żeby były jak najbardziej zgodne z założeniem. Często dopracowują kolejne etapy gry w edytorze Unreal Editor.

## Programiści

Decydują, jaka technologia i oprogramowanie będą potrzebne do stworzenia gry. Muszą dopilnować, żeby poszczególne programy były kompatybilne ze sobą. W fazie produkcji piszą kod, który sprawia, że obiekty zaprojektowane przez grafików zachowują się zgodnie z założeniami scenarzystów. Nie są bezpośrednio zaangażowani w koncepcję gry, ale programują jej zasady i funkcje. Część programistów zajmuje się także tworzeniem narzędzi potrzebnych do dalszej produkcji gry. Z kolei gry z mocno zaawansowaną grafiką często wymagają optymalizacji, czym zajmuje się specjalna podgrupa programistów.

Programiści przez większość czasu pracują w Visual Studio, gdzie piszą w języku C++. Mogą też rozwijać funkcjonalności samego silnika UE4, jeśli potrzebują czegoś specjalnego do stworzonej właśnie gry.

## Komponenty Unreal Engine 4

Silnik UE pomaga w tworzeniu gier. Składa się z kilku komponentów, które razem sprawiają, że gra działa. Zawiera mnóstwo narzędzi i edytorów, które pozwalają zapanować na obiektami i manipulować nimi.

Komponenty UE to silniki dźwięku, fizyki i grafiki, szkielet rozgrywki oraz moduł sieciowy.

### Silnik dźwięku

Dzięki niemu możemy usłyszeć dźwięki i muzykę w grze. Pozwala na odtworzenie plików, które wprowadzą odpowiedni nastrój i dodadzą grze realizmu. Jest wiele zastosowań dźwięków. Część z nich słyszymy cały czas w tle. Inne są uruchamiane wyłącznie podczas konkretnych wydarzeń.

Jeśli akcja gry toczy się w lesie, potrzebna nam będzie kombinacja śpiewu ptaków, szumu drzew i szelestu liści. To będą dźwięki tła, które będziemy odtwarzać cały czas, kiedy postać znajdzie się w lesie. Kiedy będzie chodzić, przydadzą się nam odgłosy kroków. Dźwięki, których użyjemy jednorazowo, na przykład eksplozji budynku, mogą być przypisane do konkretnego zdarzenia. Za wyzwalanie dźwięków w Unreal jest odpowiedzialne **Sound Cue**.

### Silnik fizyki

Prawdziwym światem rządzą prawa fizyki. Obiekty poruszają się i zderzają zgodnie z zasadami dynamiki Newtona. Przyciąganie opisuje prawo grawitacji i ogólna teoria względności Einsteina. Obiekty w grze powinny jak najbardziej przypominać rzeczywiste. Do tego potrzebujemy silnika, który jest programistycznym odzwierciedleniem rzeczywistych praw fizyki. Silnik Unreal używa silnika PhysX stworzonego przez Nvidię. Wewnątrz niego wykonują się złożone obliczenia, które symulują rzeczywiste zjawiska, takie jak zderzenia i dynamika płynów. Obecność silnika fizyki pozwala skupić się nam na tworzeniu właściwej gry, zamiast na programowaniu złożonych interakcji.

### Silnik grafiki

Aby obraz pokazał się na ekranie, musi zostać najpierw odpowiednio wygenerowany. Silnik grafiki odpowiada za przetworzenie takich informacji jak kolor, tekstura, geometria, oświetlenie i cienie. Musi także uwzględniać punkt widzenia oraz interakcje między obiektami, na przykład rzucanie cienia.

Do tego wszystkiego potrzebne są bardzo skomplikowane obliczenia. W efekcie uzyskujemy informacje dla każdego piksela na ekranie. To od silnika grafiki zależy, jak bardzo realistycznie będzie wyglądać nasza scena. Silnik Unreal pozwala na uzyskanie fotorealistycznej jakości obrazu. Potrafi optymalizować wygląd sceny i przeliczać wartości oświetlenia w czasie rzeczywistym.

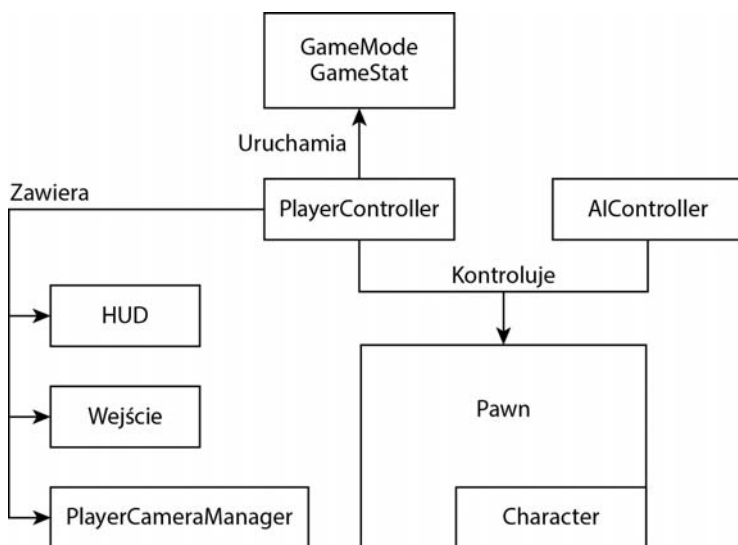
Silnik może być użyty na dowolnej platformie (PC, Xbox, PlayStation, urządzenia mobilne). Wspiera biblioteki DirectX 11/12, OpenGL oraz JavaScript/WebGL.

## Szkielet rozgrywki

Unreal Engine posiada system, który przekształca sygnały wejścia (np. wciśnięcie klawisza) w działania wykonywane przez postać w grze. Sygnały i ogólne zasady funkcjonowania gry konfigurujemy w szkielecie rozgrywki. Zawiera on funkcje, które kontrolują przebieg gry i sprawiają, że akcja dzieje się zgodnie z naszymi zasadami.

Ważnymi elementami tego komponentu są **HUD** (ang. *heads-up display* — pasek wyświetlający przebieg gry) oraz **interfejs użytkownika**. Dzięki nim gracz uzyskuje informacje potrzebne w czasie rozgrywki. Klasy, które kontrolują przebieg gry, to `GameMode` (tryb gry), `GameState` (stan gry) oraz `PlayerState` (stan gracza). Postaciami zarządzają klasy `PlayerController` (jeśli steruje nimi gracz) oraz `AIController` (jeśli steruje nimi komputer). Niezależnie od tego, kto kontroluje postać, należy ona do klasy bazowej `Pawn` (postać). Klasa `Character` (osoba) jest podklasą klasy `Pawn`, nastawioną na obiekty poruszające się w osi poziomej, na przykład na człowieka.

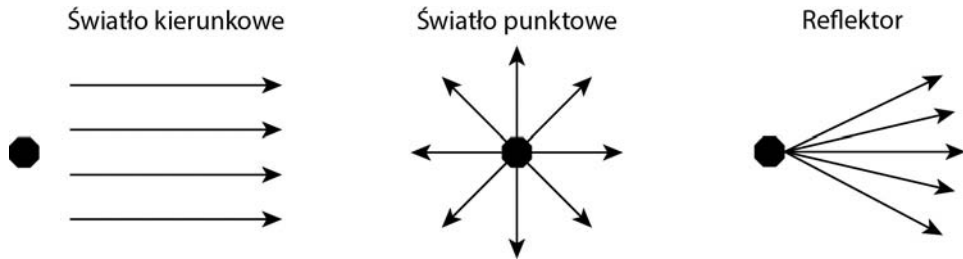
Dzięki szkieletowi rozgrywki i kontrolerom możemy w pełni zaplanować zachowanie i możliwości postępowania gracza. Działanie szkieletu przedstawia poniższy obrazek.



## Światło i cienie

Oświetlenie jest bardzo ważne przy tworzeniu gry. Można go użyć na wiele sposobów, na przykład do stworzenia odpowiedniego nastroju lub do skupienia uwagi gracza na wskazanym obiekcie. Unreal Engine 4 zapewnia zestaw światel, które można umieścić w dowolnym miejscu gry. Są to: **światło kierunkowe** (*Directional Light*), **światło punktowe** (*Point Light*), **reflektor** (*Spot Light*) oraz **światło dzienne** (*Sky Light*).

Światło kierunkowe emituje promienie równoległe do siebie. Światło punktowe działa jak żarówka — promienie padają z jednego miejsca we wszystkich kierunkach. Reflektor emituje światło w kształcie stożka, a światło dzienne naśladuje naturalne, słoneczne oświetlenie.



Dzięki użyciu światel możemy uzyskać na naszej mapie realistycznie wyglądające cienie. Wybór rodzaju oświetlenia wpływa zarówno na nastrój gry, jak i na czas renderowania sceny, co przekłada się na liczbę klatek na sekundę.

W świecie gry możemy wyróżnić dwa rodzaje cieni: statyczne i dynamiczne. Styczne mogą być wkomponowane w scenę, co daje nam dużą oszczędność czasu. Dynamiczne zmieniają się w trakcie grania i wymagają bardziej skomplikowanych obliczeń. Dokładniejsze informacje na temat światel i cieni znajdziesz w rozdziale 4.

## Efekty postprocesowe

Są to efekty, które dodajemy na sam koniec, żeby poprawić jakość wygenerowanej sceny. Unreal Engine 4 oferuje duży wybór tego typu efektów. Dodanie ich podkreśli wygląd stworzonego przez nas świata.

Do wyboru mamy między innymi rendering z użyciem szerokiego zakresu dynamicznego (HDRR). Sprawia on, że jasne obiekty stają się bardzo jasne, a ciemne — bardzo ciemne. Mimo takiego zwiększenia kontrastu na obiektach wciąż widoczne są szczegóły.

Inne efekty to czasowe wygładzanie (TXAA), rozświetlenie, korekcja barwna, głębia ostrości, automatyczna ekspozycja, flara, materiały postprocesowe, aberracja chromatyczna, rozmyte odbicie oraz winietowanie. Twórcy gier zwykle używają tych efektów, ale jeśli zajdzie taka potrzeba, mogą je wyłączyć. Zwykle dzieje się tak z powodu nadmiernego zużycia pamięci.

## Sztuczna inteligencja

Jeśli koncepcja **sztucznej inteligencji** (AI) jest jeszcze dla Ciebie obca, możesz przyjąć, że jest to zestaw algorytmów, które naśladują prawdziwe myślenie. Sztuczna inteligencja daje obiektom w grze mózg, dzięki któremu mogą myśleć i podejmować samodzielne decyzje.

Bohaterowie niezależni są świetnym przykładem sztucznej inteligencji w grach. Kiedy gramy, oni reagują odpowiednio do naszych posunięć. Mogą znaleźć sobie miejsce na przyczajenie się i atak. Mogą uciekać, chować się i bronić, kiedy to my ich atakujemy.

UE4 zapewnia całkiem niezłą sztuczną inteligencję na podstawowym poziomie. Można oczywiście rozwinąć jej możliwości i dodać brakujące nam funkcje. Szczegóły działania AI znajdziesz w rozdziale 5.

## Obsługa różnych systemów i rozgrywek przez internet

Jeśli stworzysz grę za pomocą Unreal Engine 4, będziesz mógł grać w nią w niemal dowolnym systemie. Obecnie obsługiwane systemy to Windows, Linux, iOS, Android, a także przeglądarki internetowe. Niedawno dodano również możliwość gry przez Universal Windows Platform (UWP). Dzięki specjalnemu modułowi możesz też zagrać sieciowo przez Facebooka, Steama, Xboxa Live i inne systemy.

## Niesamowite możliwości edytorów Unreal Engine

Dowiedziałeś się już trochę o komponentach UE. Teraz nadszedł czas na zapoznanie się z edytorami. Mają one rozmaite funkcje, które pomogą Ci stworzyć własną grę.

### Unreal Editor

UE ma kilka dostępnych edytorów. Podstawowy to Unreal Editor, który służy także jako przejście do innych modułów, takich jak materiały i Blueprint.

Unreal Editor to graficzny edytor, który składa się z różnych widoków i okienek. Dzięki nim możemy importować, organizować, edytować i zmieniać zachowania obiektów w grze. Po przejściu do głębiej położonych modułów możemy zająć się bardziej szczegółowymi właściwościami obiektu.

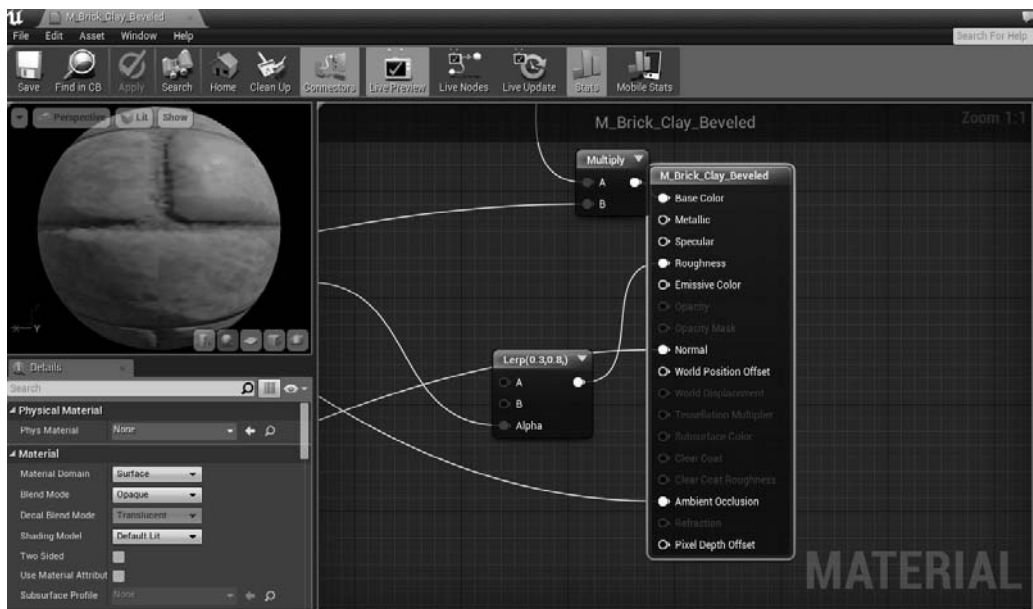
Unreal Editor to doskonałe narzędzie dla scenarzystów. Mogą oni ustawiać i kontrolować w nim obiekty bez konieczności dokonywania zmian w kodzie.



## Edytor materiałów

Materiały i cieniowanie pozwalają na nadanie obiektom unikalnego wyglądu. Unreal Engine 4 wykorzystuje cieniowanie oparte na fizyce (PBS). Daje to twórcom dużo większą kontrolę nad wyglądem i charakterem obiektu. Cieniowanie oparte na fizyce skupia się w dużej mierze na oddziaływaniach pomiędzy światłem a powierzchnią. Powiązanie mikroszczegółów powierzchni i odbicia światła pozwala na uzyskanie bardzo naturalnego wyglądu obiektu.

W przeszłości cieniowanie oparte było na innego rodzaju algorytmach. W UE4 mamy już możliwość osiągnięcia bardziej przewidywalnych i spójnych rezultatów poprzez wzajemne regulowanie algorytmów materiałów i cieniowania. Więcej na ten temat znajdziesz w rozdziale 4. Poniższy obrazek przedstawia edytor materiałów w UE4.



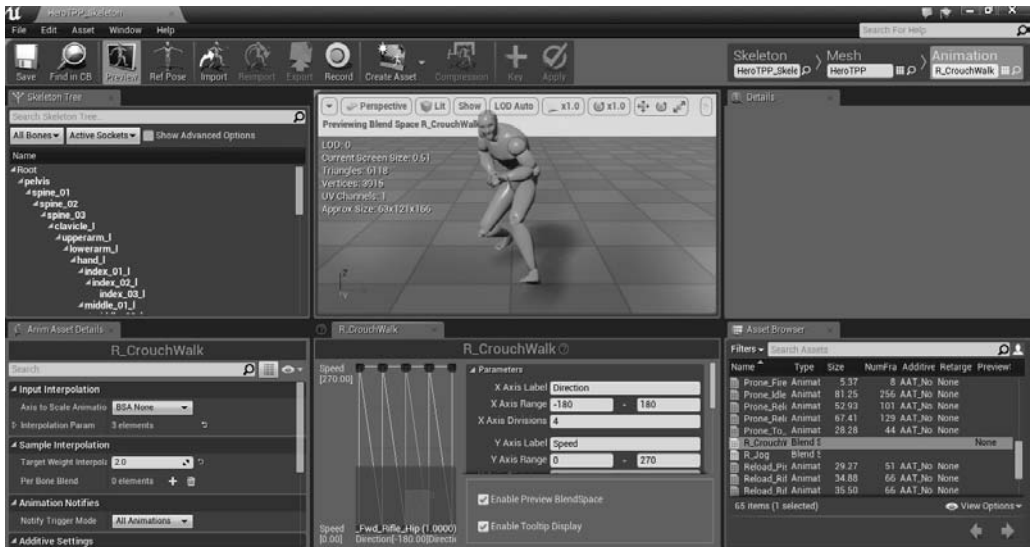
## System cząsteczek Cascade

System cząsteczek pozwala na wygenerowanie takich efektów jak dym, iskry lub ogień. Każdej z cząsteczek można dostosować rozmiar, kolor i teksturę. Można też kontrolować interakcje pomiędzy nimi, żeby zachowywały się tak, jak w prawdziwym życiu. Poniższy obrazek przedstawia edytor Cascade.



## System animacji Persona

Pozwala na zaprojektowanie szkieletu i ciała obiektu i kontrolę nad nimi. Możliwy jest także podgląd animacji oraz ustawienie przejść pomiędzy kluczowymi klatkami. Właściwości fizyki i kolizji można dostosować poprzez narzędzie **Physics Asset Tool (PhAT)**. Poniższy obrazek przedstawia system animacji Persona.

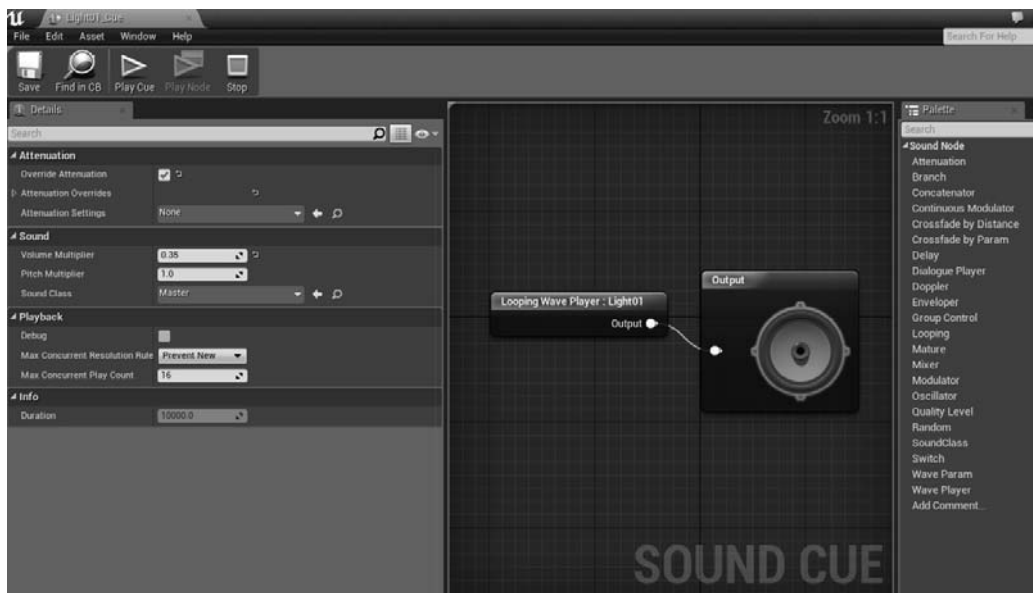


## Mapy — tworzenie świata i roślinności

Unreal Engine zapewnia narzędzia do rzeźbienia i malowania, które pozwolą Ci na stworzenie pięknych widoków. Wyjątkowo dokładny **poziom szczegółów (LOD** — ang. *level of detail*) i dobrze zorganizowane zarządzanie pamięcią ułatwiają skalowanie i kształtowanie dużych obszarów terenu. Edytor Foliage pomoże Ci narysować realistyczną trawę, liście, śnieg i piasek.

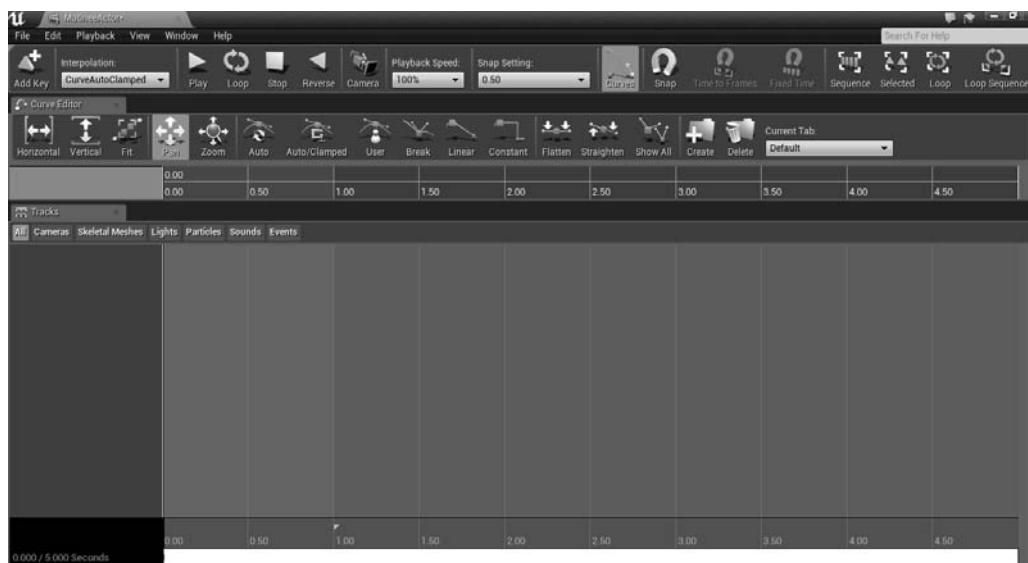
## Edytor dźwięku Sound Cue

Do zarządzania dźwiękami przyda Ci się edytor Sound Cue. Za odtwarzanie plików dźwiękowych odpowiedzialny jest system sygnałów **Sound Cues**. Edytor pozwala nie tylko na uruchamianie i zatrzymywanie dźwięków, ale też daje Ci możliwość ich zapętlenia, nasilania i wyciszenia. Poniższy rysunek przedstawia edytor Sound Cue w UE4.



## Edytor Matinee

Zestaw narzędzi Matinee umożliwia tworzenie krótkich scenek i filmików. Takie nagrania mogą Ci się przydać jako intro do kolejnego poziomu gry, opowiedzenia historii bohatera albo nawet jako reklamówka całej gry. Na poniższym obrazku możesz zobaczyć edytor Matinee.

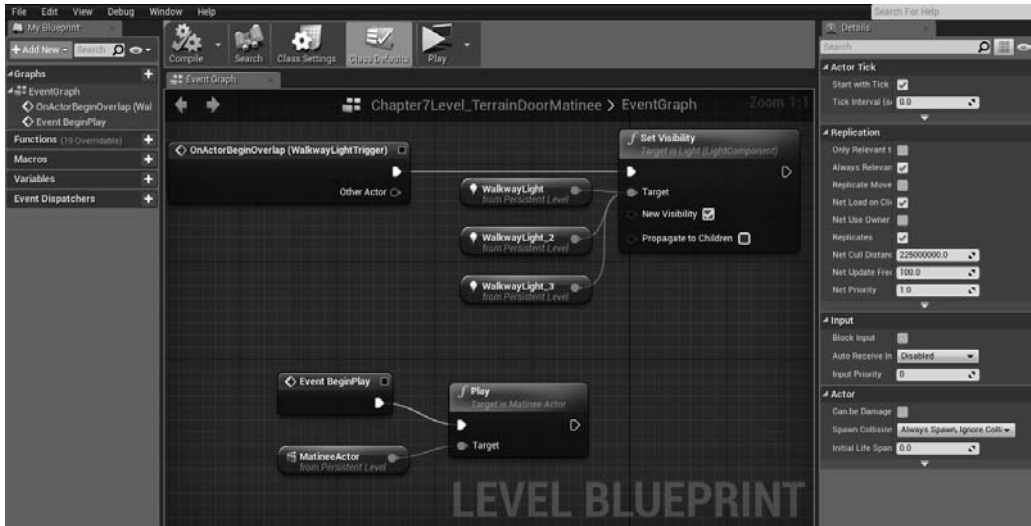


## System skryptów Blueprint

Blueprint to nowa, rewolucyjna funkcja w Unreal Engine 4. Jeżeli znasz już Unreal Engine 3, pomyśl o niej jako o bardziej rozbudowanej wersji Kismet połączonej z funkcjami Prefab. Blueprint rozszerza standardowe pisanie skryptów o możliwość wizualnego łączenia ze sobą bloków (diagramów połączonych liniami). Jeśli nie chcesz pisać i kompilować kodu, to Blueprint jest stworzony właśnie dla Ciebie. Jeżeli jesteś grafikiem lub scenarzystą i chcesz na szybko przetestować wygląd lub zachowanie obiektów bez konieczności zagłębiania się w kod, to Blueprint również Ci na to pozwoli. Bardzo fajną możliwością edytora jest tworzenie zmiennych przez kliknięcie obiektu i wybranie opcji *Create Variable* (utwórz zmienną). Możesz wtedy robić podobne rzeczy jak programista bez grzebania się w kodzie.

Jeśli chcesz zdebugować logikę ułożoną z klocków Blueprint, możesz to zrobić — podczas wykonywania kolejnych kroków programu bloki będą się podświetlać w odpowiedniej kolejności. Można również ustawić pułapkę w odpowiednim miejscu i zatrzymać wykonywanie programu właśnie tam.

Poniższy obrazek przedstawia edytor Blueprint.



## Programowanie w Unreal Engine

Dostęp do kodu źródłowego UE pozwala Ci zrobić z grą niemal wszystko, co sobie zamarysz. Możesz rozszerzyć istniejące funkcje lub dopisać swoje. Kiedy nauczysz się działania UE od środka, otworzą się przed Tobą niesamowite możliwości.

Unreal Engine posiada również przydatne funkcje do debugowania programów. Jedną z nich to **Hot Reload**. Po dopisaniu zmian w C++ zmienia się od razu zachowanie gry. Jeśli chcesz szybko wprowadzić zmiany w kodzie, kliknij wybrany obiekt, a następnie odpowiednią funkcję w panelu *Code View* (podgląd kodu). Zostaniesz od razu przeniesiony do tej samej funkcji w Visual Studio.

Unreal Engine umożliwia również ustawienie kontroli wersji w projekcie.

## Obiekty

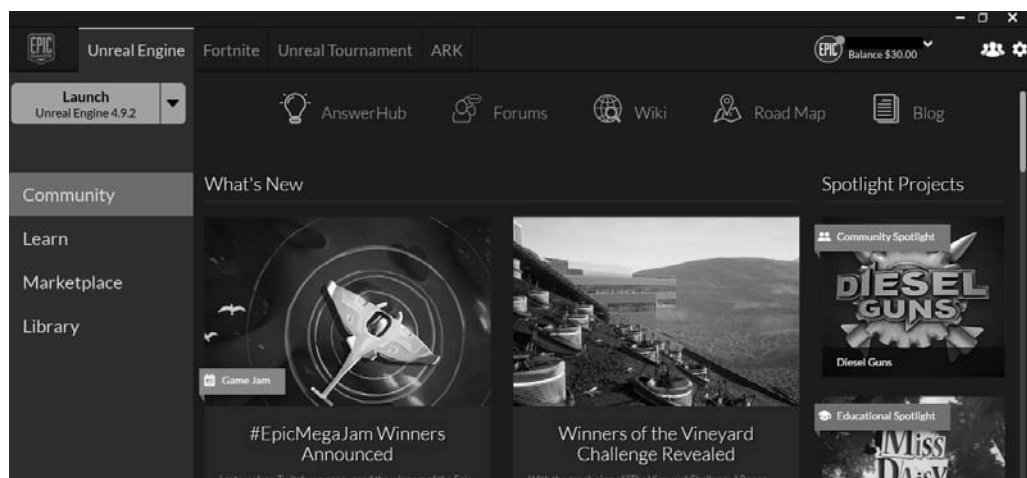
Wszystkie obiekty w grze należą do bazowej klasy Actor (aktor). Aby zapewnić więcej właściwości i funkcji, klasa Actor została rozszerzona o kilka bardziej rozbudowanych podklas. Można powiedzieć, że Actor działa jak kontener zawierający wyspecjalizowane obiekty zwane **komponentami**. Kombinacja możliwości komponentów daje klasie Actor konkretne możliwości.

# Przewodnik po edytorze Unreal dla początkujących

W tej części pokrótce przedstawię, co ciekawego możemy zrobić w edytorze Unreal Editor. Opowiem Ci, jak używać poszczególnych okien do stworzenia własnej gry.

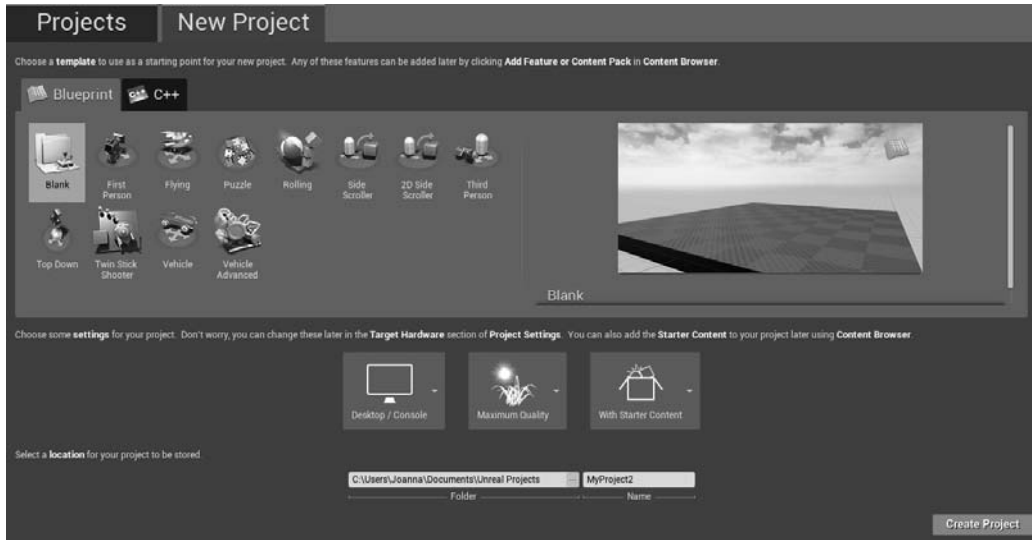
## Menu startowe

Kiedy uruchamiasz Unreal Engine, menu startowe to pierwsza rzecz, którą widzisz. Jest ono bardzo proste wizualnie i łatwe w obsłudze. Duży, rozwijany przycisk umożliwia wybranie wersji UE, której chcesz użyć, i projektów, które chcesz otworzyć. Możesz też zajrzeć do Marketplace, czyli wielkiej biblioteki zawierającej próbki gier stworzonych przez innych użytkowników. Niektóre są za darmo, za inne musisz zapłacić. Menu startowe zawiera także wiadomości od Epic, które pozwolą Ci być na bieżąco z nowościami i zmianami. Na rysunku poniżej możesz zobaczyć, jak wygląda menu startowe.



## Przeglądarka projektów

Po wybraniu właściwej wersji Unreal Engine pojawia się przeglądarka projektów. Pozwala ona na stworzenie kolejnego poziomu z istniejącego szablonu. To znaczy, że nie musisz tworzyć gry całkowicie od zera. Możesz wybrać na przykład strzelankę pierwszo- lub trzecioosobową albo przesuwającą się w bok platformówkę 2D/3D. Możesz edytować grę zarówno w Blueprint, jak i w C++. Co więcej, w zakładce *New Project* (nowy projekt) możesz wybrać docelową platformę dla swojej gry (PC/mobilna), jakość obrazu oraz to, czy chcesz mieć w swoim projekcie domyślną zawartość zapewnioną przez Unreal, czy nie. Poniższy obrazek przedstawia przeglądarkę projektów.



## Przeglądarka zawartości

Kiedy uruchomisz Unreal Editor, na samym początku zobaczysz zestaw różnych okien i paneli. Jednym z nich jest **przeglądarka zawartości**. To takie okno, w którym znajdziesz wszystkie obiekty istniejące w danej grze. Są one podzielone na takie kategorie jak dźwięki, materiały, animacje, efekty cząsteczek i inne. Znajduje się tam także przycisk *Import*, dzięki któremu możesz dołączyć do gry obiekty stworzone w innych programach. Poniższy rysunek przedstawia domyślne położenie przeglądarki zawartości.



## Pasek narzędzi

Pasek narzędzi to wstążka umożliwiająca szybki dostęp do narzędzi i edytorów. Domyślnie zawiera on ikony modułu Blueprint i Matinee, a także szybkiego uruchomienia gry. Jeżeli chcesz, możesz dodać tam wszystko, co jest Ci potrzebne. Poniższy obrazek przedstawia domyślny wygląd paska narzędzi.



## Widok

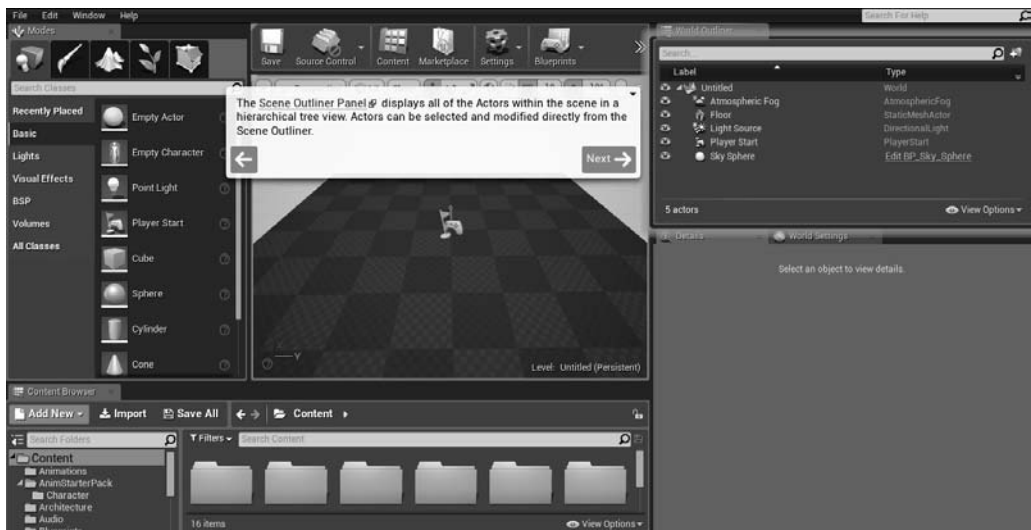
Jest to okno, w którym możesz zobaczyć, co się dzieje w grze. Jeżeli wybrałeś jeden z szablonów dostępnych w zakładce *New Project*, to zauważysz, że kamera jest ustawiona zgodnie z jego domyślnymi parametrami. W oknie widoku możesz umieszczać i obracać swoje obiekty. Kiedy klikniesz *Play* na pasku narzędzi, okno zmienia się w działającą planszę gry, w której będziesz mógł dokonywać zmian na bieżąco. Na poniższym rysunku okno widoku jest podświetlone.





## Lista obiektów

Lista ta zawiera wszystkie obiekty, które aktualnie umieszczone są na scenie. Możesz tworzyć w niej foldery i zmieniać nazwy obiektów na własne (aby móc je łatwo odnajdywać). Możesz także wybrać wiele obiektów i zmienić je jednocześnie. Poniższy rysunek pokazuje edytor z podświetloną listą obiektów.



## Tryb

W tym miejscu wybierasz, co chcesz zrobić z obiektami w swojej grze. Masz do wyboru tryb obiektu, malowania, krajobrazu, roślinności i geometrii. Tryb obiektu służy do wstawiania obiektów do gry, tryb malowania do edycji krawędzi i materiałów, tryby krajobrazu i roślinności do tworzenia naturalnie wyglądających map. Tryb geometrii pozwala na modyfikację obiektów. Podświetlony obszar poniższego rysunku pokazuje okno trybów.



## Podsumowanie

W tym rozdziale omówiliśmy pokrótce, czym jest silnik gier Unreal Engine 4, i przedstawiliśmy jego historię. Dowiedziałeś się też, jak tworzy się gry i jakie osoby są w to zaangażowane. Potem opowiedzieliśmy o różnych komponentach Unreal Engine i o tym, co można z nimi zrobić. Na końcu poznałeś edytory dostępne w UE4, które pomagają dostosować każdy element gry do własnych potrzeb.

W kolejnych rozdziałach omówimy szczegółowo kolejne moduły UE4. W następnym rozdziale będą to podstawowe funkcje edytora Unreal, dzięki którym będziesz mógł zacząć tworzyć swoją własną grę.

# Skorowidz

## A

aktor dźwięku otoczenia, 193  
animacja, 135  
animacje mieszane, 143, 146  
animowanie modeli trójwymiarowych, 136  
audio, 190

## B

badanie systemu cząsteczkowego, 180  
Blueprint, 83  
budowa terenu, 201

## C

cienie, 23  
cieniowanie, 100, 101  
czułość próbkowania, 191

## D

DirectX, 115  
dodawanie  
  dźwięków, 197  
  materiałów, 51, 56  
  nieba, 47  
  podstawowej animacji, 167  
  punktu oświetleniowego, 59  
  punktu początkowego, 47  
  ściany, 52  
  światła, 45  
  własności, 58  
dolka, 214  
drzewa zachowań, 156, 160, 171  
duplikowanie ściany, 53

## E

edytor  
  materiałów, 25, 100  
  Persona, 141  
  Sound Cue, 27, 193, 198  
  Unreal, 194  
    Editor, 30  
    Engine, 24  
    Matine, 27e, 217  
edytowanie emitera płomieni, 187  
efekty  
  filmowe, 210  
  postprocesowe, 23  
emiter  
  płomieni, 187  
  systemu, 187

## F

fale dźwiękowe, 192  
format audio, 190  
  Unreal, 192  
funkcje kamery, 212

## G

głębia  
  bitowa, 191  
  ostrości, 213  
gra, 17  
graf zdarzeń, 168  
graficy, 20

**H**

HUD, heads-up display, 22

**I**

implementacja sztucznej inteligencji, 178

importowanie  
animacji, 138  
dźwięku, 194

interfejsy API, 114

**J**

jakość dźwięku, 189

**K**

kadrowanie, 216

kamera, 213

kanały dźwiękowe, 192

kąt nachylenia

w pionie, 214

w poziomie, 214

klawisze sterowania, 39

kolizje, 73

komponenty, 29

Unreal Engine 4, 21

konfigurowanie

edytora Sound, 198

siatki szablonu postaci, 164

światła, 122

kierunkowego, 125

nieba, 126

kontroler sztucznej inteligencji, 163, 165

**L**

logika

drzewa zachowań, 157

sztucznej inteligencji, 162

**M**

mapa, 40, 50

oświetlenia, 127

mapowanie

cieni, 127

tekstur UV, 71

materiały, 69, 99

model cieniowania, 100

modulacja impulsowo-kodowa, 190

muzyka, 188

**N**

nagrywanie odgłosów, 189

narzędzia Landscape, 202

niebo, 47

**O**

obiekty, 29

obracanie obiektów, 48

odgłosy, 188

oglądanie poziomu, 49

okno widoku, 38, 42

opcje tworzenia krajobrazu, 209

oświetlenie, 216

bezpośrednie, 127

pośrednie, 127

ruchome, 128

statyczne, 127

**P**

pakiet animacji, 139

parametry konfiguracyjne kolizji, 74

pedestal, 215

pędzel

BSP, 65, 67

BSP Box, 42

pole widzenia, 212

poziom szczegółowości, 73

poziomy, 35, 36

pozycjonowanie obiektów, 46

prędkość ruchu, 169

profile IES, 123

program

3ds Max, 138

Marketplace, 139

Maya, 138

programiści, 20

programowanie w Unreal Engine, 29

projektanci dźwięku, 20

próbkowanie, 191

przekształcanie poziomu, 109

punkt początkowy, 47

**R**

reflektor, 23  
 renderowanie, 100, 111  
 ruch kamery, 213

**S**

scena, 215  
 scenarzyści, 20  
 sekwencja matinee, 217  
 sekwencje animacyjne, 138  
 shadery, 111  
 siatka
 

- nawigacji, 160
- statyczna, 58, 67, 77
- ruchoma, 68

 silnik
 

- dźwięku, 21
- fizyki, 21
- grafiki, 21
- gry, 18
- Unreal Engine, 17, 18

 składniki systemu cząsteczkowego, 181  
 sygnały dźwiękowe, 192, 196  
 system
 

- animacji Persona, 26
- cząsteczek Cascade, 25
- cząsteczkowy, 179
  - moduły, 181
  - tworzenie systemu, 183
  - zasady projektowania, 182
- P\_Fireplace, 187
- renderowania, 100
- skryptów Blueprint, 28

 szablon, 165  
 szczegółowość, 73  
 szkielet rozgrywki, 22  
 sztuczna inteligencja, 24, 155

**Ś**

światło, 85, 95, 99, 118
 

- dzienne, 23
- kierunkowe, 23
- nieba, 126
- nieruchome, 127
- punktowe, 23, 119

reflektorowe, 122  
 ruchome, 127  
 statyczne, 127

**T**

techniki filmowe, 211  
 tekstury, 71, 107  
 teren, 201  
 tło, 66  
 twardość pędzli, 66  
 tworzenie
 

- animacji mieszanej, 143
- drzewa zachowań, 157, 171
- kontrolera sztucznej inteligencji, 163
- krajobrazu, 209
- logiki sztucznej inteligencji, 162
- materiału, 70, 101, 107
- otworu na drzwi, 53
- pierwszego poziomu, 35, 40
- podłoża, 42
- postaci Blueprint, 163
- projektu, 36
- siatki nawigacji, 160
- siatki statycznej, 77
- świata i roślinności, 27

 twórcy filmowi, 20  
 typ pędzla, 66  
 typy ujęć, 216

**U**

ujęcia, 216  
 Unreal, 190, 194
 

- Editor, 24
  - lista obiektów, 33
  - menu startowe, 30
  - pasek narzędzi, 32
  - przeglądarka projektów, 30
  - przeglądarka zawartości, 31
  - tryb, 34
  - widok, 32
- Engine 4, 138
- Matinee, 217

 ustawianie mapy, 50  
 używanie drzewa zachowań, 173

## W

widoczność obiektów, 44  
widoki, 38  
wolumin, 78  
  audio, 83  
  blokujący, 78  
  blokujący kamerę, 78  
  fizyki, 80  
  oświetlenia, 83  
  postprodukcji, 83  
  rozsyłania zdarzeń, 79, 85, 95  
  selekcji według odległości, 82  
  strumieniowania poziomym, 82  
wykańczanie pomieszczenia, 61  
wyłączanie światła, 95  
wyrównywanie do siatki, 61  
wytwarzanie gier, 19

## Z

zadanie  
  Move To, 177  
  Wait, 172  
zamykanie pomieszczenia, 57  
zapisywanie poziomym, 50  
zasady kadrowania, 216  
zaznaczanie obiektów, 44  
zmiana trybu widoku, 44

# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

# Unreal Engine

## Nauka pisania gier dla kreatywnych

Wśród graczy są tacy, którzy codziennie większość swojego czasu poświęcają rozgrywkom i testowaniu nowości. Nieliczni jednak potrafią pisać własne gry. Jesteś pasjonatem? Wykorzystaj to! Sprawdź, co oferuje Unreal Engine — intuicyjne, wszechstronne narzędzie, dzięki któremu szybko zaczniesz pisać profesjonalne gry o różnorodnej tematyce. Wystarczy Twoja kreatywność — resztę zapewni Unreal Engine!

Niniejsza książka jest przeznaczona dla początkujących twórców gier, którzy postanowili nauczyć się technologii Unreal Engine. Autorka rozpoczyna od przedstawienia prostych pomysłów i projektów, które spróbujesz samodzielnie rozwinąć i dokończyć. Następnie prezentuje podstawowe wiadomości na temat Unreal Engine, dzięki czemu możesz stworzyć od podstaw prostą grę. W dalszych rozdziałach poznasz kolejne tajniki tworzenia gier z wykorzystaniem aktorów, animacji, efektów itd. Po lekturze tej książki każdy powinien umieć napisać własną grę w Unreal Engine.

**Unreal Engine — Twoje narzędzie do pisania świetnych gier!**

### W książce znajdziesz:

- podstawowe wiadomości o silnikach gier i historię Unreal Engine
- zasady obsługi edytora Unreal Engine 4 i tworzenia pomieszczeń w poziomie gry
- informacje o podstawowych strukturach obiektów w grze
- sposoby tworzenia interaktywnych obiektów przy użyciu szablonów
- podstawowe informacje na temat renderowania grafiki

**Joanna Lee** — od blisko dziesięciu lat zajmuje się projektowaniem i programowaniem gier komputerowych. Biegłe posługuje się wieloma różnymi silnikami gier. Unreal Engine zna od 2005 roku. Jest również znana jako recenzentka książek i filmów na temat CryEngine 4.

**[PACKT]** open source  
PUBLISHING community experience distilled

**Helion**

32960 numer katalogowy

księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne

☎ **0 801 339900**

☎ **0 601 339900**

Sprawdź najnowsze promocje:

● <http://helion.pl/promocje>

Książki najchętniej czytane:

● <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

● <http://helion.pl/novowosci>

Helion SA  
ul. Kościuszki 1c, 44-100 Gliwice  
tel.: 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-283-2830-3



9 788328 328303

Informatyka w najlepszym wydaniu

cena: 57,00 zł