



Tworzenie gier wideo

Kompendium wiedzy



Tworzenie gier wideo

Kompendium
wiedzy

Spis treści

I. Wstęp – czym jest gra?

II. Typologia gier

III. Proces powstawania gier

- Pomysł na grę

- Kreowanie wizji

- Zbieranie feedbacku

- Projektowanie rozgrywki

- Prototypowanie

- Roadmap

- Realizacja projektu

- Polish

IV. Narzędzia wykorzystywane przy tworzeniu gier

- Projektowanie

- Programowanie

- Tworzenie grafiki

- Silniki gier

- Współczesne silniki

V. Podsumowanie

Wstęp

Czym jest gra?

Słowo “gra” nie jest pojęciem jednoznacznym – i nie mówimy tu nawet o znaczeniach związanych z grą aktorską. W klasycznym ujęciu, jak podaje słownik języka polskiego PWN, jest to “zabawa towarzyska prowadzona według pewnych zasad” lub “rozgrywka prowadzona między zawodnikami lub zespołami według zasad określonych regulaminem danej dyscypliny”. Definicje te znajdują faktycznie zastosowanie w odniesieniu do gier analogowych – np. planszowych i karcianych – oraz do rozgrywek sportowych.

Jednak ustalone definicje nie nadążają za zmiennością gier – dla przykładu, pasjans układany przez jedną osobę nie zawiera ani rywalizacji, ani elementu towarzyskiego, a grą przecież jest. Co więcej, w minionych dekadach zrodził się całkowicie nowy rodzaj gier, które również wymykają się dotychczasowej klasyfikacji: gry wideo. Mimo że klasyczne definicje ich nie obejmują, w dalszym ciągu są to gry, które w dodatku zyskały ogromną popularność na całym świecie oraz przerodziły się w prężnie rozwijającą się branżę, a budżety niektórych gier zbliżają się do kosztów produkcji największych hitów kinowych.

Brak aktualnej definicji gier to nie tylko problem klasyfikacji naukowej. Oznacza on także, że twórcy gier są skazani na błędne-

nie po omacku w poszukiwaniu przepisu na “grę idealną” – lub przynajmniej taką, którą odbiorca będzie miło wspominał i której koszt produkcji się zwróci. Podobnie jak ma to miejsce w innych dziedzinach kreatywnych, wielu twórców nigdy tego sekretnego przepisu nie odnajduje.

Z tego powodu podejmowane są próby podania aktualnej definicji gry, która odzwierciedlałaby zmienne realia. Jak mówi na przykład Sid Meyer, jeden z najbardziej uznanych projektantów gier wideo, “gra to seria interesujących decyzji”. Zrodziła się również silna potrzeba rozwiązań, które usprawnią proces produkcji gier, by zminimalizować ryzyko porażki. W ostatnich latach, w oparciu o dekady doświadczeń, proces ten zaczął się krystalizować. Dzięki temu twórcy będą mogli skupić się na tym, co w ich pracy najlepsze: na kreatywnej twórczości.

Poniższe kompendium przedstawi pokrótce proces produkcji gier wideo oraz najważniejsze ich style i gatunki. Omówione zostaną również podstawowe narzędzia, z których korzystają twórcy oraz problemy, z jakimi muszą się mierzyć.

Typologia gier

Choć gry wideo są stosunkowo młodym medium, doczekały się już własnej klasyfikacji. Podstawowym kryterium podziału gier wideo jest ich gatunek. Wśród najbardziej popularnych gatunków gier wyróżniamy między innymi:

- **Gry akcji** – są to gry, w których główną rolę pełni szybkość rozgrywki, a gracz mierzy się przede wszystkim z koniecznością podejmowania szybkich decyzji i wykonywania określonych akcji w krótkim czasie, korzystając z własnego refleksu.

- **GRY PRZYGODOWE** – gry, w których gracz odgrywa rolę protagonisty opowieści. Rozwój fabuła odbywa się, kiedy gracz eksploduje kolejne etapy świata przedstawionego i rozwiązuje napotkane zagadki. Gatunek ten wyróżnia duży nacisk na narrację, przez co produkcje te są bliższe powieściom i filmom niż innym grom wideo.

- **GRY FABULARNE¹** – jest to gatunek, w których gracz kontroluje poczynania pojedynczego bohatera lub grupy postaci w trakcie przygód rozgrywających się w fikcyjnym świecie. Jedną z najważniejszych cech gier fabularnych jest rozwój postaci, który może odbywać się poprzez mechanikę gry (na przykład zdobywanie nowych umiejętności w walce czy w użytkowaniu maszyn) lub fabularnie, na podstawie dokonywanych

przez gracza wyborów. Często gry tego typu pozwalają graczowi stworzyć własną postać, ustalając na przykład jej płeć, cechy zewnętrzne i pochodzenie, choć w niektórych grach fabularnych gracz wciela się w postać zdefiniowaną przez twórców – ma to miejsce szczególnie w przypadku adaptacji powieści i filmów. Większość gier fabularnych korzysta z klasycznej struktury scenariusza, gdzie punktem kulminacyjnym jest konfrontacja z przeciwnikiem, do której postać przygotowuje się przez większość gry.

- **GRY LOGICZNE** – są to gry, w których gracz musi wykorzystać umiejętność logicznego myślenia, aby przejść kolejne poziomy. Gatunek ten jest szczególnie popularny na platformach mobilnych.

- **GRY SPORTOWE** – przedstawiają one wybraną dyscyplinę sportu, istniejącą w prawdziwym świecie lub fikcyjną, pozwalając graczom wcielić się w uczestniczących w niej zawodników. Gry tego typu wymagają od gracza nie tylko zręczności, ale i strategicznego myślenia. W obrębie gier sportowych występują także gry menedżerskie², w których gracz nie występuje w roli zawodnika, lecz kierownika zespołu, który zarządza całą drużyną sportowców w trakcie zmagania ligowych czy pucharów. Wiele gier sportowych korzysta z licencji istnie-

1. Ang. Role-Playing Games, w skrócie RPG.

2 Ang. Manager Game.



jących rozgrywek, na przykład pucharów FIFA i UEFA w przypadku gier symulujących piłkę nożną.

- **GRY WYŚCIGOWE** – to gatunek, w którym gracz steruje pojazdem naziemnym, wodnym lub powietrznym, biorąc udział w wyścigach z innymi pojazdami. Podobnie jak w przypadku gier sportowych, mogą być to pojazdy istniejące w rzeczywistości lub całkowicie fikcyjne. Wiele gier korzysta też z prawdziwych licencji pojazdów, szczególnie samochodów (zarówno seryjnych, jak i wyścigowych) oraz serii wyścigowych odbywających się w rzeczywistości, jak na przykład seria F1. Ponadto, do gier wyścigowych zalicza się podgatunek Simracing. Są to gry, które stawiają na jak najwierniejsze odwzorowanie warunków panujących na torze wyścigowym lub trasie rajdu przełajowego, fizyki prowadzenia samochodu

wyścigowego, eksploatacji i uszkodzeń pojazdów oraz przebiegu zawodów, wprowadzając oprócz samych wyścigów obecność sesji treningowych, kwalifikacji czy wizyt w pit-stopie. W przypadku zaawansowanych symulatorów jazdy, do pełnej satysfakcji wymagany jest dedykowany kontroler, który odzwierciedla kierownicę wraz z pedałami.

- **SYMULATORY** – jest to bardzo różnorodna kategoria gier, które łączy fakt dokładnego odwzorowania wybranej dziedziny życia lub aktywności. Mogą być to na przykład symulatory dyscyplin sportowych, lotów, pola walki, rozwoju miasta czy po prostu życia poszczególnych jednostek ludzkich. Wiele gier z tego gatunku to komercyjne wersje symulatorów tworzonych na potrzeby szkoleniowe, na przykład dla wojska lub ośrodków kształcących pilotów.

Innym kryterium podziału gier jest ilość graczy, którzy mogą jednocześnie brać udział w rozgrywce. W tym ujęciu występuje podział na:

- **SINGLEPLAYER** – gry, w których bierze udział tylko jeden użytkownik. Gracz mierzy się w tym przypadku z zadaniami i przeciwnościami wprowadzonymi przez twórców, które często wykorzystują inne postaci sterowane przez algorytmy sztucznej inteligencji.

- **MULTIPLAYER** – gry, w które może grać wielu użytkowników, jednocześnie w czasie rzeczywistym lub na zmianę w systemie turowym. W części gier multiplayer gracze rywalizują ze sobą, w innych zaś współpracują. Zależnie od gry, może być to multiplayer lokalny, co oznacza, że gracze korzystają z tego samego urządzenia (na przykład komputera osobistego lub konsoli do gier), albo multiplayer online – w tym przypadku gracze łączą się do serwera gry poprzez łącze internetowe lub sieć lokalną. Obecność trybu multiplayer w grze może w znacznym stopniu przedłużyć jej żywotność i zwiększyć dochodowość, jednak jest to rozwiązanie bardziej zaawansowane technologicznie i zazwyczaj bardziej kosztowne, niż stworzenie gry singleplayer. Gra może oferować wyłącznie tryb multiplayer lub też udostępnić go jako opcjonalny tryb, występujący jako dodatek do scenariusza singleplayer.

- **MMO³** – jest to rodzaj gry multiplayer online, w którym w tym samym momencie do serwera gry łączą się setki, a nawet

3. Ang. Massively-Multiplayer Online Game.

tysiące graczy. Bardzo często gry MMO oferują duży świat o otwartej strukturze, w którym postępy dokonane przez graczy oraz ich wpływ na otoczenie są trwałe. Gry MMO najczęściej funkcjonują w formie gry-usługi⁴, a użytkownicy wiążą się z nimi na długi czas, sięgający nieraz tysięcy godzin. Otwiera przed twórcami znacznie większe możliwości kapitalizacji produkcji, niż ma to miejsce w przypadku innych gier. Z tego powodu MMO zalicza się do najbardziej dochodowych gatunków gier, lecz też najtrudniejszych w realizacji i najbardziej kosztownych – ich budżety potrafią sięgać setek milionów dolarów.

Kolejnym istotnym kryterium podziału gier jest platforma docelowa. Wyróżnia się tu przede wszystkim:

- **GRY NA PC⁵** – platforma ta pozwala użytkownikowi korzystać z praktycznie dowolnego kontrolera, od klawiatury i myszki, poprzez gamepady⁶, aż po kierownice czy specjalistyczne kontrolery do symulatorów lotu. Sprawia to, że PC jest najbardziej uniwersalną platformą, na której występuje największa różnorodność dostępnych gier. Jest to także platforma o największych możliwościach sprzętowych. Z drugiej strony, dzięki modularnej budowie komputerów występuje niezliczona liczba konfiguracji sprzętu. Przysparza to wiele problemów

4. Ang. Game-as-a-Service.

5. Ang. Personal Computer – komputer osobisty.

6. Dedykowany kontroler do gier, który zawiera zestaw przycisków, cyfrowy krzyżak do sterowania w dwóch osiach (czasem pozwalający również wybrać kierunki pośrednie) oraz, najczęściej, także gałki analogowe o niezależnych osiach i swobodnym kącie obrotu.



twórcom, którzy muszą zadbać o to, by na każdej konfiguracji spełniającej ustalone wymagania sprzętowe gra działała poprawnie.

- **GRY NA KONSOLE** – platforma szczególnie popularna w zakresie gier z segmentu premium⁷. W chwili obecnej liczą się konsole *Microsoft Xbox One*, *Sony PlayStation 4* oraz *Nintendo Switch*. Jest to również dość uniwersalna platforma, lecz o znacznie mniejszej różnorodności sprzętu – każde urządzenie występuje w zaledwie kilku wariantach, których możliwości są ustalone ogólnie przez producenta. Sprawia to, że twórcy muszą podporządkować się w cało-

7. Klasyczny model dystrybucji gier, w którym użytkownik nabywa jednorazowo licencję gry – w przeciwieństwie do coraz bardziej powszechnego modelu *Free to Play*, w którym sama gra dostępna jest za darmo, a zarobek twórców pochodzi z mikrotransakcji dokonywanych przez graczy w trakcie rozgrywki.

ści istniejącym ograniczeniom technicznym tych urządzeń, jednak w zamian za to nie muszą troszczyć się o poprawne działanie na wielu wariantach konfiguracji. Ponadto, rzadkością jest sterowanie konsolą za pomocą klawiatury i myszki, a zamiast tego najczęściej stosuje się gamepady. Z tego powodu niektóre gatunki gier, które są popularne na PC, praktycznie nie występują na konsolach lub muszą być w dużym stopniu zmodyfikowane, aby dostosować sterowanie do gamepada.

- **URZĄDZENIA MOBILNE** – przede wszystkim smartfony i tablety z systemem *Android* lub *iOS*. Z racji ogromnej powszechności tych urządzeń, jest to platforma, z której korzystają nie tylko dedykowani gracze, ale i użytkownicy z segmentu *Casual Ga-*

ming⁸. Gry mobilne, z racji ograniczonych możliwości sprzętowych, są w ogromnej większości prostsze w konstrukcji i mniej rozbudowane, niż gry na urządzenia stacjonarne – dotyczy się to również jakości oprawy wizualnej. Do kategorii tej zaliczają się także *handheldy*⁹ – ukazują się na nich głównie adaptacje gier z konsol stacjonarnych, dostosowane do ograniczonych możliwości sprzętowych tych urządzeń.

Gry możemy także rozróżnić na podstawie obranego przez twórców stylu graficznego.

8. Są to gry przeznaczone dla szerokiej publiczności, która niekoniecznie identyfikuje się z kulturą graczy, przeważnie o prostych zasadach i łatwej obsłudze, jako że większość grupy docelowej stanowią osoby starsze.

9. Ang. potoczna nazwa przenośnej konsoli do gier, jak np. PS Vita czy Nintendo 3DS – niektóre z obecnie produkowanych handheldów dysponują mocą obliczeniową dorównującą poprzednim generacjom konsol stacjonarnych

Do najczęściej spotykanych nurtów zaliczają się:

- **REALIZM** – styl, w którym warstwa graficzna ma za zadanie odzwierciedlać świat rzeczywisty. Co istotne, grafika danej produkcji może być uznawana za realistyczną pomimo stosowania znacznych uproszczeń graficznych, jeśli tylko efekt finalny jest wiarygodny. Styl ten jest bardzo popularny w grach wysokobudżetowych, ale także w symulacjach czy w grach fabularnych, gdzie duże znaczenie odgrywa możliwość wczucia się w rolę bohatera. Przy odpowiednim wykonaniu, gry z realistyczną grafiką ocierają się o fotorealizm. Jednak olbrzymią ich wadą jest fakt, iż wskutek nieustannego rozwoju możliwości komputerów i konsol, grafika realistyczna starzeje się w przeciągu zaledwie paru lat od wydania gry.



Rys. 1. Styl graficzny – realizm; Forza Horizon 4

- **STYLIZACJA** – jest to zgoła odmienne podejście. W przeciwieństwie do realizmu, gry stylizowane nie mają za zadanie wiernie oddać rzeczywistości, lecz przedstawić jej artystyczną interpretację z wykorzystaniem grafiki prostej, lecz atrakcyjnej dzięki zabiegom stylizacyjnym. Dzięki temu gry stylizowane starzeją się znacznie wolniej od gier realistycznych, a także mają znacznie mniejsze od nich wymagania sprzętowe.

- **STYL KRESKÓWKOWY** – jak sama nazwa wskazuje, styl ten jest zbliżony do estetyki kreskówek i komiksów. Jest on stosowany z powodzeniem w grach przeznaczonych dla młodszych odbiorców, choć korzystają z niego także produkcje bardziej dojrzałe.

Należy także wziąć pod uwagę rodzaj grafiki zastosowany w grze. Wyróżniamy tutaj **GRAFIKĘ 2D**, w której obraz składa się z dwuwymiarowych obrazów oraz gry z **GRAFIKĄ 3D**, w której świat przedstawiony stworzony jest z trójwymiarowych modeli. W większości przypadków grafika 2D jest tańsza w produkcji, jednak może być przyczyną wielu ograniczeń.

Na przykład, brak trzeciego wymiaru sprawia, że poruszanie “do przodu” jest niemożliwe, dlatego często twórcy takich gier decydują się na sterowanie wyłącznie w poziomie lub w pionie. Istotną zaletą grafiki 2D jest niskie zapotrzebowanie na moc obliczeniową, dzięki czemu technologia ta sprawdza się doskonale na platformach o gorszych osiągnięciach, takich jak smartfony.



Rys. 2. Styl graficzny – stylizacja; SuperHot



Rys. 3. Styl graficzny – kreskówka; South Park: Kijek Prawdy



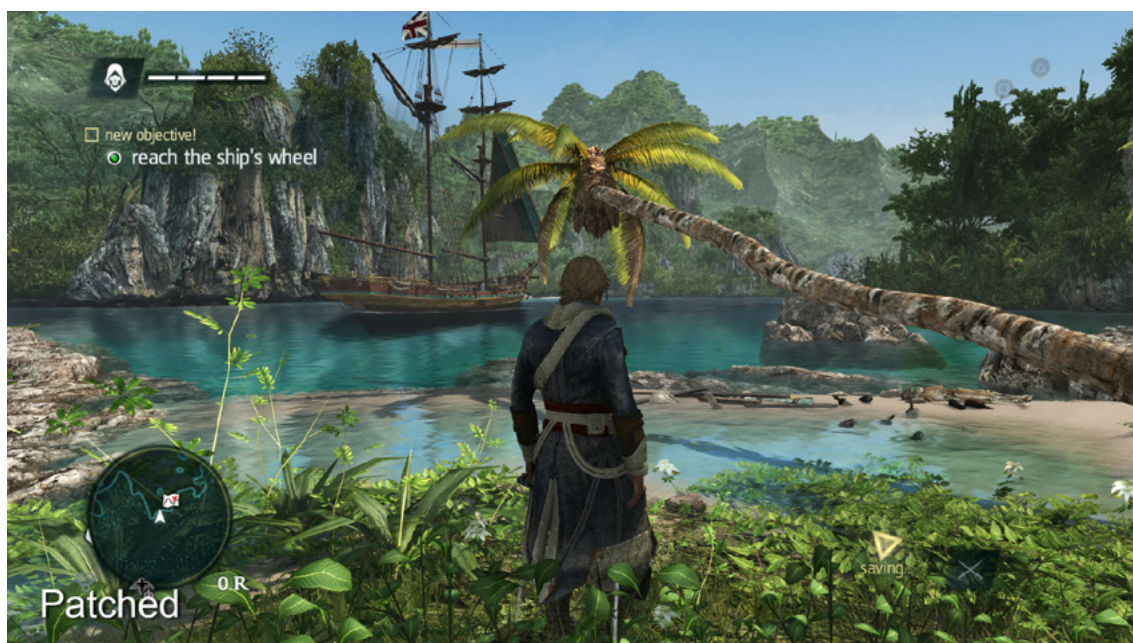
Rys. 4. Perspektywa kamery – FPP; Counter Strike: Global Offensive

Kolejną istotną kwestią jest perspektywa, z której obserwujemy przebieg gry. Najczęściej spotykane z nich to:

- **PERSPEKTYWA PIERWSZOOSOBOWA** – forma prezentacji gry, w której obserwujemy wydarzenia oczami jej bohatera. Widok ten ułatwia odbiorcy wczucie się w rolę postaci. Stosuje się ją najczęściej w produkcjach, w których protagonista jest człowiekiem lub postacią humanoidalną, szczególnie

w grach akcji. Perspektywa FPP pozwala twórcom zredukować koszt animacji głównej postaci – nawet w skrajnym wypadku gracz widzi jedynie część rąk i nóg bohatera.

- **PERSPEKTYWA TRZECIOOSOBOWA** – widok, w którym gracz widzi w całości postać bohatera. W wielu przypadkach kamera umieszczona jest za plecami protagonisty, choć nie jest to reguła. W tej perspekty-



Rys. 5. Perspektywa kamery – TPP; Assassin's Creed IV: Black Flag

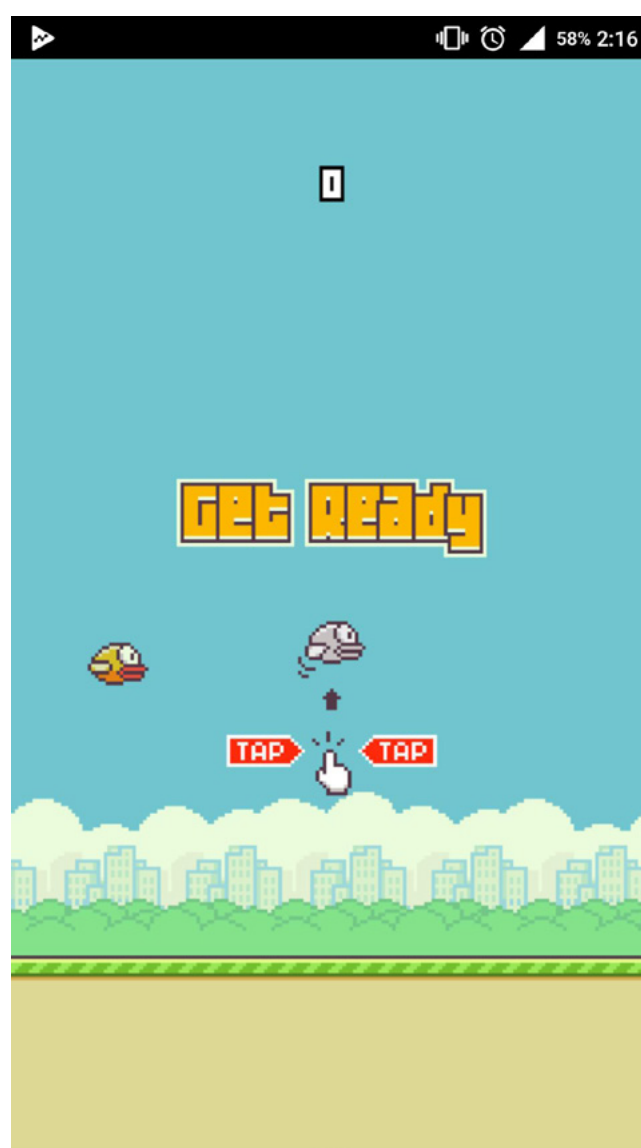


Rys. 6. Perspektywa kamery – izometria; League of Legends

wie gracz ma większe pole widzenia, niż w przypadku widoku FPP. W przypadku gier, w których nacisk położony został na precyzyjne pokonywanie przeszkód terenowych, na przykład gier platformowych, TPP może być dużym ułatwieniem dla gracza.

- **RZUT IZOMETRYCZNY** – widok popularny w grach, które w ręce gracza oddają nie jednego bohatera, a grupę postaci lub całe oddziały. Perspektywa ta jest często używana w grach fabularnych i strategicznych. Niektórzy twórcy decydują się na predefiniowany kąt kamery, podczas gdy inni umożliwiają jej obrót. Dawniej, gdy komputery dysponowały mniejszą mocą obliczeniową, rzut izometryczny pozwalał nadać dwuwymiarowym, często ręcznie rysowanym lokacjom wrażenie głębi.

- **SIDE-SCROLLER** – perspektywa ukazana w rzucie prostokątnym. Obraz jest przewijany w bok – najczęściej w prawo – tak, aby postać sterowana przez gracza pozostała w centrum ekranu.



Rys. 7. Perspektywa kamery – side-scroller; Flappy Bird



Rys. 8. Struktura świata – poziomy; Flippy Hills

Należy również wspomnieć o strukturze świata gry. Możemy tu wyróżnić następujące podejścia:

- **Poziomy** – rozgrywka jest podzielona na szereg etapów, z których każdy odbywa się w zamkniętym wycinku świata gry. Jest to jedna z najczęściej stosowanych metod, między innymi dlatego, że w znacznym stopniu ułatwia zoptymalizowanie produkcji pod urządzenia docelowe. Ponadto,

zaliczanie kolejnych poziomów wspiera odczuwanie przez gracza czynionych w grze postępów, co jest istotnym czynnikiem wpływającym na odczuwanie przyjemności z gry.

- **ENDLESS RUNNER** – styl stosowany najczęściej w grach platformowych. W tym przypadku świat gry jest jedną długą planszą, zaś zadaniem gracza jest pokonanie jak największego dystansu. Na drodze



Rys. 9. Struktura świata – endless runner; Sonic

gracza przeważnie znajdują się przeszkody, których pozycja może być predefiniowana lub też losowa.

• **OTWARTY ŚWIAT** – w tym ujęciu gracz otrzymuje do dyspozycji kompletną krainę, na ogół okazałych rozmiarów. Otwartość świata oznacza, że gracz może poruszać się po nim swobodnie, nie napotkawszy ekranów ładowania. Co więcej, w świecie takim nie powinny występować w nim sztuczne przeszkody nie do pokonania, z wyjątkiem zewnętrznych granic całkowitego obszaru gry oraz tymczasowych ograniczeń, które znikają wraz z postępami dokonywanymi przez gracza. W wielu grach tego typu świat jest podzielony na kilka zróżnicowanych obszarów, na przykład region górski, pustynię i zalesione niziny.

Niezaprzeczalną zaletą tego rozwiązania są olbrzymie możliwości kreacji wiarygodnego wirtualnego świata, stwarzającego

wrażenie tętniącego życiem, co jest szczególnie pożądane w grach fabularnych. Zaleta ta jest jednak również największą wadą otwartego świata: wiąże się z olbrzymimi kosztami produkcji oraz dużymi trudnościami w procesie optymalizacji gry.

Z tego powodu, niektórzy twórcy decydują się na stosowanie struktury łączącej cechy zamkniętych poziomów i otwartego świata: hubów¹⁰. Oznacza to, że przestrzeń gry jest złożona z kilku dużych obszarów, po których gracz przemieszcza się swobodnie, lecz przejście pomiędzy nimi możliwe jest jedynie w wyznaczonych miejscach i zazwyczaj towarzyszy mu ekran ładowania. Każdy taki obszar – hub – posiada na ogół silne cechy charakterystyczne, które wyróżniają go od pozostałych, często w sposób skrajny. Przykładowo, jeden z hubów może być jedynym w grze miastem, inny zaś dżunglą.

10. Ang. Hub – ośrodek.



Rys. 1. Styl graficzny – realizm; Forza Horizon 4

Proces powstawania gier

W tej sekcji przybliżony zostanie proces tworzenia gier od pustej kartki, poprzez utworzenie projektu aż do momentu ukończenia wymarzonej gry. Jest to jedna z możliwych ścieżek ku stworzeniu gry – wiele zależy od ludzi w zespole, ich temperamentu i umiejętności komunikacji, oraz współpracy.

Każdy dobry manager wie, że proces wytwórczy zależy od zaangażowania i umiejętności członków zespołu, musi więc uwzględniać materię ludzką. Jedni dobrze czują się w chaosie, inni muszą lubić pro-

jekt, który realizują, jeszcze inni są perfekcjonistami i jeśli powierzymy im stworzenie prototypu, możemy nie doczekać czasów prezentacji.

Dobłą praktyką jest poznanie zespołu (w miarę możliwości) jeszcze przed rozpoczęciem produkcji, aby uwzględnić umiejętności poszczególnych jego członków w planowaniu i podziale zadań, co pozwoli wykorzystać ich maksymalny potencjał. Dobrym momentem na obserwację ludzi jest wspólne poszukiwanie pomysłu na grę.

Pomysł na grę

“Pomysły są tanie” – to zdanie może potwierdzić każdy doświadczony twórca gier komputerowych. Z drugiej jednak strony, “dobry pomysł to połowa sukcesu” – przy czym oba zdania są prawdziwe. Często to moment powstawania pomysłu na grę jest początkiem sukcesu lub poważnych problemów. Sęk w tym, że o ile pomysły ma każdy, to niewiele pomysłów będzie na tyle dobrych, by stać się motorem napędowym gry.

Zdarzają się rzecz jasna gry, które zdobyły sławę ze względu na nietuzinkowy pomysł. Idealnym przykładem jest Flappy bird, oryginalny projekt zrealizowany w kilka dni przez jedną osobę. Mimo to większości pomysłów – nawet najbardziej odkrywczych – nie udaje się przekuć w finalny produkt

w tak krótkim czasie. Nie oznacza to natomiast, że dany pomysł jest z tego powodu zły – jest to po prostu inna droga do celu.

Celem tym jest stworzenie gry, a tworzenie gier jest drogie. Nawet w przypadku produkcji non-profit, osoby zaangażowane w projekt muszą poświęcić mu ogromną ilość czasu. Z biznesowego punktu widzenia, utrzymanie średniej wielkości zespołu liczącego kilkanaście osób to koszt kilkudziesięciu tysięcy złotych miesięcznie. Rzecz jasna, oczekiwane przez twórców gier wynagrodzenie zależy od ich wiedzy i doświadczenia, a pracownicy doświadczeni to większa pewność, że uda się zrealizować projekt, uniknąć opóźnień i skapitalizować produkt, aby inwestycja się zwróciła. Do tego oczywiście należy dodać koszt

zakupu sprzętu oraz opłaty jednorazowe lub abonamentowe za licencje niezbędne-
go oprogramowania.

Realizacja średniej wielkości gry to czas od sześciu miesięcy do kilku lat. Gry AAA¹ często wymagają pracy ponad 100 osób przez czas dłuższy niż 5 lat – ich budżet można liczyć w milionach, a nawet setkach milionów złotych. Nic więc dziwnego, że firmy zajmujące się tym rzemiosłem poświęcają czas i pieniądze na poszukiwania odpowiednich pomysłów, aby uzyskać solidną podstawę nowych projektów, zanim jeszcze na dobre zaczną w nie inwestować.

Skąd biorą się pomysły

Z tym samym problemem mierzy się od lat cała branża kreatywna: pisarze, dramaturdzy, scenarzyści, malarze, i tak dalej. Wielu artystów zdaje się na los, innym wystarczy odpowiedni nastrój. Jedna osoba może bez przerwy wpadać na nowe pomysły pod wpływem chwili, inna zaś pielęgnować w sobie jedną ideę przez wiele lat.

Zdarza się, że pomysł przychodzi z zewnątrz, lecz zazwyczaj nie jest on pełen i staje się dodatkowym wymaganiem, wokół którego buduje się realny pomysł – na przykład pojawia się inwestor, który oferuje swe pieniądze pod warunkiem, że będzie to gra o zwierzęciu z jego logo.

Można czekać, aż ten jeden idealny pomysł pojawi się sam, lecz można się również nie doczekać. Dlatego też w branży gier procesy pomysłotwórcze są inicjowane celowo,

1. Gry o największych budżetach na produkcję oraz promocję.

aby zwiększyć twórczość członków zespołu.

Zasypcie mnie pomysłami

Na początku należy ustalić zasady – do kiedy i gdzie zgłaszamy pomysły, jakiej mogą być maksymalnej długości i w jakiej formie. Dzięki temu nie będziemy musieli poświęcać czasu na zbieranie pomysłów z różnych miejsc ani czytać opasłych opracowań, których treść można skrócić do “mój pomysł jest najlepszy”.

Przykładem mogą być karteczki samoprzylepne przyklejane na tablicy, ścianie czy drzwiach. Dodatkowe aktywności manualne wspomagają proces twórczy, ponadto sposób ten naturalnie ogranicza miejsce na pomysły, zmuszając do wybrania najlepszych.

W tym przypadku czas na zgłoszenie swojego pomysłu powinien trwać około 3 dni, tak aby każdy na spokojnie mógł się zastanowić i spisać pomysł – lub pomysły.

Burza mózgów

Z pewnością jest to proces szybszy i silniej wzmagający kreatywność, jednakże nie zawsze przynosi oczekiwane efekty. Co gorsza, niekontrolowany może przynieść wiele szkód. Zespół pozostawiony bez nadzoru może ciągnąć burzę mózgów godzinami, nie wypracowawszy w efekcie wspólnego stanowiska. Problemem może być również zbyt luźna atmosfera – jeśli uczestnicy zagalopują się w żartach, to braknie czasu na znalezienie pomysłu.



Można temu jednak zapobiec, jeśli czas burzy mózgów zostanie ograniczony do około 30 minut, a jej przebieg będzie moderowany w oparciu o kilka prostych zasad:

- **NIE KRYTYKUJEMY** – celem burzy mózgów jest stworzenie jak najbogatszej bazy pomysłów. Otwarta krytyka i negowanie pomysłów innego uczestnika może łatwo doprowadzić do jego wycofania, w efekcie czego jego udział w burzy mózgów zaniknie, a jego pomysły pozostaną tajemnicą.

- **NOTUJEMY POMYSŁY** – wszystkie, dzięki czemu zachowamy wszystkie idee wniesione w trakcie dyskusji. Nawet pomysły, które w pierwszej chwili wydają się być abstrakcyjne lub niezwiązane z tematem mogą być cennym wkładem. Ważne jest też, by zapisane pomysły były widoczne dla wszystkich uczestników. Nie tylko pozwala to uniknąć powtórzeń, ale też wspomaga kreatywność.

- **TRZYMAMY SIĘ TEMATU** – chcemy, aby członkowie zespołu poczuli się swobodnie, jednakże trzeba umieć postawić granicę, tak by nie zepsuć całego spotkania. Dlatego staramy się hamować żarty i rozmowy nie na temat.

- **KOMENTUJEMY** – każda konstruktywna opinia zasługuje na wysłuchanie. Pomysł na grę nie pojawi się sam z siebie, będzie natomiast efektem sugestii wszystkich uczestników.

- **PILNUJEMY CZASU** – burza mózgów po 30 minutach zaczyna być męcząca. Pojawiają się wtedy rozmowy nie na temat, zaś ich brak tylko jeszcze bardziej nuży. Jeśli już musimy przeciągnąć proces, to zrobimy 5 minut przerwy. Zyskamy dzięki temu szansę, że któryś z uczestników spojrzy na pomysły z innej perspektywy.

Kreowanie wizji

Po określeniu pomysłu przychodzi pora na pierwsze zadanie – game designerzy muszą wykreować wizję gry. Jest to nie-samowicie odpowiedzialne zadanie, ponieważ wizja stanowi zestaw wytycznych, do których odwołujemy się przez cały okres produkcji. Ten dokument stanowi o całym kierunku produkcji, o tym co powinno być w niej zawarte, o tym czego nie powinniśmy zawierać.

Często wizję gry porównuje się do Konstytucji. Jest to zestaw najważniejszych praw i reguł, jakich muszą przestrzegać twórcy gry. Co więcej, jego zmiana może zaistnieć tylko w wyjątkowych okolicznościach, za zgodą przeważającej większości, w sytuacji jednoznacznej i absolutnej pewności, że aktualna wizja już się nie sprawdza.

Co powinien zawierać ten dokument? Podstawowe informacje to gatunek gry, platforma docelowa, styl graficzny, trochę referencji², drobny zarys fabularny. Jednakże warto dodać od siebie coś więcej – wszystko, co pomoże w przyszłości rozwiązać problemy zespołu, pozwalając oszczędzić jakże ważny i najbardziej kosztowny czas właściwej produkcji. Poniżej znajduje się kilka przykładów co powinno znaleźć się w dobrej wizji gry.

Baza odbiorców

Jeżeli zabieramy się za produkcję, to warto

2. Odwołania do produktów, na których chcemy się wzorować.

wiedzieć dla kogo tworzy się grę. Oczywiście są różne gry, niektóre bardziej uniwersalne, a niektóre jasno skoncentrowane na docelowej grupie odbiorców. Choćby ogólne określenie głównego odbiorcy pozwala ustalić kierunek produkcji – w końcu nikt nie wstawi nagości czy przemocy do gier dla dzieci.

Przeanalizujmy bardzo proste zdanie: “gra przeznaczona dla osób w wieku 18 – 25 lat, grających na telefonie podczas podróży komunikacją miejską do pracy lub na studia”. Po pierwsze już znamy platformę docelową – urządzenie mobilne. Wiemy, że są to osoby dorosłe, więc dozwolona jest nagość i przemoc – jednakże produkcja nie powinna nimi ociekać, bo jednak ma być to gra uruchamiana w miejscu publicznym, raczej dość zatłoczonym.

Przed pracą lub ciężkim dniem na uczelni odbiorca raczej chce się zrelaksować – gra nie powinna być zbyt angażująca czy wymagać wyjątkowych umiejętności. Codzienną podróż komunikacją miejską można oszacować na około 15 minut – rozgrywka właściwa powinna trwać około 10-12 minut. Do tego można dodać czas na dodatkowe “obowiązk” – aktywności, które można przerwać w każdym momencie bez ponoszenia konsekwencji, na przykład handel czy rozwój postaci.

A gdyby wiedzieć więcej? Gdyby wiedzieć, co nasz docelowy gracz lubi, w jakim społeczeństwie przebywa, jakie ma pasje

i zainteresowania, w jakie gry już grał, jaki wykonuje zawód? Nic prostszego – wystarczy go stworzyć.

Tworzenie persony

Zadanie to polega na wyobrażeniu sobie konkretnej osoby, która ma największe szanse na zagranie i polubienie naszej gry.

Nadajmy jej imię, określmy konkretny wiek – jeśli jest to dla nas istotne, to nawet znak zodiaku! Wklejmy zdjęcie, które najbardziej pasuje do naszej projekcji, napiszmy charakterystykę, opiszmy wycinek z życia takiej osoby – na przykład tydzień (jedni grają codziennie, inni tylko w weekendy, inni jeszcze inaczej).

Dzięki temu zawsze, gdy pojawią się wątpliwości, można zamknąć oczy i zobaczyć oczyma wyobraźni, czy Ania cieszyłaby się z gry, gdybyśmy dodali do niej element, o którym właśnie dyskutujemy.

Statystyki

Cała wcześniejsza analiza bazuje na prywatnych doświadczeniach, niekoniecznie prawdziwych. Może się okazać, że przeciętny czas podróży do pracy to 20 a nie 15 minut – niby niewiele, a jednak ponad 30% czasu więcej w naszej grze. Być może nadmierna przemoc jednak wcale nie przeszkadza w komunikacji miejskiej? Tylko, skąd niby mamy wiedzieć takie rzeczy, udać się do wróżki?

Na szczęście nie będzie to potrzebne. Na świecie prowadzony jest ogrom statystyk, a do dużej części jest darmowy dostęp – wy-

starczy wpisać odpowiednie hasła w wyszukiwarce. Nie chodzi tu tylko o statystyki prowadzone przez inne firmy tworzące gry komputerowe, każda gra się od siebie różni, więc takie statystyki mają porównywalną wartość do wszystkich innych.

Wracając do naszego przykładu z podróżami – okazuje się, że ktoś prowadził badania pod tytułem “Dojazdy Polaków do pracy”, dzięki którym możemy dowiedzieć się jaki % ludzi w Polsce podróżuje do pracy komunikacją miejską, ale również otrzymać kilka informacji wiążących komunikację miejską i stres. Jeżeli potrafiliśmy wywnioskować tak wiele na bazie własnych doświadczeń, to co można by osiągnąć, posiadając realne statystyki na wszelkie potrzebne nam tematy?

Zapewne odpowiednie badania już czekają, aby po nie sięgnąć. Trzeba tylko umieć postawić właściwe pytanie.

Monetyzacja

Gdy poznamy już na wskroś naszego użytkownika, przychodzi pora na kolejne pytanie, czyli: “jak zarobić?”. Rozwiązań jest wiele i z czasem zapewne przybędzie jeszcze więcej. Podstawowym podziałem są tak zwane gry darmowe (Free to Play) oraz gry płatne.

Gry darmowe mogą zarabiać na 3 sposoby: mikropłatności, reklamy i płatne DLC. Zależnie od platformy, każde z nich są mniej lub bardziej popularne – wykorzystuje się też często mieszankę kilku dopływów gotówki.



Jeżeli chodzi o gry płatne, to zazwyczaj ich głównym źródłem zarobku jest sprzedaż dostępu do gry. Dostęp ten może być opłacany dwójako: jako jednorazowa płatność lub w formie abonamentu płatnego cyklicznie (spotyka się również mix – czyli jednorazowy zakup gry, a później kontynuacja abonamentu np. w mniejszej stawce). Jednakże pomimo płatnego dostępu, sposoby monetyzacji z gier darmowych nadal są dostępne i coraz częściej stosowane.

Jaką obrać taktykę dla swojej gry? Nie ma jednej odpowiedzi lub prostego schematu, który odpowie na to pytanie. Podstawowymi wyznacznikami są typ gry oraz dogłębna analiza grona odbiorców.

Elevator Pitch i unikalne cechy gry

Lista USP to podstawowe i najlepsze narzędzie marketingowe w branży gier. Zazwyczaj odbiorcy są leniwi i nie lubią czytać, długie filmy ich nudzą a opisy wręcz odpychają. Zwięzła lista punktów obrazujących, dlaczego gra jest fajna, czym się wyróżnia

i za co można ją polubić, idealnie wpisuje się w potrzeby współczesnego świata, gdzie chcąc poznać jak najwięcej gier (bo wszystkich już nie sposób) mamy zaledwie kilka minut na zdecydowanie, czy warto poświęcić czas i pieniądze na ten właśnie tytuł.

Pozytywne przejście pierwszego etapu nie oznacza jeszcze zwycięstwa. Jeśli gracza zainteresują nasze USP, przychodzi kolej na etap właściwy. Jest nim Elevator Pitch – coś na wzór krótkiej, 30-sekundowej autoprezentacji, która ma pozwolić odbiorcy dowiedzieć się, o czym jest gra i go zaintrygować.

Aby lepiej zobrazować sobie powagę sytuacji, wyobraźmy sobie, że lądujemy w windzie sam na sam z milionerem. Mamy jedyną w życiu szansę, aby przekonać go do zainwestowania w nasz produkt – tym właśnie jest Elevator Pitch. To narzędzie jest wykorzystywane w promocji gier pośród użytkowników oraz jako prezentacja dla inwestorów.

Zbieranie feedbacku

Wielu młodych twórców ukrywa swoją wizję gry przed światem – tak, aby przypadkiem nikt nie ukradł tego świetnego rozwiązania albo, co gorsza, nie skrytykował go. Jednak działanie takie ma negatywny wpływ na rozwój game designera.

Mając ułożony i usystematyzowany pogląd na kształt gry, powinniśmy go prezentować jak najszerszej publiczności – nie tylko tej jej części, do której chcielibyśmy dotrzeć. Feedback³ zbiera się przez całe życie projektu, jednakże ten na początku projektu jest

3. Ang.; informacje zwrotne.

najcenniejszy. Zbieranie opinii już na etapie kreowania kształtu rozgrywki pozwala zmienić kierunek, w którym zmierzamy, bez ponoszenia kosztów związanych z produkcją lub daje nam czas na znalezienie rozwiązania, zanim problem realnie zaistnieje.

Nikt nie lubi krytyki, szczególnie jeśli jest emocjonalnie przywiązany do projektu, który chce realizować, jednakże unikanie feedbacku jest jak unikanie lekarzy, bo mogą zdiagnozować chorobę. Owszem, być może jest to nieprzyjemne, ale najprawdopodobniej uratuje życie – Twoje lub projektu.



Projektowanie rozgrywki

Gdy określiliśmy kierunek, w którym chcemy iść, a feedback i wizja ciężkiej pracy nie są nam straszne, przychodzi pora na opracowanie szczegółów. Podstawowym zadaniem designera jest przekuwanie pomysłów w projekt – niezależnie skąd te pomysły pochodzą. Wiele mówi się o GDD⁴, jest sporo źródeł na temat jak stworzyć dobrą księgę designu. Niektórzy kłócą się o ilość stron, inni o formę czy poziom szczegółowości. Jednakże należy pamiętać, że naszym celem jest stworzenie jak najlepszej gry – nie jak najlepszej dokumentacji, nie najczystszy kodu, customowego silnika czy innych rzeczy, które z czasem okazują się zbędne.

Dokumentacja projektowa powinna być taka, żeby odpowiadała całości zespołu, tak żeby każdy rozumiał projektanta, wiedział gdzie znaleźć to czego szuka, oraz jak to zrealizować. Pewne schematy i wzory wytwarzają się w trakcie, w bólu i łzach gdy po raz kolejny ktoś nas nie zrozumiał lub design pozwalał na zbyt dużą dowolność.

Ważne jest, żeby projekt danej funkcjonalności był spisany i dostępny, by móc razem z nim dokonać weryfikacji i na podstawie analizy, unikać błędów w przyszłości.

4. Ang. Game Design Doc – dokumentacja projektowa tworzonej gry.

Od czego zacząć

W życiu bywa tak, że czegoś bardzo pragniemy, a gdy zaczyna się to naprawdę dziać, to nie mamy pojęcia co zrobić. W designie jest podobnie. Odnajdujemy pomysł, budujemy wizję, zbieramy opinie i nagle przychodzi pora, żeby “projektować”. Kłębi się w głowie od pomysłów i różnorodnych rozwiązań, jednakże nic się ze sobą nie klei. Aby rozwiązać ten impas, warto spojrzeć na dostępne strategie przetwarzania informacji.

Top-down

Jest to strategia przeznaczona dla osób, które mają ogólną wizję całości. Wiedzą co chcą osiągnąć i jaki powinno mieć to kształt. W ten sposób znając ogół produktu, zaczyna się opracowywać potrzebne szczegóły.

Założmy dla przykładu, że dokładnie wiemy jak powinien wyglądać finał całej rozgrywki. Możemy wtedy od tego zacząć i iść w tył, opracowując mechaniki i elementy oraz etapy, w których gracz będzie się ich uczył. Pojawia się u gracza potrzeba, którą projektant stara się zaspokoić.

Bottom-up

Innym rozwiązaniem jest strategia oddolna. Jeśli jest napływ koncepcji na różne

mechaniki, ale brakuje pewności czym tak do końca ma być gra, to takie rozwiązanie powinno sprawdzić się najlepiej.

Podejście to stawia na jak najlepsze opracowanie ciekawych mechanik, zakładając, że jeżeli będą naprawdę dobre to z czasem złożą się one w pewną kompletną całość. Czyli jeśli wiemy, w jaki sposób postać główna ma walczyć, mamy pomysł na dwa wątki fabularne i strój postaci trzecioplanowej, to zaczynamy to realizować i poszukiwać w tym części wspólnej.

Core gameplay loop

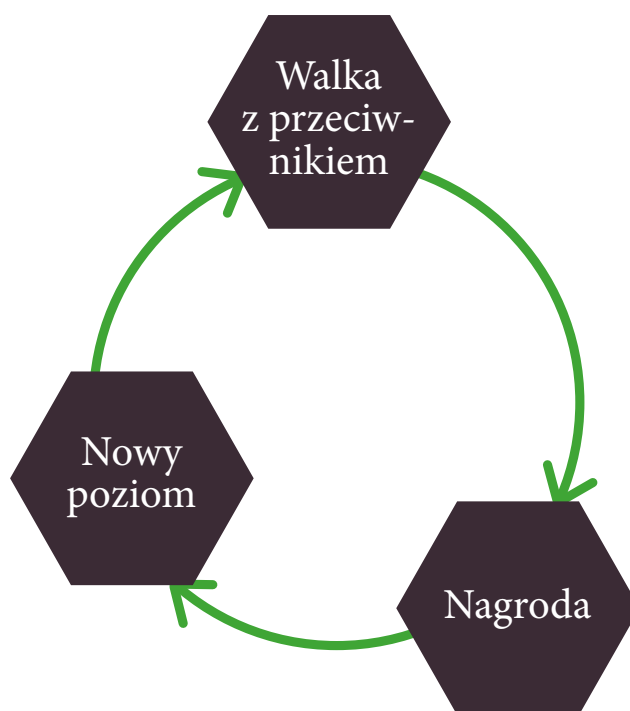
Elementem odróżniającym gry od innych mediów jest występowanie interakcji i podejmowanie decyzji przez użytkownika. Tak więc każda gra posiada akcje, które są wykonywane przez odbiorcę.

Zazwyczaj akcje te są powtarzalne – oczywiście zmieniają się warunki i otoczek, jednakże można je wsadzić do wspólnego worka. Jeżeli powtarzają się cyklicznie, możemy mówić o pętli rozgrywki. Jeżeli te aktywności możemy uznać za podstawowe – występują dość często, są ważnym elementem gry i stanowią bazę rozgrywki gracza – to możemy mówić o podstawowej pętli rozgrywki, czyli właśnie *core gameplay loop*⁵.

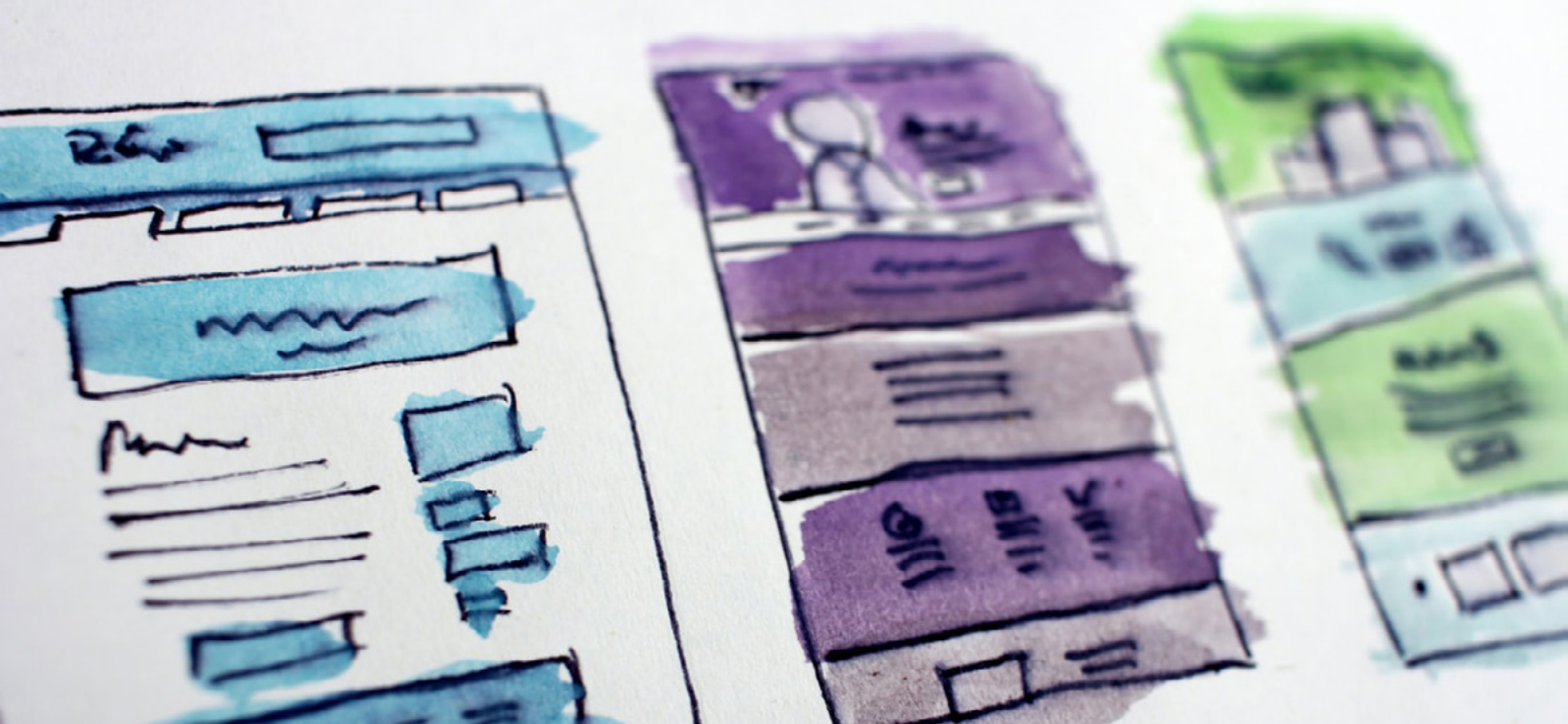
Aby lepiej to zobrazować, poniżej znajduje się przykładowy diagram takiej rozgrywki. Jest to przykład gry, która skupia się na walce z przeciwnikiem i rozwoju postaci. Wszelkie inne aktywności stanowią jedynie dodatek – występują pomiędzy lub obok

aktywności głównych, najprawdopodobniej tworząc kolejne pętle rozgrywki.

Odkrycie, jakie aktywności będą najczęściej wykonywane, pozwala uświadomić sobie co jest najważniejsze w naszym produkcie. Po utworzeniu prostego diagramu jak ten powyżej, należy w pełni skupić się na implementacji bazy gry, która będzie zawierać wymienione elementy. To one zdecydują, czy produkt, który tworzymy ma szansę na sukces. Jeśli nie sprawią frajdy, to czy dodanie kilku fajerwerków obok nagle odmieni produkt w arcydzieło?



5. Ang.; Podstawowa pętla rozgrywki.



Prototypowanie

Zgodnie z podejściem *Fail Faster*⁶, im szybciej poniesiemy porażkę, tym większe mamy szanse na wdrożenie poprawek i finalnie osiągnięcie sukcesu. Filozofia ta odnosi się do jak najszybszego i najtańszego prototypowania, tak aby potwierdzić czy obrana koncepcja jest słuszna. Ale tak naprawdę czym jest prototyp? Gdzie jest granica między prototypem a prostym produktem? Niestety, tu również wszystko jest kwestią poszukiwania złotego środka i jest zależne od percepcji zespołu.

Podstawową zasadą prototypu jest tworzenie go tanio i szybko, bez przywiązywania uwagi do szczegółów, będących jedynie dodatkiem produktu lub wybranego systemu. Dlaczego? Ponieważ często prototypy lądują w koszu, a im więcej takowemu

poświęcimy, tym większe koszty ponosi zespół. Na początek najlepiej wziąć na warsztat Core Gameplay Loop, dodać niezbędne elementy jak na przykład poruszanie postaci głównej.

Gdy to zadziała, możemy przejść do prototypowania kolejnych elementów oraz realizacji właściwej zatwierdzonych prototypów – zależnie od zespołu i sytuacji biznesowej projektu, realizuje się obie te rzeczy z różnym natężeniem. Przykładowo, fragment gry zbudowany na *core gameplay loopie*, w opracowanej wersji, która ma być zbliżona do finalnej (gotowe modele, dźwięki, mechaniki, teksty, UX⁷) nazywany jest *Vertical Slice*⁸.

6. Ang.; Ponoś porażki szybciej.

7. Ang. User Experience – doświadczenia gracza.

8. Ang.; Pionowy wycinek, coś na wzór wyciętego kawałka tortu.

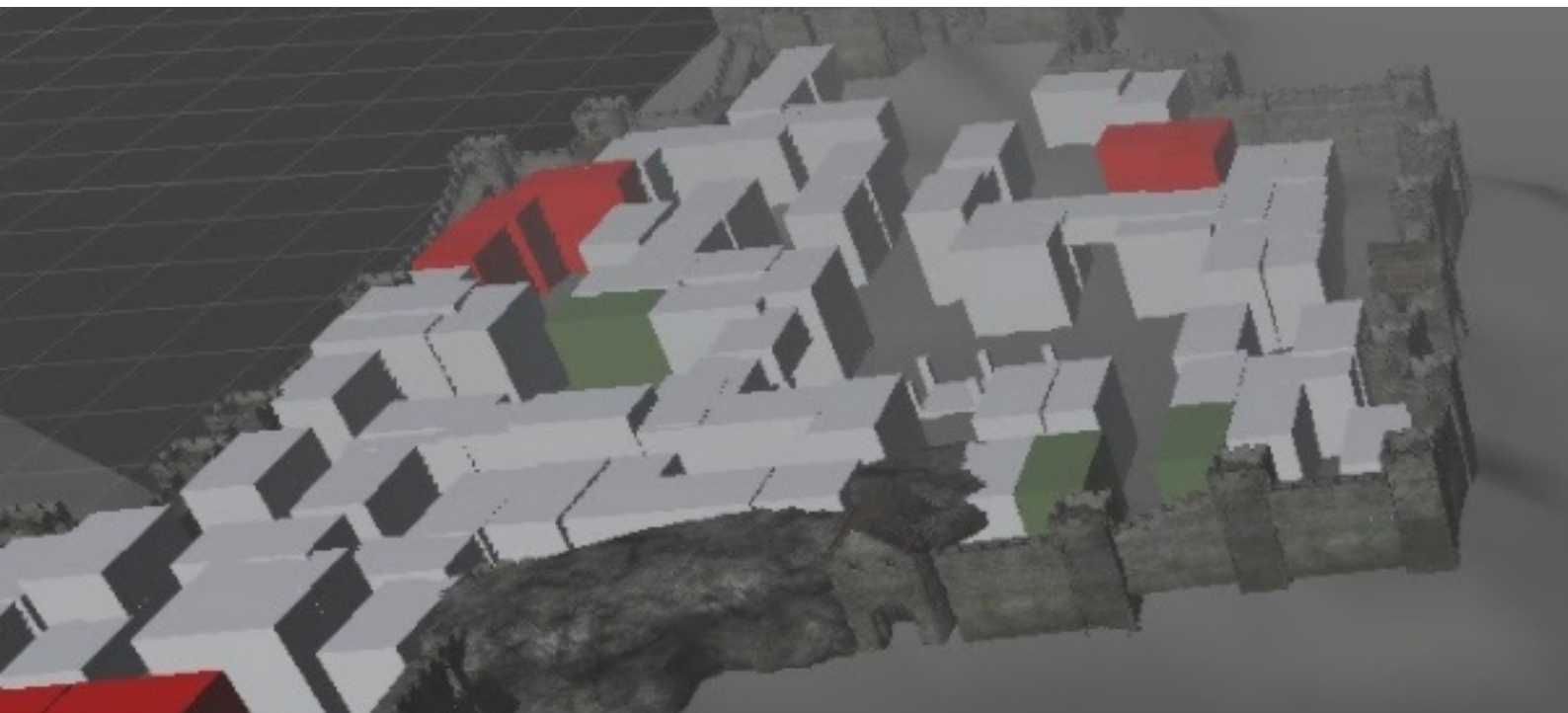
Whiteboxing

Jest to przykładowa metoda prototypowania stosowana w trakcie projektowania poziomów. *Whiteboxing* polega na zbudowaniu uproszczonego poziomu. W tym celu prototypowany poziom jest na początku tworzony nie z gotowych modeli, a z podstawowych brył geometrycznych: sześciątów i prostopadłościanów.

Metoda ta pozwala stworzyć w bardzo krótkim czasie grywalny prototyp poziomu, który można przetestować w praktyce, aby upewnić się, że jego budowa odpowiada wymaganiom stawianym przez zaprojektowaną rozgrywkę. Ponadto, korzystając z *whiteboxingu* twórcy mogą niskim kosztem

wprowadzać zmiany w oparciu o opinie uzyskane podczas wstępnych testów wewnętrznych.

Co najważniejsze, zastosowanie *whiteboxingu* pozwala wykrywać błędy jeszcze w trakcie fazy prototypowania, która często odbywa się jeszcze na etapie projektowania gry. Dzięki temu maleje ryzyko, że złecone zasoby, które zostaną użyte do stworzenia finalnej wersji poziomu – w szczególności modele 3D – zostaną stworzone w oparciu o jego błędny układ. To zaś mogłoby w konsekwencji skutkować koniecznością drogich poprawek lub, w skrajnych przypadkach, niewykorzystaniem w grze stworzonych za środki z budżetu gry modeli lub poziomu.



Rys. 12. Prototypowanie – whiteboxing

Roadmap

Gdy posiadamy wizję, prototyp podstaw gry i zapewne kilka projektów systemów, które powstały w trakcie prototypowania, przychodzi pora na usystematyzowanie pracy. *Roadmap*⁹ to rozłożenie projektu na części, zaplanowanie kolejnych kroków i umiejscowienie ich w czasie.

Na potrzeby tego zadania, kolejne elementy mogą być dość ogólne – z czasem można pod nie podpinać linki do kolejnych projektów. Dzięki takiej mapie, cały zespół a nawet gracze czy inwestorzy znają plan działania. Wiedzą czego i kiedy mogą się spodziewać. Jednakże, aby stworzyć *roadmapę*, musimy dobrze znać produkt i siły zespołu.

Priorytetyzacja i estymacja

Aby móc ustalić kolejność realizacji kolejnych elementów, musimy ustalić priorytety. Bez tego nie będziemy wiedzieć, które zadanie jest ważniejsze. Jedną z technik przypisywania wagi elementom jest MoSCoW – czyli co musi (ang. *must*), powinno (ang. *should*), mogłoby (ang. *could*), a co warto dodać ale raczej nie zostanie dodane do gry (ang. *would*).

Zaletą takiego podejścia jest szybkość działania – określamy co jest najważniejsze a co mniej ważne, bez porównywania bezpośrednio każdego z elementów. W teorii mając ustalone priorytety, wiemy od czego zacząć, jednakże pojawia się też

9. Ang.; mapa drogowa.

inny aspekt – estymacja.

Estymowanie to szacunkowe ocenianie czasu potrzebnego na wykonanie danego zadania. Jest to wyzwanie szalenie trudne, zwłaszcza na początku, gdy nie mamy żadnego doświadczenia w kwestii estymacji ani zadań, które realizujemy. Nawet najlepsi znając zapędy do widzenia świata w różowych okularach, potrafią mnożyć swe estymacje dwukrotnie, a nawet trzykrotnie!

Dopiero posiadając ocenę potrzebnego czasu i priorytetyzację, jesteśmy w stanie zbudować plan – bo czasem warto odłożyć zadanie z najwyższym priorytetem, na rzecz tego z odrobinę niższym jeśli różnica w czasie realizacji jest znaczna.

Definicja słowa “zrobione”

Każdy plan, który powstaje, ma dwa zastosowania. Pierwsze to oczywiście uproszczenie i usystematyzowanie procesu realizacji planu. Drugie to kontrola postępów, czy aby na pewno wszystko idzie zgodnie z planem, czy zdążymy z projektem na czas, czy projekt zmierza w ustalonym kierunku.

Obu tym aspektom przyda się tak zwane *Definition of Done*¹⁰ – jako wyznacznik tego, co uznajemy za wykonane i zakończone zadanie. I choć wydaje się to marnowaniem czasu, to w rzeczywistości jest to bardzo cenne narzędzie. Bo tak naprawdę kiedy możemy mówić o zakończeniu zadania? Gdy używając sztuczek developerskich

10. Ang.; definicja słowa “zrobione”.

jesteśmy w stanie przetestować daną mechanikę? A może kiedy dana funkcjonalność jest w grze i działa bez stosowania takich furtek? Idąc dalej – coś jest zrobione, gdy jest dostępne w grze oraz gdy projektant tego elementu wraz z testerem potwierdzają, że działa on zgodnie z założeniami.

Jest jeszcze wiele rozwiązań pośrednich. Ukształtowanie takiej definicji, ma wpływ na całą realizację projektu – buduje proces implementacji, mówi kiedy projektant i tester weryfikują kolejne elementy, oraz kiedy można przejść do kolejnego zadania.



Realizacja projektu

Podczas realizacji projektu, każdy członek zespołu powinien ściśle trzymać się swojej profesji. Producent dogląda postępów, koryguje estymacje i na ich podstawie buduje szczegółowe plany na kilka/ kilkanaście dni do przodu.

Projektant tworzy nowe dokumenty na temat kolejnych funkcjonalności, jednocześnie współpracując z osobami realizującymi jego projekty. Graficy, programiści i cała reszta (od projektantów poziomów i oświetleń, przez dźwiękowców, animatorów, aż do pisarzy), realizują zaprojektowane elementy, według priorytetów i kolejności

przyjętej przez producenta. Tylko tyle i aż tyle.

Nie dość, że uzyskanie pomysłu, stworzenie wizji, roadmapy i wszystkie inne przygotowania pochłonęły kilka miesięcy pracy, to dopiero teraz nazywa się to prawdziwą realizacją. Co więcej, na zespół czyha wiele problemów. Producent może źle dobrać zadania, przez co okaże się, że ktoś ma zbyt wiele na karku lub nie ma co robić. Jeśli projektant pominie coś w dokumentacji albo zmieni zdanie w trakcie implementacji, będzie ona nieczytelna. W efekcie zespół nie do końca zrozumie dany projekt, lub

pozostawi on zbyt wiele miejsca na interpretację, gdy ta nie jest akurat pożądana.

Oprócz zwykłych błędów – bo w końcu wszyscy jesteśmy tylko ludźmi – pojawiają się też inne, których nie pominiemy, jak bardzo byśmy się nie przykładali. W trakcie realizacji może się okazać, że myliliśmy się co do założeń albo że nauczyliśmy się tak wiele, że elementy z początku projektu chcielibyśmy zrobić od nowa, by uzyskać lepszy efekt. Tu pojawia się ważne stwierdzenie – tworzenie gier to proces iteracyjny, często gdy coś stworzymy, później poprawiamy bądź zmieniamy. Tak jak ekipy filmowe kręcą duble, poprawiają dialogi i sceny, tak w grach poprawia się funkcjonalności, odświeża interfejs czy zmienia mechaniki.

Znacznie łatwiej jest ocenić dany element, mogąc w niego zagrać, zamiast tylko go sobie wyobrażać. Dlatego na początku często prototypujemy poszczególne elementy, np. przed złożeniem zamówienia na modele czy animacje warte wiele czasu i pieniędzy. Jednakże trzeba zachować umiar, tak żeby nie zamknąć się w nieskończonej pętli poprawiania projektu, bo nigdy go nie wydamy.

Metodyki prowadzenia projektu

Chcąc zminimalizować straty korzystamy z wcześniej wymienionych technik, jak chociażby roadmapa, która wymusza na zespole wprowadzanie kolejnych elementów wraz z upływem czasu. Jednakże to metodyki zarządzania tak naprawdę służą

do utrzymania porządku i doprowadzenia projektu do celu. Istnieje wiele wariacji i gotowych rozwiązań, możemy z nich korzystać lub szukać rozwiązania pośredniego.

MODEL KASKADOWY¹¹ – opracowany około 1970 roku do zarządzania tworzeniem dużych systemów informatycznych. Polega on na podzieleniu projektu na konkretne fazy i wykonywanie ich po sobie. Kończąc jedną kaskadę, przechodzimy do kolejnej. Dzięki temu docieramy do kolejnych etapów, będąc zmuszonym decydować czy jest coś zakończone czy nie. Jeżeli, w którymś momencie uznamy, że produkt jest niesatysfakcjonujący, cofamy się o iteracje (kaskady) i poprawiamy projekt, aż nie osiągniemy oczekiwanego celu – dopiero wtedy możemy pójść dalej.

ZWINNE WYTWARZANIE OPROGRAMOWANIA – znane pod skrótową nazwą *Agile* (ang. “zwinny”). Manifest *Agile* to zbiór wspólnych zasad dla zwinnych metod wytwarzania oprogramowania. Jest bardzo prosty i zawiera jedynie cztery punkty:

- Ludzie i interakcje ponad procesy i narzędzia
- Działające oprogramowanie ponad obszerną dokumentację
- Współpracę z klientem ponad formalne ustalenia
- Reagowanie na zmiany ponad podążanie za planem

Do zwinnych metod zaliczamy bardziej zaawansowane i opisane, takie jak Lean Software Development, Programowanie

11. Ang. waterfall – wodospad.



Ekstremalne, znany każdej firmie Scrum, czy też Kanban. Ta ostatnia jest bardzo intrygująca. David J. Anderson w 2010 roku opisał metodę bazującą na rozwiązaniu, które uratowało Toyotę, a być może całą Japonię w latach 50-tych.

Każda z wymienionych metod bazuje na obserwowaniu postępów i dostosowywaniu się do zmian.

Testy

Bardzo ważnym elementem tworzenia gier są testy. Jako, że gry zazwyczaj bazują na decyzjach i dowolności gracza, nie obejdziesz się bez testów manualnych. Prosto ujmując, ktoś musi uruchomić grę i w nią zagrać. I choć brzmi to jak wymarzona praca, to nie jest to takie przyjemne jak mogłoby się to wydawać.

Grając, sprawdzamy konkretne elementy – często będąc zmuszonym do oglądania konkretnego elementu przez kilka lub kilkanaście dni, żeby sprawdzić każdą akcję możliwą do wykonania przez gracza. Zamiast rozkoszować się rozgrywką, tester sprawdza każdy jeden kąt i każdy jeden piksel, czy nie powstało coś niezamierzonego, co może zepsuć rozgrywkę gracza.

Aby testy były efektywne, stosuje się listy kontrolne czy też *User Stories* – krótkie scenariusze opisujące akcje użytkownika. Dzięki temu, można obserwować co zostało przetestowane, co jeszcze należy przetestować i ewentualnie dopisać nowe przypadki do przetestowania.

Gdy implementacja idzie dobrze i zbliża się ku końcowi, czujemy, że już wiemy wszystko na temat finalnego kształtu projektu, nareszcie przychodzi moment na najprzyjemniejszą część implementacji, czyli *polish*¹². Zaczynamy przejmować się najmniejszymi drobnostkami, dodajemy dźwięki efekty, finalne grafiki.

Nasz projekt zaczyna wyglądać jak pełnoprawna gra. Zamiast dyskutować o głównych mechanikach czy w ogóle powinny się znaleźć w projekcie, dyskutujemy o kolorach, pozycjach przycisków czy włosach postaci. Faza ta skupia się na dopięciu detali projektu na ostatni guzik, tak by móc z radością na ustach pokazać stworzoną grę całemu światu.

Balans rozgrywki

Poziom trudności gry ma ogromny wpływ na finalny odbiór użytkownika – jeśli gra będzie zbyt prosta, gracz bardzo szybko się znudzi. Z drugiej strony, jeśli będzie zbyt trudna, może doprowadzić do frustracji. Patrząc na inne aspekty – jeśli gra będzie prezentować wiele dostępnych rozwiązań, ale tylko jedno pozwoli dążyć do zwycięstwa, to gracz poczuje się oszukany.

Oto kilka rozwiązań, które mogą przyspieszyć prace nad balansem rozgrywki:

- **DZIEL PRZEZ 2, POMNÓŻ PRZEZ 2** – podczas balansowania warto wprowadzać duże zmiany. Przykład: Jeśli postać chodzi zbyt wolno, przyspieszmy ją dwukrotnie,

jeśli wtedy jest za szybka to ustawmy prędkość po środku między poprzednim a aktualnym stanem. Dzięki tej technice bardzo szybko znajdziemy odpowiednią wartość.

- **WEŹ POD UWAGĘ RÓŻNE STANY** (min, max, średni) – produkt trafi do różnych osób o różnych umiejętnościach, wiedzy i doświadczeniach, podczas balansowania należy wziąć pod uwagę różne możliwości. Wyliczenie ekonomii, gdzie nie można popełnić ani jednego błędu, zazwyczaj kończy się fiaskiem i frustracją

- **STWÓRZ KALKULATOR** – proste i szybkie w obsłudze narzędzie, które pozwoli przed wprowadzeniem zmian w projekcie obliczyć różne dostępne opcje i wybrać najlepszą. Kalkulator nie musi być ładny, ale powinien być czytelny

- **KONTRY** – jedna z wielu technik upraszczających balansowanie. Polega na budowaniu składowych walki czy konkurencyjności w ten sposób, że każdy element kontruje i jest kontrowany przez inne. Dzięki takiemu podejściu dość łatwo przypiszemy nowy element do jednej z grup. Nie ma tu również jednej ścieżki, która będzie lepsza od innych

- **EVERYTHING OP¹³** – kolejna technika, zakładająca zalety każdego z elementów. Gracz patrząc na nie dostaje ślinotoku i nie wie co wybrać. Każdy może znaleźć coś dla siebie i po wyspecjalizowaniu czuje się wyjątkowy i potężny.

12. Ang.; polerowanie.

13. ang. everything overpowered – wszystko zbyt potężne.



Doświadczenia gracza

Gdy dopinamy wszystko na ostatni guzik, liczy się każdy szczegół. Chcemy, żeby nasz gracz czuł się wyjątkowo, miał dobre wspomnienia i nic nie zepsuło jego rozgrywki. Warto poświęcić czas na cenne artykuły, na przykład *8 golden rules of interface design*¹⁴, czy o psychologii kolorów. Tak, kolory mają ogromne znaczenie i mogą zmienić losy całej gry. Na przykład kolor czerwony wzmacnia agresywność i porywczność gracza, kolor biały jest kolorem śmierci w kulturach Wschodu, a pomarańczowy to kolor szczęścia i idealnie pasuje do nagród czy znajdziek.

Jednak nawet najlepiej dobrana oprawa nie zawsze dotrze do użytkownika. Aby zwiększyć szanse na dotarcie do gracza, często wytacza się najcięższe działa.

Tutorial

Pomimo największych starań o czytelność i intuicyjność, nie wszystko będzie od razu oczywiste dla gracza. *Tutorial*¹⁵ służy do przeprowadzenia gracza przez najbardziej skomplikowane elementy rozgrywki. Spotykany jest zazwyczaj na początku gry, tak aby zapoznać się z potrzebnymi mechanikami a później mógł zagłębić się w przedsta-

wiony świat.

Występuje w różnorodnych formach, czasem to po prostu tekst i strzałka, czasem nauczyciel NPC¹⁶, na przykład w formie trenera na arenie, a czasem tak ukryty w zadaniach, że tylko najbardziej doświadczeni gracze orientują się, że to tylko szkolenie.

Optymalizacja

Bardzo ciężko jest docenić optymalizację, bo gra "po prostu działa". Problem pojawia się, gdy sprzęt nie radzi sobie z daną produkcją i albo w ogóle jej nie uruchamia lub zmusza użytkownika do katongi na minimalnych ustawieniach bez płynności rozgrywki.

Stosowane techniki to na przykład upraszczanie modeli czy stosowanie do nich systemów LOD¹⁷, które podmieniają wersje modelu na mniej lub bardziej skomplikowane zależnie od odległości kamery. Poprawia się obliczenia, zazwyczaj te w najważniejszych punktach jak poruszanie postaci, czy skomplikowane efekty wizualne. Wszystko po to, by dotrzeć do większej liczby użytkowników. Prosta gra przeglądarkowa z wymogiem sprzętu najwyższej klasy może przynieść sławę, ale raczej nie takiej, jakiej się oczekuje.

14. Ang.; 8 złotych zasad projektowania interfejsów.

15. Ang.: instruktaż

16. Ang. Non-playable character – bohater niezależny.

17. Ang. Level of detail

Narzędzia wykorzystywane przy tworzeniu gier

Projektowanie

D **DIAGRAMY** – np. aplikacja draw.io, która umożliwia tworzenie zarówno prostych diagramów, jak i gotowych prototypów interfejsu przy użyciu dostępnych szablonów, umożliwia też pracę nad danym plikiem kilku osobom naraz, z udostępnianiem zmian każdego użytkownika w czasie rzeczywistym.

PAKIETY BIUROWE – np. Google Docs, które umożliwia pracę nad danym plikiem kilku osobom naraz, z udostępnianiem zmian każdego użytkownika w czasie rzeczywistym.

Programowanie

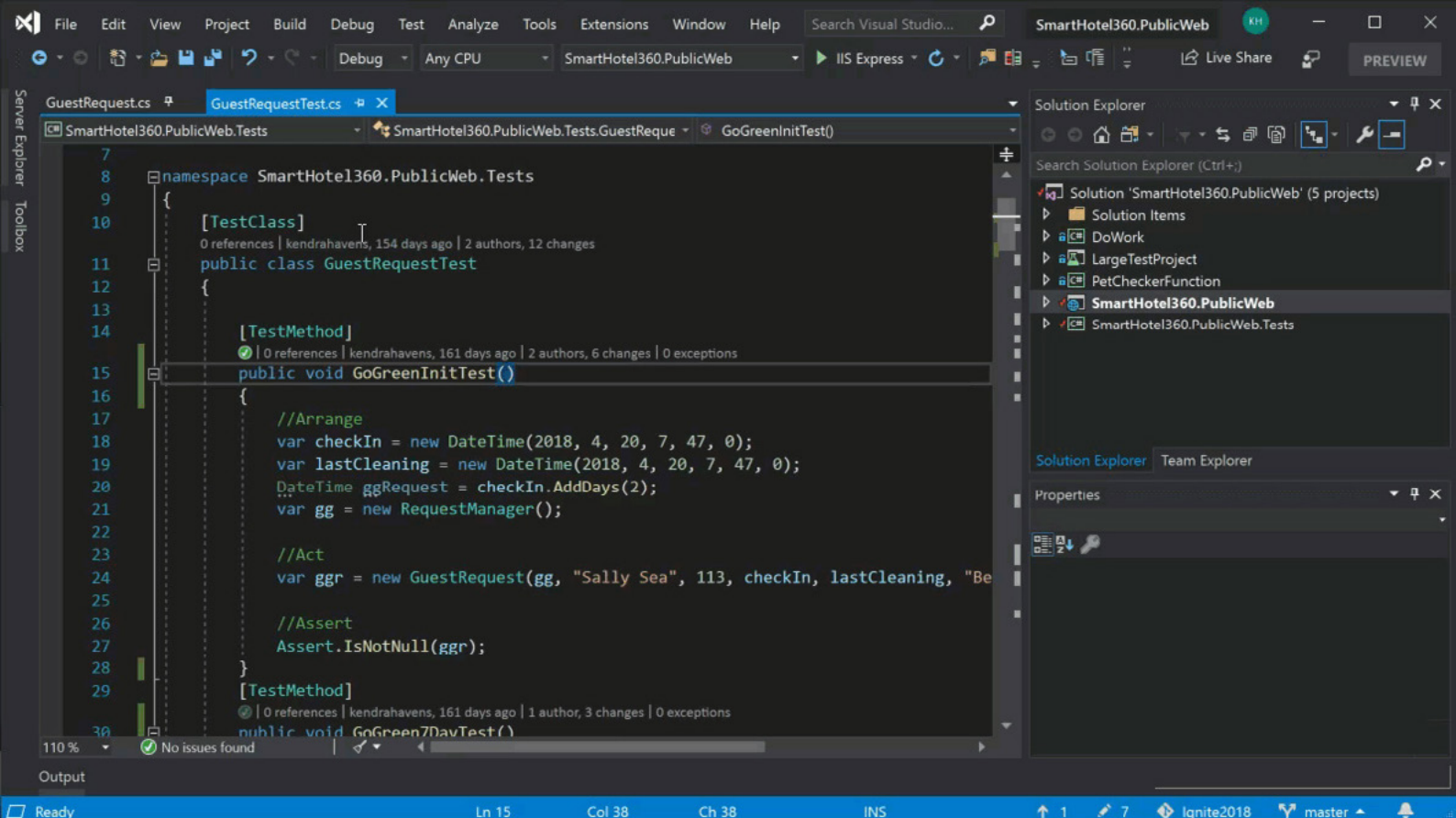
Microsoft Visual Studio

Microsoft Visual Studio od firmy Microsoft to zintegrowane środowisko programistyczne, czyli program (lub zespół programów) przeznaczony do tworzenia oprogramowania w formie kodu, czyli zbioru instrukcji wykonywanych przez komputer.

Visual Studio pozwala tworzyć oprogramowanie dla wielu platform docelowych, wliczając w to systemy operacyjne Windows i Windows Phone czy konsole do gier XBOX. Zintegrowane środowisko programistyczne, takie jak Visual Studio, pozwala stworzyć w całości działające oprogramowanie oraz prostsze gry, które nie wymagają interfejsu graficznego.

Visual Studio umożliwia programowanie w wielu językach programowania, w tym C++ oraz C#, które są wykorzystywane odpowiednio przez Unreal Engine oraz Unity – jedne z najpopularniejszych spośród dostępnych komercyjnie silników do tworzenia gier komputerowych. Ponadto, twórcy obu wymienionych silników zapewniają pełną integrację z Visual Studio.

Z tych powodów, Visual Studio to jedno z najczęściej używanych narzędzi dla programistów zaangażowanych w produkcję gier komputerowych.



Rys. 13. Narzędzia – programowanie; Microsoft Visual Studio

Najważniejsze elementy składowe środowiska Visual Studio:

Edytor kodu

Jest to podstawowe narzędzie wykorzystywane przez programistę. Edytor kodu pozwala programiście wprowadzanie fragmentów kodu, czyli instrukcji dla urządzenia, które będzie odtwarzało stworzone w ten sposób oprogramowanie.

Edytor kodu dostępny w Visual Studio posiada wiele dodatkowych funkcjonalności, które znacznie usprawniają pracę programisty. Należą do nich między innymi:

PODŚWIETLANIE SKŁADNI – polega na wyróżnianiu w edytorze kodu, za pomocą koloru oraz atrybutów, elementów składniowych kodu źródłowego tworzonego oprogramowania, takich jak np. komentarze czy

słowa kluczowe. Jest to rozwiązanie, które w znacznym stopniu zwiększa czytelność kodu, a w konsekwencji komfort pracy programisty.

KOMPILACJA W TLE – jest to narzędzie, które na bieżąco kompiluje tworzony kod w celu informowania programisty o popełnionych błędach składniowych oraz błędach kompilacji, które mogłyby całkowicie uniemożliwić odtworzenie programu lub spowodować jego niewłaściwe działanie.

Debugger

*Debugger*¹ to zawarty w IDE program, którego zadaniem jest analiza tworzonego oprogramowania w czasie rzeczywistym, w celu wykrycia i określenia występujących w nich błędów.

1. Nazwa pochodzi od potocznej nazwy dla błędu w kodzie (ang. bug – robak)

Tworzenie grafiki

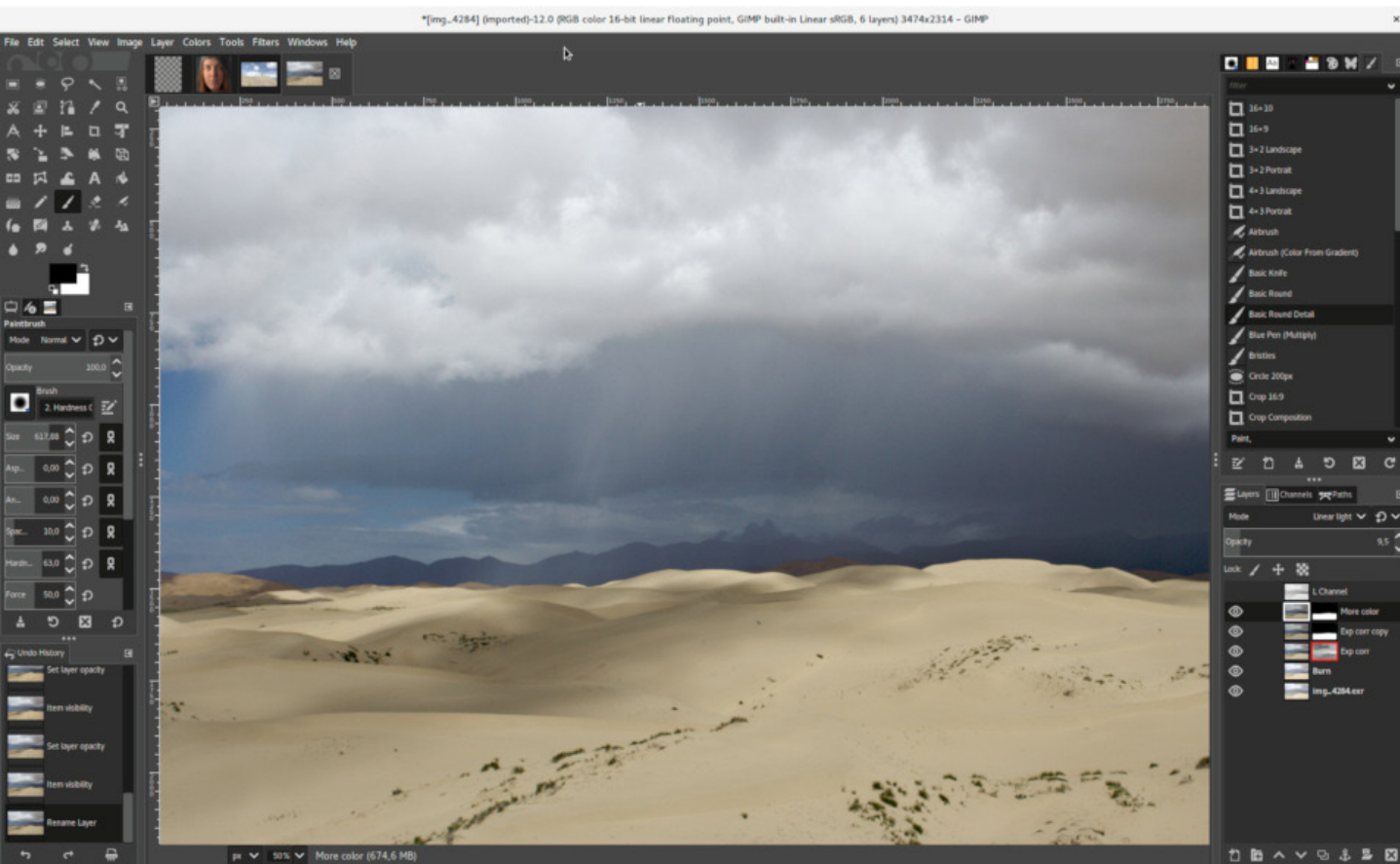
Grafika 2D – GIMP

Najbardziej podstawowym narzędziem służącym do tworzenia grafiki, z której korzysta obecnie praktycznie każda gra, jest edytor grafiki rastrowej. Edytor taki umożliwia tworzenie od zera dwuwymiarowych grafik oraz dokonywanie zaawansowanych przekształceń obrazu.

Edytory grafiki rastrowej mają wiele zastosowań w procesie produkcji gier. Za ich pomocą tworzone są elementy interfejsu, za pomocą którego gracz jest informowany o zachodzących w świecie gry zmianach

oraz który pozwala graczowi na dokonywanie interakcji. Edytory grafiki rastrowej pozwalają też tworzyć tekstury, które muszą zostać nałożone na modele 3D. Z kolei w przypadku produkcji rozgrywających się w dwóch wymiarach, można za ich pomocą stworzyć cały świat gry.

Najsilniejszą pozycję spośród używanych w branży gier edytorów grafiki rastrowej jest Adobe Photoshop. Jest to jednak rozwiązanie komercyjne, dlatego wiele mniejszych zespołów oraz początkujących grafików decyduje się na konkurencyjny darmowy odpowiednik: GIMP.



Rys. 14. Narzędzia – tworzenie grafiki; GIMP

Najważniejsze cechy powszechnie używanych zaawansowanych edytorów grafiki rastrowej, takich jak GIMP:

- **STRUKTURA WARSTW** – umożliwia tworzenie finalnej grafiki wykorzystując wiele warstw obrazu. Poszczególne warstwy są wyświetlane według ustalonej przez użytkownika kolejności: najpierw wyświetlana jest najwyższa warstwa, a pod nią kolejne. Pozwala to na mieszanie ze sobą poszczególnych warstw w celu uzyskania pożądanego efektu, a także dokonywanie poszczególnych przekształceń jedynie na wybranych warstwach.
- **ZAZNACZENIE** – funkcja ta pozwala użytkownikowi na określenie obszaru roboczego na powierzchni obrazu, w którym wprowadzane przez niego w danym momencie zmiany mają odnieść skutek. Oprócz zaznaczenia w formie figur geometrycznych, zaawansowane edytory pozwalają na zaznaczenie odręczne.
- **RYŚOWANIE** – zbiór funkcji pozwalających użytkownikowi na tworzenie nowej grafiki od zera. Może być to rysowanie z wykorzystaniem linii prostych lub krzywych, albo rysowanie odręczne. Ważne są narzędzia pozwalające ustalić rozmiar, kształt i siłę pędzla, szczególnie podczas korzystania z tabletu graficznego.
- **PRZEKSZTAŁCENIA** – narzędzia pozwalające na przeróżne modyfikacje całego obrazu, poszczególnych warstw lub zaznaczonego obszaru. Mogą być to proste operacje, jak zwykle skalowanie obrazu, lub bardziej zaawansowane, jak modyfikacja balansu

kolorów czy ich odwrócenie, wygięcie obrazu względem danej płaszczyzny etc.

- **KOREKTY** – funkcje, których zadaniem jest automatyczne usunięcie jak największej liczby niedoskonałości, lecz bez zniekształcenia obrazu bądź negatywnego wpływu na jego ostateczną jakość.

Grafika 3D – Blender

Drugim rodzajem grafiki, powszechnie stosowanym w grach, jest grafika 3D. Tworzona jest ona za pomocą dedykowanego zbioru narzędzi, zwanego modelerem. Jednym z przykładowych rozwiązań tego typu jest Blender, aktywnie rozwijane przez Blender Foundation. Podobnie jak opisany powyżej GIMP, jest to narzędzie w pełni darmowe, czemu zawdzięcza wielką popularność i konkurencyjność.

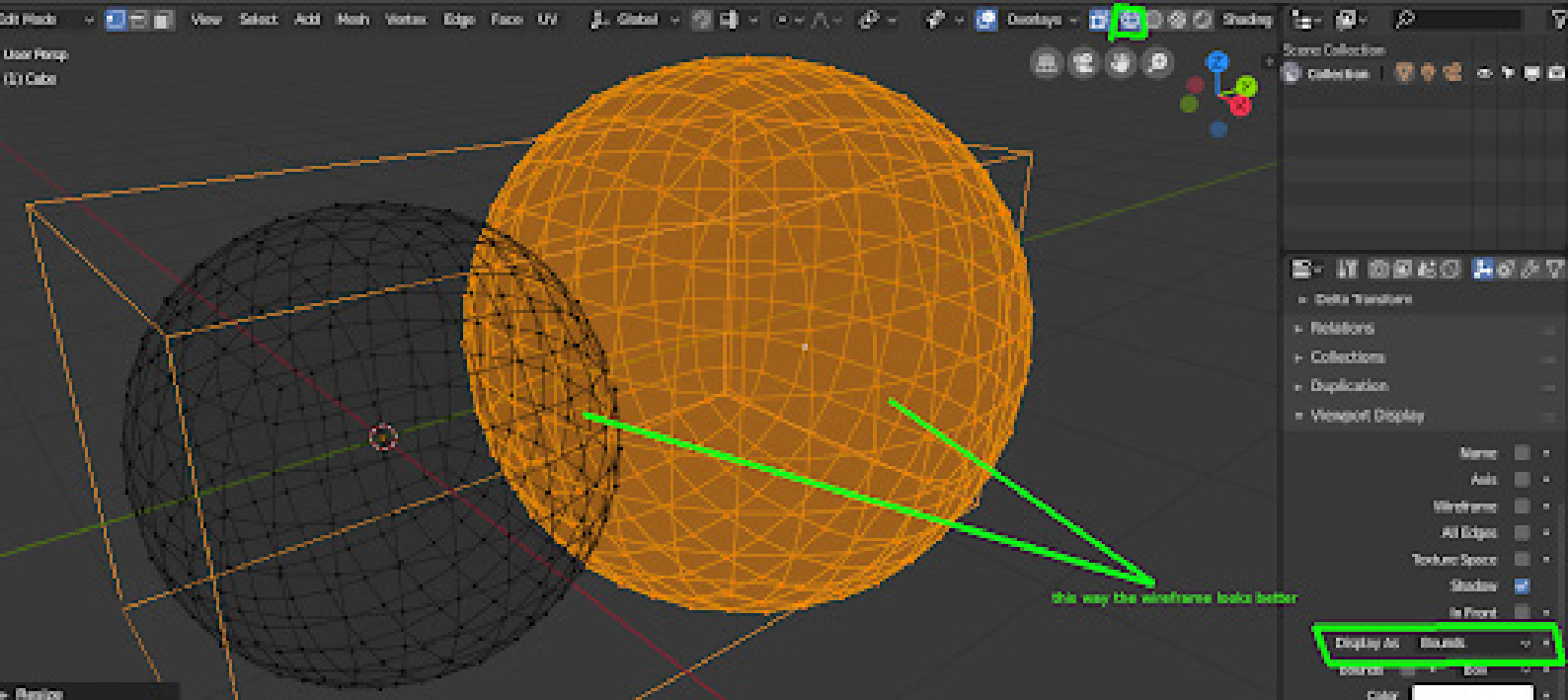
Modeler to narzędzie, za pomocą którego grafik 3D tworzy geometrię poszczególnych modeli, które zostaną wykorzystane w grze i wyrenderowane jako dwuwymiarowy obraz przez silnik graficzny gry.² Najważniejsze cechy popularnych narzędzi do grafiki 3D:

- **MODELOWANIE SIATKI³ WIELOKĄTÓW** – najczęściej stosowana metoda modelowania w grach wideo. Obiekt 3D jest w niej zbudowany z wielu płaskich wielokątów, najczęściej trójkątów lub czworokątów, które mają wspólne krawędzie i wierzchołki⁴. Im

2. Patrz: Silnik graficzny

3. Ang. mesh, w terminologii branżowej używane zamiennie dla słowa "model 3D".

4. Ang. vertex.



Rys. 16. Blender – modelowanie siatki

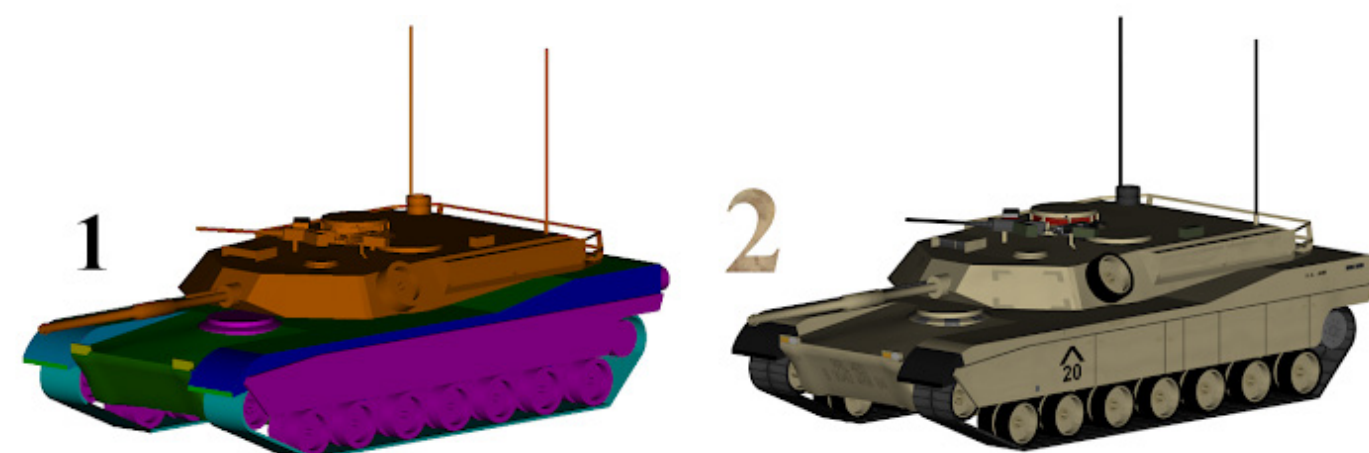
większa gęstość siatki (ilość vertexów), tym dokładniej model może przedstawiać obiekty o skomplikowanej geometrii, lecz również tym większe zapotrzebowanie na moc obliczeniową potrzebną do uruchomienia gry.

- **MAPOWANIE TEKSTUR⁵** – funkcjonalność pozwalająca przypisać do danych obszarów powierzchni modelu 3D odpowiednich

5. Ang. texture mapping

wycinków tekstur, które mają być na nim wyświetlone w grze.

- **TWORZENIE ANIMACJI 3D** – zestaw narzędzi, które pozwalają użytkownikowi modyfikować kształt modelu w celu symulacji ruchu. Przykładem może być animacja biegu postaci ludzkiej – ruch ten składa się z jednoczesnego przemieszczenia wielu kości.



Rys. 17. Blender – mapowanie tekstur; 1 – model bez tekstur, 2 – ten sam model z teksturami.

Silniki gier

Pod nazwą zwyczajową silnik gier⁶ rozumie się zintegrowane środowisko programistyczne – zespół programów – składające się z wielu narzędzi i pomniejszych silników.

Dostępne obecnie silniki zawierają liczne moduły – wbudowane komponenty, które w tradycyjnym procesie tworzenia gier musiały zostać stworzone od zera przez twórców danej gry. W związku z tym, najważniejszą zaletą współcześnie dostępnych silników gier jest znaczna oszczędność czasu oraz kodu wymaganego do stworzenia działającej gry czy nawet jej pojedynczej funkcji.

Środowisko takie jest obecnie głównym narzędziem wymaganym do tworzenia gier wideo. Dzięki licznym korzyściom, jakie niesie ze sobą korzystanie z gotowych silników do gier, znalazły one także zastosowanie w innych sektorach, często wykraczających poza ramy IT, jak choćby oprogramowanie przeznaczone dla biznesu, aplikacje dla klientów instytucji obsługiwane przez urządzenia dotykowe, wizualizacje trójwymiarowe oraz zaawansowane symulatory stosowane podczas szkolenia operatorów maszyn, pilotów i żołnierzy, a także w medycynie.

Większość współczesnych silników gier zawiera:

Kompilator

Najważniejszą częścią gry jest jej kod źródłowy, dzięki któremu zaprojektowana

mechanika gry może zostać wykonana przez urządzenie docelowe. Kompilator to moduł silnika, który przekształca kod źródłowy – wprowadzone przez programistę instrukcje – na język maszynowy, czyli na zestaw rozkazów procesora, który zostanie wykonany po uruchomieniu gry.

Silnik graficzny

Silnik graficzny to moduł silnika gier, który odpowiada za tworzenie grafiki dwu- lub trójwymiarowej, dzięki czemu może być ona wyświetlona na urządzeniu wyświetlającym obraz.

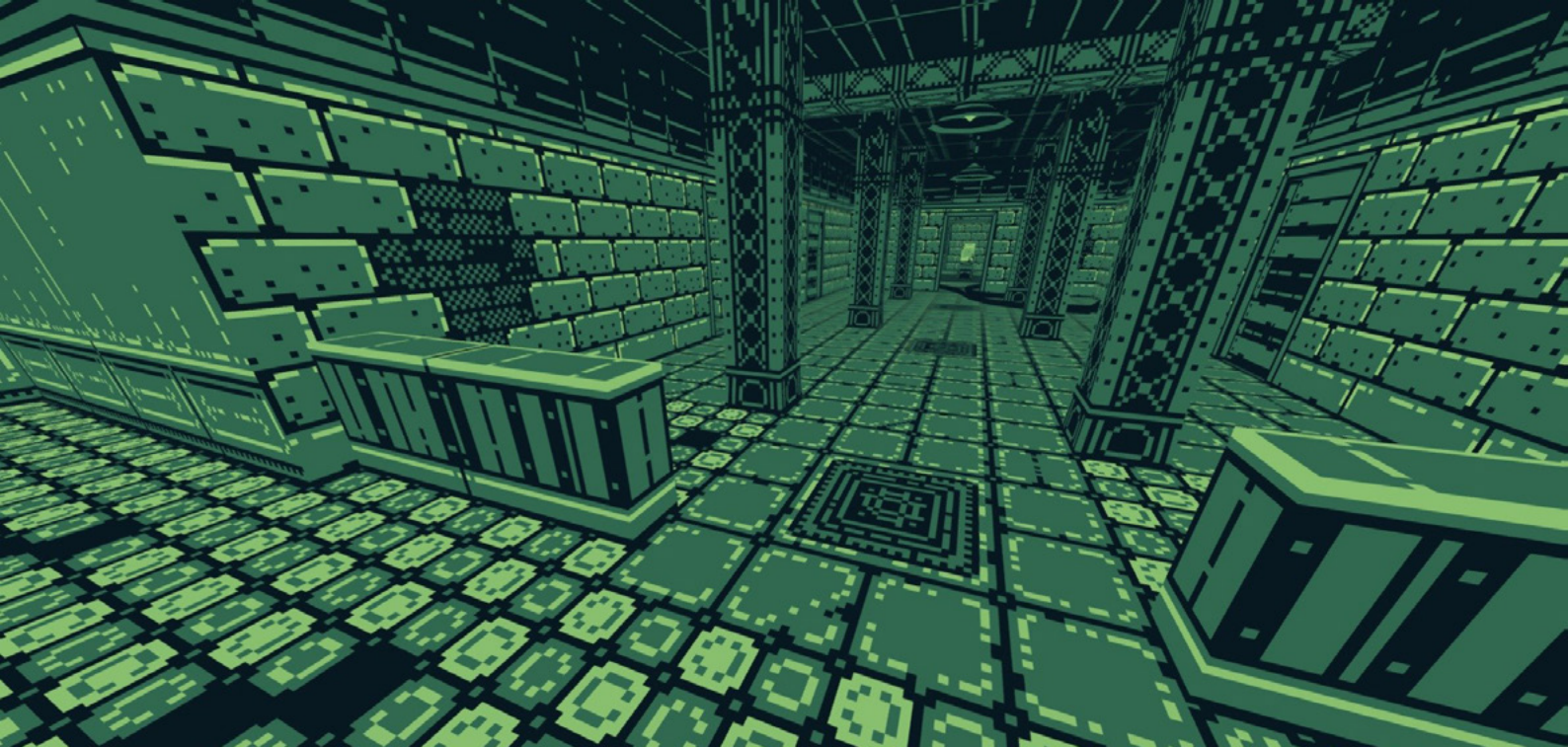
Zależnie od rodzaju silnika oraz rodzaju gry, wyświetlany obraz może składać się w całości z tekstur, czyli dwuwymiarowych grafik, lub zarówno z tekstur jak i trójwymiarowych modeli. W przypadku grafiki trójwymiarowej, silnik graficzny musi zawierać renderer, czyli pomniejszy silnik służący do generowania grafiki 3D w oparciu o przekształcenia elementów grafiki i złożone obliczenia matematyczne.

Ponadto, silnik graficzny odpowiada za przetworzenie oświetlenia, które może być predefiniowane lub generowane dynamicznie. W drugim przypadku umożliwia to stworzenie zaawansowanych efektów graficznych, takich jak cykl dnia i nocy.

Silnik audio

Silnik audio to moduł, który służy do odtwarzania dźwięków oraz utworów muzycznych podczas gry.

⁶. Ang. game engine.



Podstawowe silniki audio zawierają algorytmy pozwalające załadować żądane pliki dźwiękowe do pamięci komputera, a następnie zdekompresować je odtworzyć. Z kolei zaawansowane silniki umożliwiają także symulację złożonych efektów dźwiękowych, takich jak pogłos czy efekt Dopplera, oraz modyfikację odtwarzanych dźwięków w czasie rzeczywistym.

Silnik fizyki

Jest to moduł odpowiedzialny za emulację praw fizyki i zjawisk fizycznych w czasie rzeczywistym. Chociaż możliwości faktycznej symulacji praw fizyki jest w dużym stopniu ograniczona możliwościami współczesnego sprzętu elektronicznego, silniki fizyczne dążą do jak najwierniejszego odzwierciedlenia występujących w grze zjawisk, w szczególności sił fizycznych oraz kolizji zachodzących pomiędzy różnymi obiektami w świecie gry.

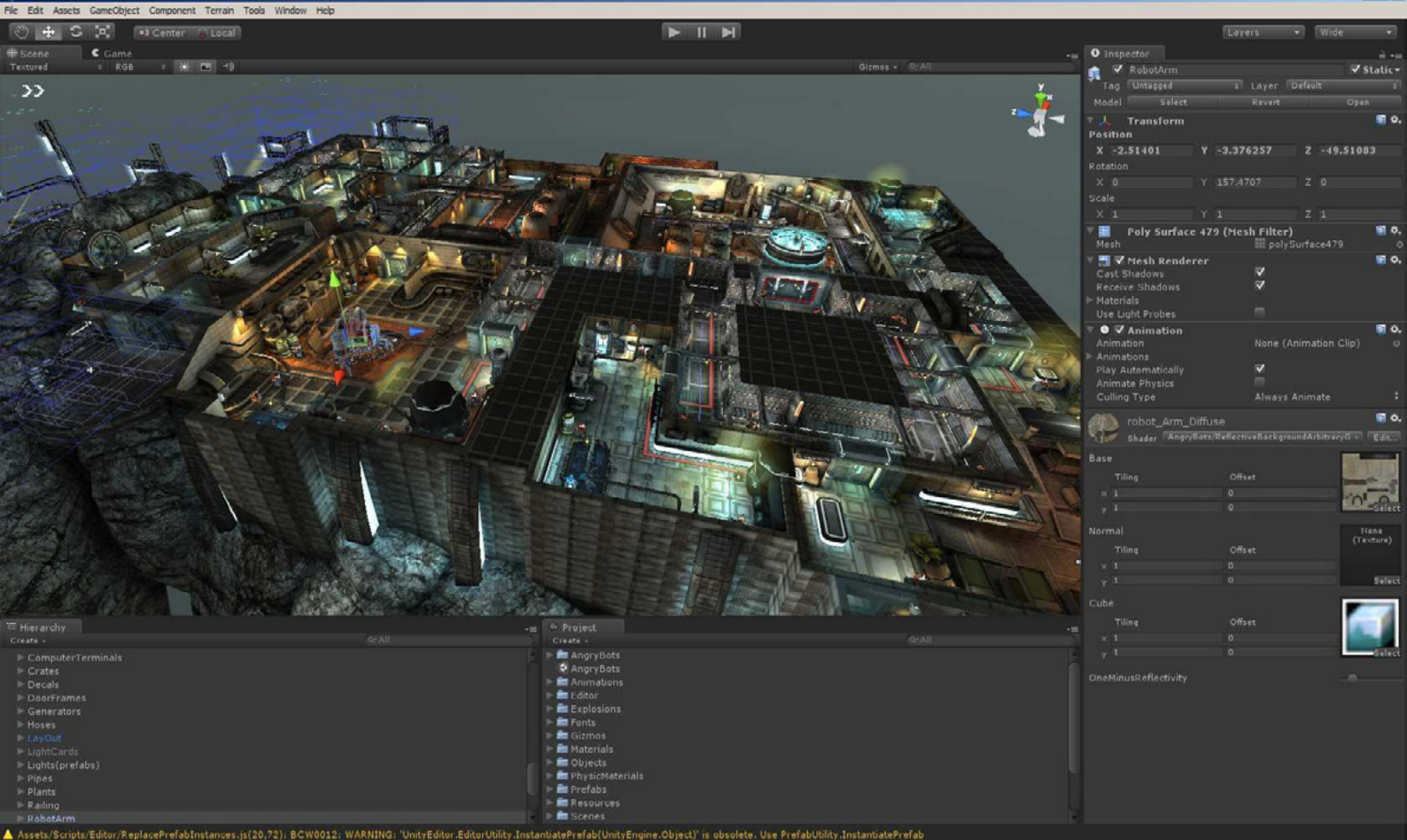
Edytor poziomów/scen

Moduł ten umożliwia stworzenie przestrzeni, w której odbywa się akcja gry. Zależnie od ustalonego charakteru gry, może być to zbiór zamkniętych poziomów lub świat o otwartej strukturze. Za pomocą tego modułu tworzone są kolejne lokacje, a także dodatkowe sceny, takie jak menu główne.

Dzięki temu modułowi, projektant poziomów⁷ rozmieszcza w przestrzeni gry zasoby graficzne, takich jak modele 3D, tekstury, źródła światła, obiekty interaktywne czy spawnery (obiekty, które generują żądane zasoby: przeciwników, postaci poboczne czy obiekty, które gracz może zebrać). W ten sposób powstają lokacje, w których odbywa się akcja gry.

Wiele edytorów poziomów zawiera także narzędzia służące do modyfikacji terenu w formie mapy wysokościowej i edycji szaty roślinnej występującej na terenie.

7. Ang. level designer – osoba zajmująca się tworzeniem geometrii poziomów, w których rozgrywa się akcja gry.



Rys. 18. Silniki gier – Unity

Współczesne silniki

Obecnie największą popularnością wśród silników do gier cieszą się *Unity3D* oraz *Unreal Engine 4*. Są to kompletne i uniwersalne komercyjne silniki, które umożliwiają tworzenie różnorodnych złożonych gier z wysokiej jakości grafiką. Istotne jest również, że są przy tym dostępne za darmo do użytku prywatnego, edukacyjnego oraz, do pewnego stopnia, komercyjnego.

Unity3D

Jest to wieloplatformowy silnik stworzony przez *Unity Technologies*, korzystający z języka C#. Wieloplatformowość oznacza, że każda gra może zostać wydana na wiele platform docelowych naraz, co umożliwia

znaczne zwiększenie kapitalizacji danej gry. Wybrane platformy wspierane przez *Unity3D*:

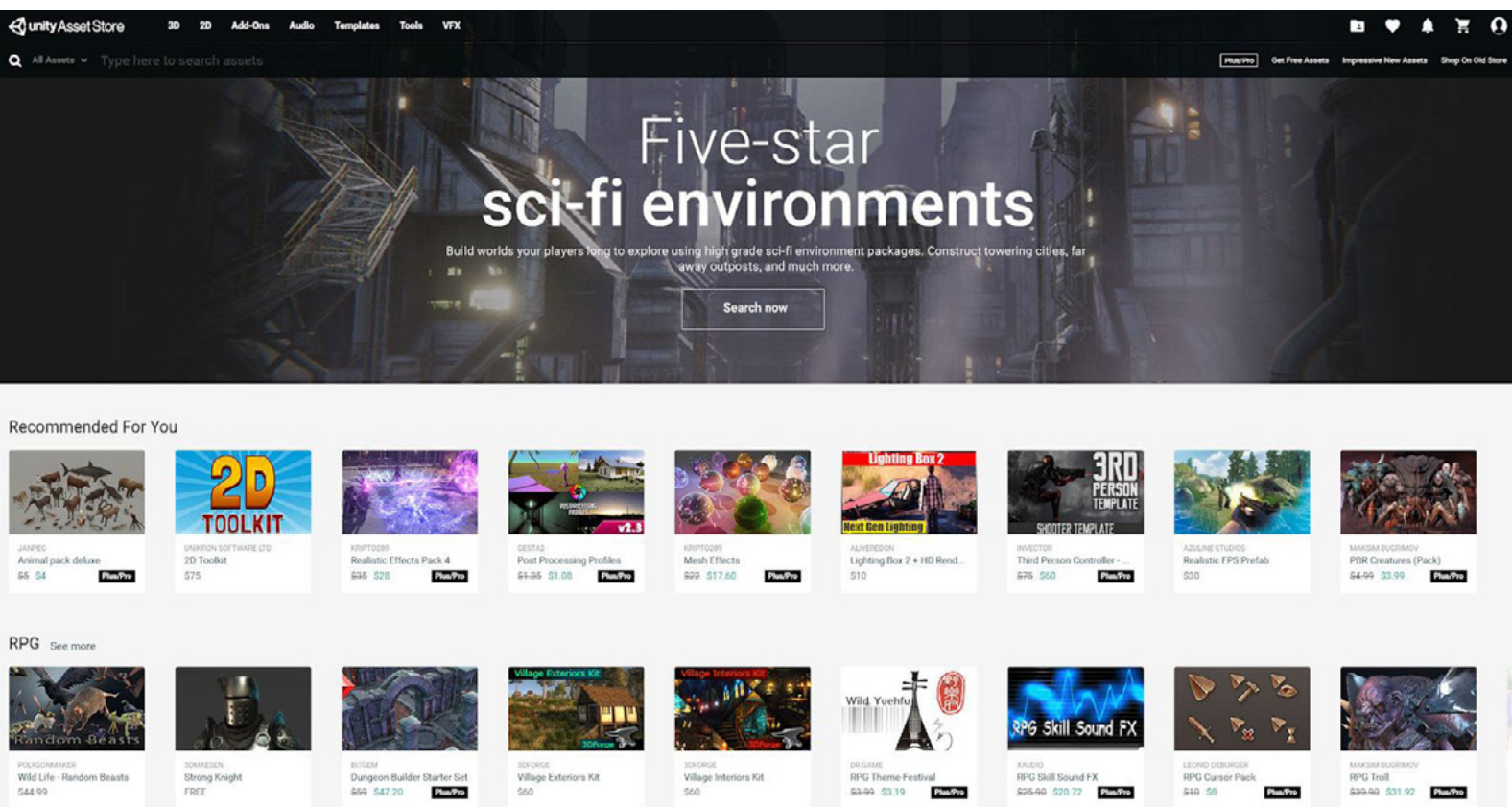
- Windows
- Linux
- Mac
- PlayStation 4
- Playstation Vita
- Xbox One
- Nintendo 3DS
- Nintendo Switch
- iOS
- Android
- Android TV

Kolejne zalety Unity3D to łatwość obsługi, dostęp do rozbudowanej dokumentacji oraz powszechność tego silnika. Jest to "klasyczny" silnik korzystający z wielu standardowych rozwiązań stosowanych przez lata zarówno w branży gier, jak i w szeroko rozumianym IT. Z tego powodu jest on najczęściej polecanym silnikiem dla osób, które dopiero zaczynają naukę tworzenia gier. Z drugiej zaś strony, silnik ten jest coraz częściej wybierany przez uznane zespoły deweloperskie jako znacznie tańsza alternatywa dla tworzenia własnych silników. Przemawia za tym również ciągłe wsparcie techniczne oferowane przez twórców Unity3D.

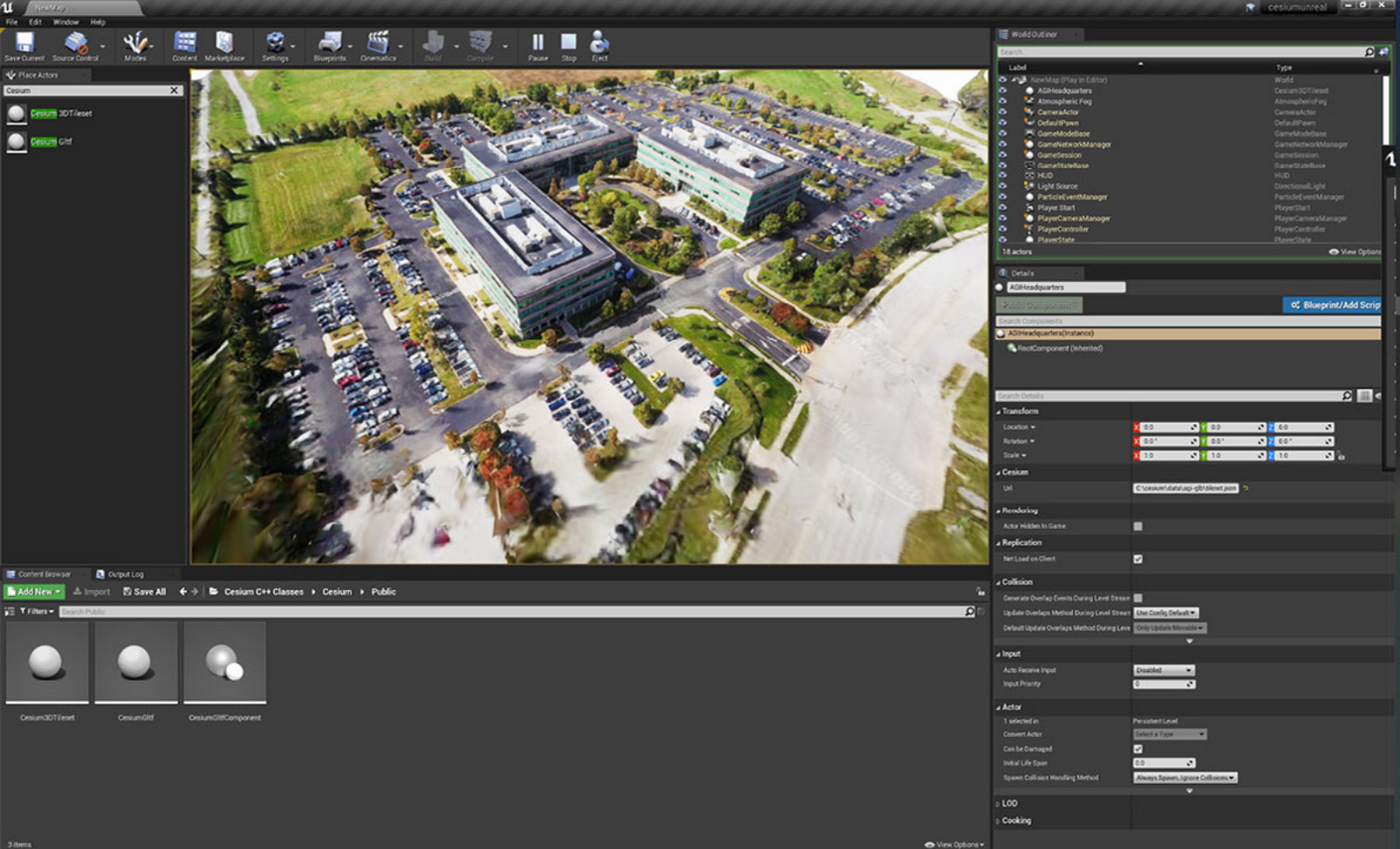
Z racji swojej popularności, Unity3D ma bardzo rozbudowaną społeczność, któ-

rej członkowie aktywnie pomagają sobie wzajemnie w rozwiązywaniu napotkanych problemów wynikających z korzystaniem z silnika.

Ponadto, Unity3D to jeden z pierwszych silników, który oferuje wbudowany rynek zasobów: Unity Asset Store. Za pośrednictwem tej platformy, która działa aktywnie od 10 lat, użytkownicy Unity3D mogą pobierać stworzone przez innych członków społeczności zasoby – darmowe oraz płatne – które później wykorzystają w tworzeniu swoich gier lub podczas prototypowania nowej gry. Obecność platformy Unity Asset Store przyczyniło się do ogromnej popularności silnika Unity3D, a także, w skali globalnej, do znacznego zmniejszenia "bariery wejścia" dla początkujących twórców gier.



Rys. 19. Unity – Asset Store



Rys. 20. Silniki gier – Unreal Engine 4

Unreal Engine 4

UE4 to silnik stworzony przez *Epic Games* działający w języku C++. Podobnie jak *Unity3D*, jest to silnik wieloplatformowy, któremu również towarzyszy platforma dystrybucji zasobów o nazwie *Unreal Engine Marketplace*. UE4 zawiera znacznie więcej niestandardowych rozwiązań.

Jedną z największych zalet UE4 jest jego silnik graficzny, który umożliwia osiągnięcie niezwykle realistycznej grafiki – do tego stopnia, że UE4 jest obecnie szeroko wykorzystywane do wizualizacji architektonicznych.

Ponadto, UE4 wyróżniają jego ułatwienia umożliwiające szybkie prototypowanie gier. Szczególnie istotny jest tu *Blueprints* – system programowania graficznego (ang.

visual scripting, visual programming), który pozwala na tworzenie działającego kodu za pomocą systemu blokowego w formie szablonów, nie wymagając przy tym właściwego programowania.

Ponadto, UE4 posiada system *Brushes* – form geometrycznych, które umożliwiają projektantom poziomów szybkie prototypowanie rozbudowanych poziomów gry przy użyciu predefiniowanych kształtów trójwymiarowych modeli, bez angażowania dedykowanego grafika 3D.

Mimo wspomnianych rozwiązań, które znacznie przyspieszają proces prototypowania, UE4 jest postrzegany jako bardziej zaawansowany w obsłudze silnik niż *Unity3D* i z tego powodu nie jest tak często polecany początkującym twórcom.

Podsumowanie

Docieramy do końca naszego kompendium. Omówiliśmy w nim najważniejsze style i gatunki gier, przybliżyliśmy proces ich produkcji oraz przedstawiliśmy podstawowe narzędzia stosowane przez twórców.

Czas więc przejść do wniosków, choć nie wszystkie będą optymistyczne. Należy przede wszystkim pamiętać, że ogromna większość powstających gier zostaje porzucona przed ukończeniem. Z kolei spośród projektów, które doczekały się wersji finalnej, mało która jest w stanie wygenerować zysk choćby pokrywający koszty produkcji. Z tych zaś nielicznych twórców, którym ta sztuka się uda, mało który może mówić o jakimkolwiek sukcesie – czy to finansowym, czy wizerunkowym. Jest to spowodowane ogromnym nasyceniem rynku, i tego nie przeskoczmy.

Nowych gier powstaje codziennie tyle, że potencjalny konsument nie ma szans zaznajomić się nawet z tymi, które trafiają bezpośrednio w jego gusta. Oznacza to, że tworzenie gier wiąże się z ryzykiem – kosztu produkcji mogą się nie zwrócić, gra może przeminąć bez echa, wreszcie twórca może doznać “wypalenia”, nim jego projekt będzie gotowy do publikacji. Nie znaczy to jednak, że mamy porzucić tworzenie gier – wręcz przeciwnie! Choć współcześni twórcy gier muszą mierzyć się z zupełnie

innymi wyzwaniami, niż pionierzy branży, to właśnie sprawia, że nasze gry stają się coraz lepsze.

Istotna jest też świadomość, że taka ekspansja rynku gier wideo jest skutkiem znacznego uproszczenia samego procesu produkcji gier. Dzięki darmowym narzędziom, takim jak kompletne silniki gier, edytory graficzne czy systemy skryptowania wizualnego, do których w dodatku powstają setki darmowych poradników, stworzenie gry jest łatwiejsze niż kiedykolwiek – nawet w pojedynkę i bez doświadczenia. Jednak to ostatnie odgrywa decydującą rolę i determinuje, czy gra odniesie sukces, czy się zwróci, czy – przede wszystkim – jej tworzenie będzie sprawiało przyjemność. I choć wiedzy teoretycznej na temat tworzenia gier znajdziemy zatrzęsienie, niezbędne jest doświadczenie praktyczne. To zaś zdobywamy tworząc kolejne gry, od początku aż do ukończenia produkcji.

Rozwój twórcy gier można przyrównać do metafory procesu kształcenia pilota samolotu. Startuje on z pełnym workiem szczęścia i pustym workiem umiejętności – sztuka polega na tym, by napełnić worek z umiejętnościami, zanim worek ze szczęściem się opróżni. Analogicznie, choć początkujący twórca gier nie posiada umiejętności, to dysponuje zapałem i kreatywnością. Poniższe zasady mają za

zadanie zapewnić, że w trakcie produkcji twórca nabierze wystarczająco dużo umiejętności i doświadczenia, zanim wyczerpie się jego zapał i inwencja twórcza, dzięki czemu uniknie zniechęcenia wywołanego potencjalnymi porażkami. Najważniejsze zasady to:

- **START SMALL/KEEP SMALL** – rozpoczęcie przygody z tworzeniem gry rozbudowanego i bardzo ambitnego projektu (jak RPG z nieliniową fabułą czy gra MMO) w ogromnej większości przypadków prowadzi do frustracji i porzuconych produkcji. Zaczynając od prostych projektów zdobywamy po kolei niezbędne umiejętności, nie gubiąc się jednocześnie w trakcie produkcji.

- **GET THINGS DONE** – coś jest zrobione, gdy jest dostępne w grze oraz gdy projektant tego elementu wraz z testerem potwierdza, że działa on zgodnie z założeniami. A dopóki nie jest zrobione, pozostaje jedynie pomysłem – pomysły zaś są tanie i ma je każdy.

- **FAIL FASTER** – im szybciej poniesiemy porażkę, tym większe mamy szanse na

wdrożenie poprawek i finalnie osiągnięcie sukcesu. Wymaga to pewnej dozy pokory, jednak oszczędza mnóstwo czasu i frustracji.

- **WIZJA GRY TO ŚWIĘTOŚĆ** – wizja stanowi zestaw wytycznych, do których odwołujemy się przez cały okres produkcji. Ten dokument stanowi o całym kierunku produkcji. Zmiana wizji w trakcie produkcji prowadzi przeważnie do sprzeczności różnych założeń, przez co może zaistnieć konieczność usunięcia lub przebudowy już wdrożonych systemów. W wielu przypadkach zajmuje to więcej czasu, niż dokończenie gry według pierwotnej wizji i stworzenie kolejnej. Dlatego unikamy tego jak ognia, a wizję traktujemy jak konstytucję.

Jednakże pamiętając o powyższych zasadach, możemy uniknąć wielu błędów i przejść przez etap tworzenia z uśmiechem na ustach. Mamy nadzieję, że to kompendium rozjaśni wiele niepewnych obszarów i przybliży nie jeden zespół do wydania świetnej gry. Pamiętaj, że tworzenie gier powinno przede wszystkim sprawiać Ci przyjemność!





Ministerstwo Cyfryzacji

Dofinansowano ze środków Ministerstwa Cyfryzacji.



Mistrzostwa w Projektowaniu Gier Komputerowych



TEATRIKON
TEAM



Lublin
GameDev