



**Politechnika Krakowska
im. Tadeusza Kościuszki**

Wydział Fizyki, Matematyki i Informatyki



Łukasz Pytlak

Numer albumu: 99909

**Analiza jakości i wydajności wybranych
silników uruchamiania gier sieciowych na
podstawie gry typu MOBA**

**Quality and performance analysis of selected
game network engines based on MOBA-type
game**

**Praca magisterska
na kierunku Informatyka**

Praca wykonana pod kierunkiem:
Dr inż. Paweł Jarosz

Uzgodniona ocena:.....

.....
podpisy promotora i recenzenta

Kraków 2019

Spis treści

Streszczenie pracy.....	4
Abstract.....	4
1. Cel pracy	5
2. Przegląd literatury	6
Kontekst historyczny – historia gier komputerowych	6
Gry komputerowe	9
Gry darmowe?	12
Gry typu MOBA	14
Matchmaking.....	20
Liga Legend.....	23
E-sport.....	25
Architektura Klient-Serwer.....	34
Tworzenie własnej gry, silniki sieciowe i badanie ich jakości	35
3. Specyfikacja wymagań	36
3.1. Projekt gry	36
3.2. Wymagania нефunkcjonalne	36
4. Metodyka pracy.....	38
4.1. Wykorzystane narzędzia	38
4.1.1. Unity 3D	38
4.1.2. Visual Studio.....	39
4.2. Wykorzystane technologie.....	40
4.2.1. C# (C Sharp).....	40
4.3. Wykorzystane usługi sieciowe	41
4.3.1. Unity Networking - UNet	41
4.3.2. Photon i PUN	42
4.3.3. GameSparks.....	45
5. Wykonanie prototypu gry.....	54
5.1. Organizacja projektu.....	54
5.2. Implementacja funkcjonalności	55
6. Wyniki.....	70
6.1. Analiza otrzymanych wyników	70
7. Wnioski.....	73
7.1. Wnioski z otrzymanych danych	73
7.2. Możliwość dalszego rozwoju	75

Podsumowanie	76
Adnotacje i słownik	78
Bibliografia	80
Spis rysunków	85
Spis tabel	86

Streszczenie pracy

Rynek gier jest coraz prężniej rozwijającą się gałęzią branży IT. Przychody ze sprzedaży gier z samego 2018 roku przekraczają oszałamiającą kwotę blisko 140 miliardów dolarów.

51% udziałów stanowią gry na urządzenia mobilne, 25% to gry konsolowe i 24% udziałów w zyskach przypadło produkcjom na komputery osobiste, niemniej jednak w dalszym ciągu mowa o przychodach około 33 miliardów dolarów.

Co jednak z grami typu *free-to-play*? W jaki sposób one zarabiają i jaki mają wkład w informacje podane powyżej?

Jedną z rzeczy, które znajdą się w tej pracy będzie omówienie gier typu *MOBA*, ich historii i ich udział w rynku gier komputerowych. W pracy podane, a następnie krótko omówione zostaną poszczególne tytuły oraz ich znaczenie w branży gier darmowych.

Następnie poruszone zostanie główne, tytułowe zagadnienie pracy, którym jest analiza jakości i wydajności wybranych silników uruchamiania gier sieciowych na podstawie gry typu *MOBA (Multiplayer Online Battle Arena)*.

Przedstawione w pracy zostaną niektóre istniejące silniki/usługi sieciowe i rozwiązania z nimi związane, wraz z ich późniejszą oceną.

Abstract

The gaming market is an ever-growing branch of the IT industry. Revenues from the sale of games from the year 2018 exceed a staggering amount of almost 140 billion USD.

51% of shares are games for mobile devices, 25% are console games and 24% shares in profits have been generated for production on personal computers, nevertheless, there are still revenues of about 33 billion USD.

But what about free-to-play games? How do they earn and how they contribute to the information given above?

One of the things that will be included in this work will be discussing MOBA-type games, their history and their participation in the computer games market. The work given and then briefly discusses the individual titles and their meaning in the free games industry.

Next, the main, title issue of the work will be discussed, which is the analysis of the quality and performance of selected engines for running network games based on the MOBA type game (Multiplayer Online Battle Arena).

Some of existing engines / network services and related solutions will be presented at work, along with their subsequent evaluation.

1. Cel pracy

Głównym celem pracy dyplomowej magisterskiej jest analiza jakości i wydajności wybranych silników/usług sieciowych niezbędnych do uruchamiania gier online na przykładzie gry typu *MOBA*.

Kolejnym celem jest uświadomienie czytelnika czym jest rynek gier komputerowych, na jakie gry obecnie panuje moda, czym dokładnie jest system F2P i na czym polega oraz przede wszystkim jakie znaczenie dla rynku gier ma e-sport.

Dalej zostanie przedstawiona rola silników gier takich jak Unity 3D w procesie tworzenia projektu, zostaną przedstawione inne dostępne na rynku silniki i rozwiązania oraz zostaną omówione *Unity Networking*, *Photon Unity Networking* i *Gamesparks*.

W kolejnym kroku praca jak najprościej przeprowadzi czytelnika przez proces tworzenia gry komputerowej typu *MOBA* w oparciu o darmową wersję silnika gier Unity 3D. Przedstawione zostaną następujące etapy i procesy zachodzące przy tworzeniu gry:

- prezentacja funkcjonalności Unity wraz ze zaznajomieniem czytelnika z *AssetStorem*,
- przebieg wyboru zasobów ze sklepu *Assetów*,
- tworzenie mapy gry,
- *Pathfinding*,
- *Matchmaking*.

Utworzona aplikacja posłuży za tester niektórych usług sieciowych. Przeprowadzone zostaną badania porównujące wszystkie powyższe zbiory usług pod kątem użyteczności i jakości.

Na podstawie otrzymanych danych przedstawione zostaną wyniki, a w kroku następnym praca przedstawi wnioski z nich wynikające.

Kończąc podsumowany zostanie przebieg postępów nad pracą i zostanie oceniony proces jej tworzenia wraz z niego wynikającymi problemami i nabytą w trakcie pisania wiedzę.

Ostatnimi rozdziałami owej pracy będą słownik niektórych pojęć wykorzystanych w pracy oraz bibliografia, na podstawie której autor czerpał wiedzę niezbędną do stworzenia pracy.

2. Przegląd literatury

Kontekst historyczny – historia gier komputerowych

Historia gier komputerowych sięga połowy lat 50. ubiegłego wieku, kiedy naukowcy zajmujący się komputerami zaczęli projektować proste gry i symulacje w ramach swoich badań. I choć za pierwsze gry komputerowe można przyjąć powstałe tuż po II wojnie światowej symulatory wystrzelenia pocisku raketowego to prawdziwy rozwój gier komputerowych przyjmuje się na lata 50. i 60. ubiegłego wieku. Wtedy to powstała jedna z pierwszych serii gier uznanych za komputerowe – w roku 1958 William Higinbotham stworzył symulację tenisa stołowego o nazwie *Tennis for Two*, pierwowzór późniejszej gry *Pong*. Wykorzystał do tego oscyloskop i pomimo braku generowanego sygnału wideo gra została uznana za pierwszą grę w historii branży twórców gier.[19]

Kolejnym z większych tytułów na tamte czasy była gra *Spacewar!* opracowana w 1962 roku przez Stephena Russella i jego zespół na M.I.T. na jeden z pierwszych komputerów PDP-1. Gra z miejsca stała się hitem w małych środowiskach programistycznych. Praktycznie z miejsca ludzie rozpoczęli jej reprodukcję na inne urządzenia na szeroką skalę, aż we wczesnych latach 70. trafiła na pierwsze mikrokomputery.[20]



Rysunek 1. *Spacewar!* na komputerze PDP-1 firmy DEC

Do obiegu masowego gra *Tennis for Two* trafił dzięki wcześniej wspomnianej produkcji *Pong* w roku 1972 autorstwa firmy Atari, do jej popularyzacji przyczyniło się przede wszystkim pojawienie się automatów do gier oraz pierwszych konsol (z kolei te zyskały popularność dzięki takim produkcjom jak *Space Invaders* czy *Pac-Man*). [19]



Rysunek 2. Automat do gry w Pong

W latach 70. i 80. XX wieku do wolnej sprzedaży weszły pierwsze domowe konsole do gier z joystickami, przyciskami i innymi kontrolerami wraz z dostępem do grafiki na ekranie monitora (domowego komputera lub telewizora). Od lat 80. w wielu miejscach na świecie gry wideo stały się popularną formą rozrywki oraz częścią współczesnej popkultury. Miejsca do gry na dużych, przyozdobionych kolorowymi grafikami automatach na monety otwierano wszędzie. Salony gier powszechne były przede wszystkim w centrach handlowych, pełne gier zręcznościowych, gier akcji i strzelanek. Równoległy rozwój konsol domowych sprawił, że grać można było wszędzie, w domu, w szkole, na zakupach. Popularyzacja gier na automaty spowodowała zyski w postaci około 5 do 7 miliardów dolarów.[19]



Rysunek 3. Atari 2600 – jedna z pierwszych popularnych konsol domowych

Lata mijaly, kolejne generacje konsol wychodziły z mniejszym lub większym sukcesem naprzeciwko stale rosnącym wymaganiom nowych i starych graczy. W piątej generacji konsol domowych po raz pierwszy pojawiła się firma Sony wraz ze swoim PlayStation (1994 rok).[19]



Rysunek 4. Sony PlayStation

Sześć lat później wraz z nadejściem szóstej generacji konsol Sony wypuściło na rynek konsolę PlayStation 2 – jedną z pierwszych konsol z napędem DVD na płyty o pojemności 4.7Gb. PS2 zapewniała lepszą grafikę od swojego poprzednika oraz umożliwiała równoczesną rozgrywkę nawet 4 graczom. [19]

Rok później swoją premierę miała również pierwsza konsola firmy Microsoft, 15 listopada 2001 roku światu ukazał się Xbox. Bazująca na procesorze Intel Pentium III konsola korzystała z dużej ilości technologii zaciągniętej z komputerów stacjonarnych dzięki czemu gry przeznaczone na PC łatwo można było przenosić na konsolę. Aby zdobyć udziały w rynku, Microsoft przyjął strategię tańszego sprzedawania konsol z równoczesną koncentracją w kierunku czerpania zysków z tworzonych i publikowanych gier.[19]



Rysunek 5. Pierwszy Xbox firmy Microsoft

Dziś rynek gier komputerowych jest bardziej dochodowych zajęć. Najbardziej podstawowym sposobem podziału udziałów rynkowych jest podział ze względu na urządzenia docelowe, z których korzystają gracze. Wymienia się:

- gry na urządzenia mobilne stanowiące lwią część rynku - 51% przychodów z całej branży gier, tj. ponad 70 mld USD rocznie;
- gry konsolowe z przychodami za ubiegły rok w wysokości 35mld USD;
- gry komputerowe i ich przychody w okolicach 33 mld USD.[1]

To co należy dodać w tym miejscu to to, że z każdym upływającym rokiem przychody w każdej z tych trzech sekcji rosną. Jak podaje Tom Wijman z portalu *newzoo.com*, zajmującego się analizą danych gier i e-sportu, na przestrzeni ostatniej dekady producenci gier osiągnęli wzrost o 110 mld USD. Przychód z produkcji gier w 2012 roku wyniósł „zaledwie” 70 mld USD. 11% wzrost każdego roku.[1]

Co się tyczy gier na komputery osobiste Wijman uważa, że te osiągnęły już szczyt swojej monetyzacji i przychody z nimi związane będą się stopniowo zmniejszać. Jednym z powodów tego zjawiska jest rosnąca liczba gier darmowych.[1]

Gry komputerowe

Gry komputerowe powstały w celu zaspokojenia ludzkiej potrzeby jaką jest rozrywka. Od wielu lat skutecznie rywalizują z grami planszowymi, filmami czy innymi formami zabawy. Fakt rozprzestrzenienia się gier w połączeniu z ludzkim zapotrzebowaniem na rozrywkę stał się tematem wielu dyskusji. O grach zaczęto mówić na różnych płaszczyznach, m.in. socjologicznej, psychologicznej, kulturowej i sportowej.

Jeśli chodzi o aspekt socjologiczny gier to Augustyn Surdyk[2] w swojej pracy *Edukacyjna funkcja gier w dobie „Cywilizacji Zabawy”* pisze następująco:

[...]gdyby porównać obecne zainteresowanie grami komputerowymi z popytem na inne gry sprzed „ery komputerów”, prawdopodobnie uzyskalibyśmy te same dane statystyczne, zmieniłby się jedynie rodzaj gier.[2]

Widać tu jasno postawioną przez Pana Surdyka tezę – ludzie bawią się, grają tak samo w gry jak to robili do tej pory z tą różnicą, że zmienił się tylko ich rodzaj. Dalej autor tekstu pisze jeszcze:

[...]wśród przedstawicieli najmłodszego pokolenia szeroko rozumiane gry cyfrowe wypierają choćby tradycyjne gry towarzyskie, jak karciane lub klasyczne gry planszowe (szachy, warcaby, chińczyk itp.)[...] [2]

Dzięki tej wypowiedzi widać, że gry cyfrowe, gry komputerowe mają coraz większy wpływ na młodą część społeczeństwa. Poza funkcją zapewnienia ludziom rozrywki gry docierając do najmłodszych graczy spełniają również funkcje edukujące (gracze angażują się nieraz emocjonalnie w rozgrywkę), przykładem tego jest choćby gra WARSAW będąca dwuwymiarową grą historyczną opowiadającą o powstaniu warszawskim.

Aspekt psychologiczny gier komputerowych dotyczy negatywnego wpływu gier na psychikę graczy. Głównymi zarzutami rzucanymi przez część naukowców są pobudzenie agresywności wśród graczy wraz ze zbyt dużą emocjonalną postawą odbiorczą występującą w przypadku gier w większym stopniu niż w przypadku innych mediów. Kolejnym zarzutem było oderwanie od rzeczywistości oraz skłonności graczy do popadnięcia w uzależnienie. [21] W większości przypadków te zarzuty nie mają pokrycia w przeprowadzonych badaniach. Dodatkowo część ludzi uważa, że gry komputerowe mogą wspomagać rozwój człowieka lub nawet przyczynić się do uratowania życia drugiej osoby. I nie chodzi tu wcale o gry wyścigowe i kontrolę za kierownicą. Ten temat w swoim artykule z listopada 2012 roku porusza Pani Hannah Orenstein z *The Huffington Post*[3].

Artykuł traktuje o eksperymencie przeprowadzonym przez naukowców z uniwersytetu Texas Medical Branch w Galveston, w którym to owi naukowcy przeprowadzili badanie na trzech grupach ludzi. Pierwszą grupę stanowili gracze będący uczniami liceum, drugą grupą byli gracze na co dzień uczęszczający do koledżu a ostatnią grupę stanowili rezydenci medyczni. Te trzy grupy miały za zadanie rywalizować ze sobą w tym, która grupa najlepiej będzie posługiwała się wirtualnymi narzędziami chirurgicznymi. Licealiści grający dziennie średnio dwie godziny, studenci grający około czterech godzin dziennie i rezydenci, z których większość nie spędzała czasu wolnego przed komputerem przeprowadzili szereg zadań na maszynie, która imitowała warunki operacji chirurgicznej. Urządzenie mierzyło przy okazji umiejętności badanych w trzydziestu dwóch kategoriach, w tym koordynację na linii ręka-oko

i czas. Wygrała najmłodsza grupa badanych.[3]

W swoim artykule Orenstein pisze ponadto:

Wcześniej tego roku wynalazcy z Nowej Zelandii stworzyli grę komputerową zaprojektowaną do leczenia depresji wśród nastolatków. Gracze obrali za cel zniszczenie „ponurych, negatywnych myśli” przy pomocy interaktywnego świata fantasy. Czterdzieści cztery procent graczy zostało wyleczonych z depresji w porównaniu do dwudziestu sześciu procent uczęszczających na zwykłą terapię.[3]

Jak widać powyżej, gry komputerowe potrafią przyczynić się do wielu wspaniałych rzeczy jeśli tylko się skorzysta z nich w odpowiedni sposób.

Aspekt kulturowy przejawia się w grach w formie czerpania inspiracji z danych źródeł. Książki, komiksy, filmy, muzyka. Wszystkie wcześniej wymienione media oddziałują na gry komputerowe, są paliwem dla ich twórców i przyczyniają się wespół z grami w rozwój kultury masowej. Jednym z przejawów oddziaływania innych mediów w grach są często spotykane w nich tzw. *easter egg*i – na pozór ukryte przed graczami treści odwołujące się w konkretny sposób do popkultury.[21] Zdarzają się również też takie przypadki, że deweloperzy puszczaają oko sobie nawzajem. Takie zdarzenie miało miejsce m.in. w grze *Max Payne 2: The fall of Max Payne*. W jednym z pomieszczeń twórcy gry umieścili wyłączony telewizor, który po włączeniu przedstawiał scenę z zupełnie innej gry ale też wyprodukowanej przez firmę *Rockstar*, pierwszej części gry *Grand Theft Auto*.



Rysunek 6. Kadr z filmu umieszczonego w ramach easter egga GTA w grze Max Payne 2

Gry komputerowe tak jak większość innych mediów mogą być podzielone na gatunki. Gatunki te są używane do klasyfikowania gier na podstawie ich interakcji z użytkownikiem w grze i każdy z nich definiowany jest przez zestaw wyzwań związanych z rozgrywką. Jest klasyfikowany niezależnie od ustawień gry lub zawartości jej świata. Jest to przeciwieństwo tego co dzieje się np. w przypadku innych dzieł opartych na fikcji literackiej takich jak filmów czy książek.[22] Przykładem jest gra typu strzelanki, będzie ona tym gatunkiem niezależnie czy miejscem akcji jest świat fantasy czy przestrzeń kosmiczna.

Obecnie rynek gier komputerowych może pochwalić się ponad 40 gatunkami gier komputerowych. Przykładami tych najpopularniejszych wraz z konkretnymi tytułami gier są m.in.:

- strzelanki - gry z serii Battlefield i Call of Duty, CS:GO,
- przygodowe, akcji - GTA V,
- gry walki – seria Tekken, Dragon Ball FighterZ,
- MOBA – League of Legends, Dota2, HotS,
- battle royal – PUBG, Fortnite, APEX Legends.

Pisząc o grach komputerowych należy również wspomnieć o tych pozycjach, które na stałe zapisały w się annałach ze względu na swoją popularność i wyniki sprzedaży.

Najlepiej sprzedające się gry w historii[23]:

Tytuł	Sprzedanych egzemplarzy	Platforma	Data wydania	Wydawca
Tetris	170,000,000	Wiele platform	Czerwiec 1984	Spectrum HoloByte
Minecraft	154,000,000	Wiele platform	Listopad 2011	Mojang
GTA V	100,000,000	Wiele platform	Wrzesień 2013	Rockstar Games
Wii Sports	82,860,000	Nintendo Wii	Listopad 2006	Nintendo
PUBG	50,000,000	Wiele platform	Grudzień 2017	PUBG Corporation
Pokemon R/G/B/Y	47,520,000	Wiele platform	Luty 1996	Nintendo
Wii Fit/Wii Fit Plus	43,800,000	Nintendo Wii	Grudzień 2007	Nintendo
Super Mario Bros.	43,170,000	Wiele platform	Wrzesień 1985	Nintendo
Mario Kart Wii	37,140,000	Nintendo Wii	Kwiecień 2008	Nintendo
Wii Sport Resort	33,090,000	Nintendo Wii	Czerwiec 2009	Nintendo

Tabela 1. Ranking najlepiej sprzedających się gier w historii

Najlepsze gry komputerowe wg krytyków i dziennikarzy z branży w oparciu o ranking gier na metacritic.com[10]:

Tytuł	Ocena krytyków	Ocena graczy	Data wydania
Half-Life 2	96	9.1	Listopad 2004
GTA V	96	7.7	Kwiecień 2015
The Orange Box	96	9.2	Październik 2007
Half-Life	96	9.0	Listopad 1998
BioShock	96	8.5	Sierpień 2007
Baldur's Gate II	95	9.2	Wrzesień 2000
Portal 2	95	9.0	Kwiecień 2011

Tabela 2. Najlepsze gry komputerowe wg krytyków i dziennikarzy metacritic.com

Gry darmowe?

System gier free-to-play (F2P, FtP), dla fanów *Nintendo* również znane pod nazwą free-to-start to system gier, które dają graczowi dostęp do znacznej części swojej zawartości bez potrzeby płacenia.[24] Stoją w opozycji do systemu gier pay-to-play, w którym wymaga się od gracza kupienia gry lub uiszczenia części danej opłaty przed skorzystaniem z niej po raz pierwszy. Potocznie w języku polskim są one nazywane grami darmowymi.

Istnieje kilka modeli biznesowych gier free-to-play a najbardziej znanym jest model oprogramowania w wersji *freemium*. W modelu tym gracz zyskuje dostęp do w pełni funkcjonalnej gry, z której może korzystać do woli. Schody zaczynają się dopiero w momencie gdy użytkownik chciałby uzyskać dostęp do dodatkowej zawartości gry w postaci choćby dodatkowych surowców lub waluty. Gracz musi skorzystać z systemu mikropłatności w grze, opłacić konkretny dodatek, który mu się spodobał i wtedy po dokonaniu płatności dodatek dołączany jest do konta gracza.[24]

Omawiany model gry jest najczęściej spotykany w przypadku masowych gier multiplayer online (MMOs), które celują w ściągnięcie do siebie coraz to większych rzeszy okazjonalnych graczy. Takich, którzy mogą zagrać w produkt po raz pierwszy i przy nim pozostać. Ci gracze chętniej korzystają z możliwości grania w coś za darmo, zwłaszcza w towarzystwie swoich przyjaciół i znajomych niż skuszą się na wersję demonstracyjną gry w mocno okrojonej wersji, na którą potem być może nie będzie ich stać.[24]

Kolejnym wartym podania faktem w przypadku F2P jest to, że system ten został bardzo dobrze przyjęty jako forma zapobiegania i walki z piractwem w sieci. Ważnym jest też to, że dzięki grom free-to-play użytkownik często nie musi martwić się wysokimi wymaganiami sprzętowymi pozycji, w którą chce zagrać. Firmy produkujące nowe tytuły zabiegając o graczy muszą stale obserwować rynek sprzętowy starając się odpowiednio zoptymalizować swoje gry by wyjść naprzeciw większej liczbie graczy.[24]

Zwykle rzeczy zakupione przez system mikropłatności w grach FtP nie wpływają na rozgrywkę, ułatwiając grę lub utrudniając ją przeciwnikom. Jednak są pewne gry, w których gracze będący skłonni zapłacić za specjalne przedmioty lub dodatkową jej zawartość mogą zyskać przewagę nad graczami grającymi w pełni za darmo, którzy potrzebowaliby więcej czasu na zakup konkretnej rzeczy w sklepie za ogólnodostępną walutę w grze. Gry tego typu określa się mianem pay-to-win i spotykają się z olbrzymią krytyką graczy i mediów na zachodzie. Inaczej podchodzą do tego gracze w Chinach gdzie społeczność graczy jest przyzwyczajona do płatności za grę choćby ze względu na obecność kawiarenek gamingowych. Przykładem dużej gry, która musiała się zmagać z potężną krytyką ze strony graczy, aż w końcu ograniczyła w większym stopniu system mikropłatności była gra *Star Wars: Battlefront II* od Electronic Arts. W artykule Pawła Musiolika z portalu *ppe.pl*[4] można przeczytać:

Star Wars Battlefront 2 kosztować będzie Electronic Arts znacznie więcej niż utrata zaufania graczy i negatywną prasę. Od dnia premiery, spółka na giełdzie straciła 8% wartości, co przekłada się na 3,1 miliarda dolarów. Jest to pierwszy tak wyraźny spadek wartości akcji całej firmy od ciągłych wzrostów z ostatnich lat.[4]

Chęć jak największego zarobku nie powinna przysłonić producentom oczu na zadowolenie graczy z produktu. W przypadku Battlefronta II frustracja graczy z powodu niemożności wygrania rozgrywki w normalny sposób przełała czarę goryczy co przełożyło się na spadek wartości firmy na giełdzie.

Przykład gier free-to-play w poszczególnych podgatunkach:

Nazwa	Gatunek	Wydawca	Ocena krytyków	Ocena użytkowników
League of Legends	MOBA	Riot Games	78	5.5
Dota2	MOBA	Valve Corporation	90	6.5
Need of Speed: World	MMOR	Electronic Arts	62	5.7
StarTrek Online	MMORPG	Cryptic Studios	66	6.2
Star Wars: Battlefront II	Strzelanka	Electronic Arts	65	1.0

Tabela 3. Tabela przedstawiająca przykładowe gry f2p wraz z ich ocenami na metacritic.com

Przedstawiony model ma przed sobą dobrą perspektywę na przyszłość. Jak uważa redaktor IGN, Charles Onyett „*drogie, jednorazowe zakupy stoją w obliczu wyginięcia*”. Onyett wierzy, że obecny model płatności za grę jednorazowo zniknie w końcu całkowicie. [11] I nie jest to odosobniony głos.

Podobnie uważa m.in. Dr Greg Zeschuk, vice prezydent i współzałożyciel firmy BioWare. Na pytanie Neona Kelly z *videogamer.com* czy jest w stanie wyobrazić sobie, że system free-to-play będzie dominującym modelem na rynku gier, Pan Zeschuk stwierdził, że jest ku temu dobra okazja i okoliczności by ów system stał się dominującym na rynku. Uważa jednak, że nigdy w pełni nie zastąpi innych rodzajów płatności za gry takich jak np. subskrypcji.[12]

Ludzie z Electronic Arts uważają z kolei, że w związku z sukcesem gier modelu freemium mikropłatności wkrótce nieuchronnie staną się częścią każdej wydanej gry. Menadżer generalny i wiceprezydent EA, Nick Earl:

Przyszłość nie polega na jednorazowych płatnościach, przyszłość polega na freemium. Pewna liczba graczy przejdzie na nowy model i może nie będzie płacić dużo ale większość z nich zapłaci więcej niż się ludziom wydaje.[5]

Dalej w wywiadzie dla *gamespot.com* widać, że Earl wychodzi z dość prostego założenia. Jest nim to, że ludzie lubią darmowe rzeczy i gracze chętniej zapłacą za coś czego spróbowali i co im się spodobało.[5]

Gry typu MOBA

Gatunek gier *MOBA* (*Multiplayer Online Battle Arena*) w dużej mierze rozpoczął swój żywot od *Aeon of Strife*, mapy do gry typu *RTS* (ang. *Real-time strategy* – strategia czasu rzeczywistego) o nazwie *StarCraft*, w której czwórka graczy kontrolujących jedną potężną jednostkę wspomaganą przez słabsze, sterowane komputerowo jednostki, została skierowana do gry z w pełni sterowanym przez komputer przeciwnikowi. Gra kończyła się w momencie gdy kluczowe budynki, w jednej z baz, graczy lub komputera, ulegały zniszczeniu lub gdy cała czwórka graczy zginęła. Niedługo później dodano do gry możliwość rozgrywki *PvP* (*Player vs Player*) rzucając wcześniej wspomnianą liczbę czterech graczy do walki ze sobą w dwóch dwuosobowych drużynach.[25]



Rysunek 7. Kadr z gry *Starcraft: Aeon of Strife*

Kolejnym ważnym krokiem w kierunku powstania i uformowania gier *MOBA* była *Defense of the Ancients* (w skrócie *Dota* lub *DotA*), modyfikacja do gier *Warcraft III: Reign of Chaos* i *Warcraft III: The Frozen Throne*. *Dota*, oparta na *Aeon of Strife*, była jednym z pierwszych głównych tytułów tego gatunku i pierwszym przedstawicielem gier *MOBA*, dla którego przeprowadzono sponsorowane turnieje.[26]



Rysunek 8. Ekran ładowania gry *DotA*

Jak już zostało powiedziane w poprzednim rozdziale gry *F2P* to takie gry, do których za dostęp użytkownik nie musi zapłacić. Należą do nich gry przeglądarkowe, gry mobilne, istnieją ich odpowiedniki na konsole, na urządzenia mobilne jak i na *PC*. W styczniu 2018 roku ukazał się raport firmy *SuperData Research* ukazujący udział danych tytułów gier w rynku gier *F2P* (*free-to-play*) w roku 2017. O tym raporcie możemy przeczytać w artykule Tannera Dedmona z portalu *comicbook.com*, w którym autor pokazuje jak bardzo dominującą produkcją na przestrzeni ostatnich lat stała się *League of Legends*. [6] W artykule tym są również odnośniki

do całego raportu jak i jego głębszej analizy, natomiast w tej pracy zostanie przedstawiona tylko kwestia związana ze zwycięską produkcją i jej gatunkiem.

Najpierw trzeba sobie powiedzieć czym jest gra typu *MOBA*. Cytując ze słownika gracza na stronie *gryonline.pl* jest to:

Odmiana strategii czasu rzeczywistego przeznaczona do rozgrywki wieloosobowej za pośrednictwem sieci Internet. Gry rodzaju MOBA (Multiplayer Online Battle Arena) nastawione są na dynamikę starć kosztem rezygnacji z takich elementów RTS jak budowanie i rozwijanie własnej bazy. Podstawę rozgrywki w grach MOBA stanowi rywalizacja dwóch kilkusobowych zespołów na ograniczonej wielkości planszach. Każdy z graczy kontroluje poczynania jednego bohatera i pełni określoną rolę w drużynie. Zwycięstwo zależy od wykorzystania indywidualnych umiejętności oraz dobrej współpracy w grupie.[7]

Warto dodać do powyższego ważną rzecz. Gry tego podgatunku charakteryzuje podobieństwo do szachów. Są to szachy wieloosobowe. Na poparcie tej tezy przychodzi firma stojąca za *League of Legends*, która zresztą sama stworzyła ów termin gier typu *MOBA*, *Riot Games*. System rankingowy ich gry oparty jest na rankingu *Elo*[8], który wykorzystuje się w szachach. Gracze szachowi w rozgrywce muszą chronić swojego króla jednocześnie starając się matować przeciwnego, mają do tego zasoby w postaci pionów i figur szachowych. Tak samo jest w grach typu *MOBA*, gracze bronią swojej bazy współpracując i wykorzystując zasoby jakimi są złoto, przedmioty lub czary. I jedni i drudzy stosują konkretne strategie, dostosowując je wraz z upływem czasu rozgrywki próbując przechytrzyć rywala na wszelkie możliwe sposoby.

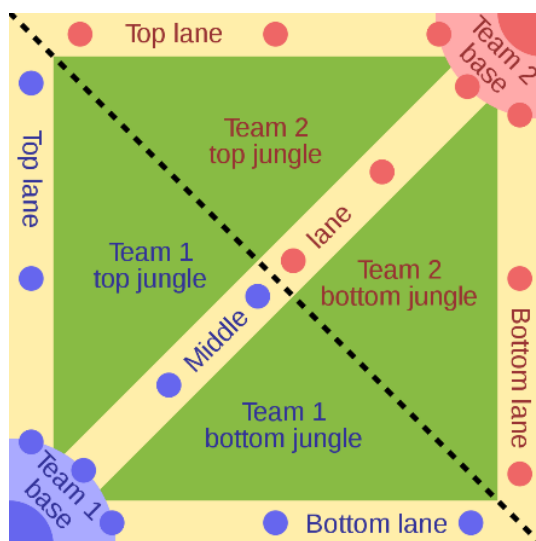


Rysunek 9. Logo firmy Riot Games

Co się tyczy rozgrywki w omawianym typie gier to już nieco zostało na ten temat powiedziane jednak w celu uściślenia przebiegu walki w grach typu *MOBA* w kilku następnych akapitach przedstawiona została charakterystyka gry.

Istnieją dwie drużyny przeciwne sobie, których celem jako zespołu by wygrać jest zniszczenie bazy wroga. Zdarza się, że niektóre gry lub modyfikacje mają inne warunki zwycięstwa jednak zazwyczaj jest to odstępstwo od reguły. Najczęściej każdy zespół składa się z pięciu graczy i istnieje jedna główna struktura, którą należy zniszczyć by wygrać mecz. Zniszczenie innych struktur i obiektów przynosi zazwyczaj inne korzyści, najczęściej jest to dodatkowe złoto lub wzmocnienie dla konkretnego gracza i/lub całej jego drużyny. Struktury obronne wyposażone są zawsze w mechanizm strażniczy, w postaci działek ochronnych. Jeśli gracz podejdzie zbyt blisko to te go zaatakują. Dodatkowym mechanizmem obronnym głównej budowli są małe, stanowiące niewielkie zagrożenie i będące głównie mięsem armatnim, stworki. Są one sterowane komputerowo i okresowo pojawiają się w każdej bazie, podróżując w dół wcześniej zdefiniowanymi ścieżkami (potocznie zwanymi *liniami*) w kierunku bazy przeciwnika. Zazwyczaj istnieją trzy linie będące głównym sposobem dojścia do bazy przeciwnego zespołu a wszystko pomiędzy tymi pasami będące niezbadanym

i zazwyczaj ukrytym terenem nazywane jest dżunglą (potocznie nazywane *lasem*). Gracz kontroluje jedną potężną jednostkę zwaną *bohaterem*. [27]



Rysunek 10. Obrazek poglądowy ukazujący jak wygląda typowa mapa gry MOBA

Bohater stojący w pobliżu zabitej jednostki wroga lub ją zabijający zdobywa punkty doświadczenia pozwalające mu na awans w poziomach. Zyskując poziomy bohaterowie mają możliwość wzmocnienia swoich umiejętności, których zazwyczaj jest cztery. Gdy bohater umiera, musi odczekać pewien okres czasu, z reguły czas ten rośnie wraz ze zdobywanym przez niego poziomem. Po odczekaniu danego czasu gracz odradza się na terenie bazy swojego zespołu.[27]

Bohaterowie należą do jednej z wielu dostępnych w grze ról, których liczba i typ może się różnić w zależności od konkretnego tytułu. Wyróżnia się postacie odpowiadające za ochronę, wsparcie, leczenie i zadawanie obrażeń wrogom. Te ostatnie można podzielić jeszcze ze względu na typ i rodzaj zadawanych obrażeń jak np. zabójca, strzelec czy mag. Do tego postacie grające w dżungli nazywane są *dżunglerami* lub *leśnikami*. Każdy bohater jest wyjątkowy, ma własne unikalne zdolności, których nie dzieli z żadną inną postacią, nawet jeśli ta dzieli z nim rolę. W każdej grze istnieje duża, stale rozwijana pula bohaterów. Przykładem tego jest gra *League of Legends*, w której na starcie gracz dysponował 40 postaciami, w marcu 2019 roku ta liczba wynosiła już 143 jednostki. Jest to czynnik utrudniający rozgrywkę, gracz musi nauczyć się nie tylko celów i strategii gry ale również nauczyć się grać przynajmniej jedną postacią i dodatkowo znać możliwości wszystkich pozostałych bohaterów w grze.[27]

Jakby tego było mało to dodatkowo gracz musi również mieć na uwadze dobro zespołu w postaci bardzo ważnego czynnika, którym jest użyteczność danej postaci w zależności od pozostałych, już wcześniej wybranych przez innych graczy, bohaterów. Grupę wybranych przez wszystkich graczy postaci nazywa się *kompozycją* i wielokrotnie zdarza się, że przy równorzędnych umiejętnościach wszystkich graczy to właśnie kompozycja drużyny A i drużyny B odgrywa kluczową rolę. Dodatkowo każdy bohater jest celowo ograniczony w swoich rolach. Żaden bohater nie może być na tyle silny, by wygrać grę w pojedynkę. Stwarza to silny nacisk na pracę zespołową i współpracę pomiędzy graczami.[27]

Tabela poglądowa objaśniająca podział ról jakimi mogą dysponować postacie w grach MOBA:

Rola	Opis postaci
Obrońca	Duża ilość punktów zdrowia, nastawione na ochronę sojuszników, posiadające czary umożliwiające rozpoczęcie walki, obdarzone stosunkowo małą ilością punktów obrażeń.
Wspierający	Obdarzone czarami ułatwiającymi grę pozostałym sojusznikom np. poprzez leczenie lub unieruchomienie wroga, same stanowią dla niego małe zagrożenie.
Dżungler	Obdarzona przeżywalnością w lesie, posiadająca czary umożliwiające rozpoczęcie walki. Zbalansowana ilość punktów obrażeń i zdrowia.
Strzelec	Dystansowe, skupione na zadawaniu obrażeń fizycznych, duża liczba obrażeń fizycznych, mała ilość zdrowia.
Mag	Zadające obrażenia magiczne, największa ilość obrażeń magicznych, mała ilość zdrowia.

Tabela 4. Role postaci z opisem.

Każdy gracz otrzymuje niewielką ilość złota na sekundę w trakcie gry. Do tego złoto można zdobyć za zabicie wcześniej już wspomnianych wrogich stworków i struktur, a większe kwoty są dostępne za zabicie wrogich bohaterów. Złoto jest używane przez bohaterów do kupowania różnych przedmiotów o różnej cenie i wpływie na postać, przeciwnika lub rozgrywkę. W dużej mierze wiąże się to z poprawą zdolności bohatera, mogą istnieć też inne elementy, które wspierają bohatera lub cały jego zespół na różne sposoby, chociażby zwiększając przychód złota na minutę lub dając dodatkowy pancerz. [27]

Gdy bohaterowie wraz z poziomami i zakupionymi przedmiotami stają się silniejsi, mogą skorzystać z wielu strategii by zyskać przewagę w grze. Jak już wcześniej zostało wspomniane, należy brać pod uwagę kompozycję danego zespołu gdyż nie każda strategia znajdzie swoje zastosowanie w każdej, ułożonej przed grą, kompozycji. Strategie te obejmują zabezpieczanie celów na mapie jakim są struktury, neutralne potwory w dżungli, zabijanie wrogich bohaterów czy też pasywne zdobywanie przewagi w złocie przez zabijanie wrogich stworków. Im silniejsza jest drużyna, tym bardziej jest ona zdolna do niszczenia wroga i jego bazy.[27]

Wracając do raportu ze strony *comicbook.com*, widać w nim zarobki *League* za rok 2017 przekraczające 2 mld USD. Skoro cały rynek gier na PC to 33 mld USD oraz skoro *League* jest grą darmową to widać jaki sukces odniosła ta produkcja i jaki ma to wpływ na rynek. Należy dodać, że na miejscu piątym owego rankingu widnieje konkurencyjna gra *MOBA* od firmy *Valve*, jest nią *DOTA 2* z przychodami powyżej 400 mln USD. [6]

Dodając do siebie wszystkie produkcje z rankingu 10 najbardziej dochodowych gier *F2P* w ramach swoich gatunków i porównując z powyższymi 2,5 mld USD widać, że i tak daleko im jeszcze do gatunku jakim jest *MOBA*.

Może się wydać intrygującym fakt, że skoro *League* jest grą darmowa to jak dokładnie zarabiają na niej jej twórcy. Tę sprawę również można znaleźć w artykule Tannera Dedmona, który przedstawia jak gra spod szyldu *Riot Games* zdobyła te 2 mld USD:

Skórki bohaterów, skórki totemów, obrazki profilowe graczy znane jako Ikony Przywoływaczy i najnowsze emotikony są dodatkami wizualnymi, przedmiotami, które mogą

zostać zakupione w sklepie w kliencie gry, niemniej żaden z powyższych przedmiotów nie wpływa na przebieg rozgrywki. Również postacie mogą być zakupione przy pomocy prawdziwej gotówki, wszystko dzięki systemowi waluty premium w grze znanej jako *Riot Points*.^[6]

Jak widać powyżej, gracze kupują zawartość gry, którą lubią i nad którą spędzają setki godzin rocznie. *LoL* ma już 10 lat i dalej utrzymuje się w czołówce wszystkich rankingów traktujących o *esporcie*, grach *F2P* czy *MOBA*. Dodatkowo widać jak bardzo dochodowym zajęciem może być stworzenie tego typu gry.



Rysunek 11. Kadr z trybu dla oglądających z gry *League of Legends* ukazujący walkę drużynową trzech na trzech na górnej alei

Lista wybranych gier MOBA wraz z platformami, na które są dostępne:

Tytuł	Wydawca	Data wydania	Platforma	Free-to-play	Dostępność
Defense of the Ancients	Eul, Guinsoo, IceFrog	2003	Windows, Mac	Modyfikacja do gry płatnej	Dalej grywalna
League of Legends	Riot Games	2009	Windows, Mac	Tak	Dalej grywalna
Guardians of Middle-earth	Monolith Production	2012	Xbox 360, PS3, Windows	Nie	Dalej grywalna
Warhammer Online: Wrath of Heroes	Mythic Entertainment	2012	Windows	Tak	Niedostępna od 29 marca 2013
Dota 2	Valve Corporation	2013	Windows, Mac, Linux	Tak	Dalej grywalna
Smite	Hi-Rez Studios	2014	Windows, Xbox One, PS4	Tak	Dalej grywalna
Heroes of the Storm	Blizzard Entertainment	2015	Windows, Mac	Tak	Dalej grywalna
Arena of Valor	Timi Studio Group	2015	iOS, Android, Nintendo Switch	Tak	Dalej grywalna
Warhammer 40000: Dark Nexus Arena	Whitebox Interactive	2016	Windows	Tak	Niedostępna od 31 marca 2016
Star Wars: Force Arena	Netmarble	2017	Windows	Tak	Niedostępna od 18 marca 2019

Najbardziej znane tytuły *MOBA* ostatnich lat.

Matchmaking

Sesje sieciowych gier wieloosobowych mogą być aranżowane na wiele sposobów. Do najbardziej popularnych z nich zalicza się zebranie grupy znajomych w lobby gry, na jej serwerze lub, jeśli jest ich więcej, wybranie serwera z listy i dołączenie do już trwającej rozgrywki. Problem zaczyna się w momencie kiedy gracz jest sam i chce dołączyć do rozgrywki wieloosobowej po sieci. Jak wtedy przebiega dobranie sojuszników i przeciwników?

Dobłą opcją umożliwiającą połączenie z grą jest matchmaking. Jest to proces łączenia graczy ze sobą w rozgrywkę sieciową przy założeniu jak najmniejszej roli graczy, dodawanych do niej w trakcie procesu. Gracze indywidualni i/lub całe zespoły, gdyż matchmaking tego nie wyklucza, klikają przycisk *wyszukaj grę* i są parowani przez system z innymi podobnymi jednostkami chętnymi do gry. Istnieje kilka opcji matchmakingu. Najprostsze z nich to lista chętnych do gry – gracze dołączają do lobby dopóki nie zbierze się określona liczba chętnych lub nie minie czas oczekiwania; wcześniej wspomniana drużyna – gracze formują się w małą grupkę i walczą przeciwko innej małej grupce graczy; najczęściej jednak stosuje się rozgrywkę w oparciu o konkretny mechanizm rankingowy gry. Przy wszystkich wcześniej wymienionych opcjach twórcy gier zazwyczaj dołączają opcje listy kontaktów w celu zaproszenia do gry swoich znajomych oraz opcję czatu lub czatu głosowego ułatwiającą komunikację pomiędzy członkami w zespole.[29]

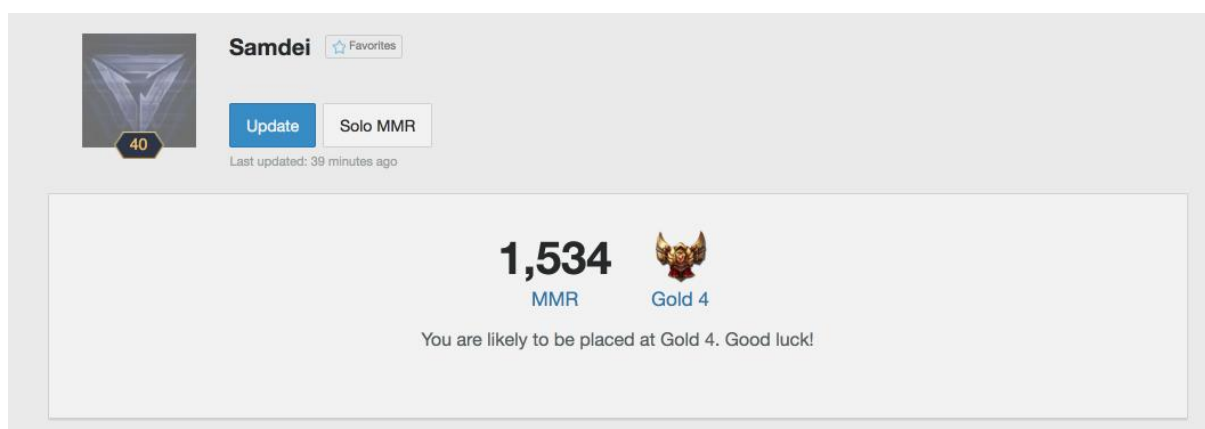
W momencie gdy znaleziona zostanie konkretna, wcześniej ustalona liczba graczy, tworzona jest nowa jednostka gry zwana meczem. W przypadku gier typu *MOBA* i w zależności od tytułu zazwyczaj w tym momencie, po znalezieniu meczu, gracze przechodzą do menu wybierania bohaterów. W innych typach gier proces wyboru postaci może przebiegać przed matchmakingiem. Po zakończeniu tego etapu mecz się rozpoczyna.

Główną korzyścią wynikającą z matchmakingu jest zwiększona konkurencyjność rozgrywek, możliwość zebrania całej drużyny i gra na inną, kompletną drużynę na danym poziomie umiejętności oraz różnorodność konfiguracji opcji wyszukiwania takich jak różne rodzaje map w grze, różne role, różne rangi i poziomy wśród graczy.

Wadą matchmakingu może być długi czas wyszukiwania gry w momencie braku konkretnego typu szukanych na serwerze graczy. Przykładem tego jest sytuacja graczy z dywizji rankingowej o nazwie *challenger* we wcześniej wspomnianej *League of Legends* i późna pora nocna wyszukiwania meczu. Z racji tego, że *challenger* skupia w swoich szeregach tylko najlepszych dwustu graczy na serwerze to późna pora wyszukiwania meczu może spowodować wielominutowy czas oczekiwania na grę.

Jak już to zostało przedstawione powyżej, matchmaking działa na zasadzie zacieśniania kręgu szukanych graczy. Wiele systemów wyszukiwujących mecze działa w oparciu o estymację umiejętności graczy na podstawie ich rang w grze. Pozwala to graczom o zbliżonych umiejętnościach na grę przeciwko sobie, ograniczając przypadki gdy gracz słaby trafi na zawodowego gracza e-sportowego. Wpływa to na jakość przeprowadzanych gier, zwiększenie radości czerpanej z gry przez jej użytkowników, a zwłaszcza zwiększa konkurencyjność wśród graczy dzięki czemu ci stają się lepsi. Dzięki temu mogą awansować w swojej randze a dzięki temu matchmaker następnym razem może przyporządkować im lepszych graczy, na ich nowym poziomie umiejętności.

Pomimo tego co już zostało przedstawione należy pamiętać, że sam proces matchmakingu rzadko kiedy jest perfekcyjny. Matchmaking nie zmusza graczy do wykonywania wielu kroków w celu znalezienia gry jednak jest od nich w pełni uzależniony. Przykładowo omówiony zostanie matchmaker wielokrotnie już przywołanej na łamach tej pracy gry *League of Legends*. Twórcy gry wprowadzili w swojej produkcji ukryty przed graczem ranking znajdowania gier o nazwie *MMR* – *Matchmaking Ratio*. Ważnym jest by nie mylić tego rankingu z rangą gracza w grze. W przypadku *LoL* *MMR* jest dynamiczniejszym rankingiem przez to, że bazuje na ostatnich kilku grach. Liga i dywizja gracza czyli jego ranga w grze często są wypadkową kilkudziesięciu lub nawet kilkuset gier w sezonie, do tego gracze stosują różne sztuczki w celu podniesienia swojej rangi. *MMR* w kontekście wyszukiwania meczu jest bardziej wiarygodną opcją określenia poziomu umiejętności gracza dla systemu. Gdy gracz wygrywa *MMR* rośnie, gdy gracz przegrywa *MMR* spada.[9]



Rysunek 12. Istnieje wiele stron internetowych, na których gracze mogą próbować podejrzeć swój MMR. Jedną z najbardziej popularnych jest strona op.gg

Przy doborze przeciwników system uwzględni też poziom przeciwników. Szukając odpowiedniej gry, system będzie próbował znaleźć graczy o podobnym *MMR*. Ważną rzeczą do zapamiętania jest też to, że jeśli w grze jak w *LoL* jest więcej niż jedna kolejka rankingowa to *MMR* również będzie tyle ile trybów gry.[9]

Gdy gracze czekają w kolejce aż system dobierze im przeciwników, mechanizm znajdowania gier tworzy ją usiłując zrównoważyć trzy czynniki, o których można przeczytać na stronie wsparcia firmy *Riot Games*. Tymi czynnikami są:

- *Sprawiedliwość meczów* – obie drużyny są zbliżone poziomem umiejętności,
- *Preferencje dotyczące pozycji* – grasz na pozycji, na której chcesz,
- *Krótki czas oczekiwania* – gracze czekają w kolejkach jak najkrócej.

Idealny mecz to taki, w którym wszystkie powyższe aspekty są równorzędnie traktowane przez system a szansa na zwycięstwo obydwu drużyn wynosi między 49 a 51%. W idealnej rozgrywce 10 graczy o identycznym *MMR* stanęłoby naprzeciwko siebie jednocześnie a każda z tych osób wybrałaby niepowtarzalną pozycję, do której się nadaje.[9]

W *League of Legends* gracze wchodząc do lobby przed matchmakingiem wybierają dwie pozycje, główną i drugorzędną. Główna rola zostanie potraktowana jako rola priorytetowa a druga zostanie zakwalifikowana jako pozycja awaryjna. Jeśli system gry nie będzie w stanie skompletować idealnej drużyny na danym poziomie umiejętności graczy w oparciu o wybrane

pozycje istnieje szansa, że system wyrzuci pewnego gracza do zespołu automatycznie przyporządkowując mu inną rolę nawet jeśli nie jest ona zgodna z preferencjami tej konkretnej jednostki. Zazwyczaj dzieje się tak, gdy czas oczekiwania na grę jest zbyt długi.[9]

Jak więc widać matchmaking nie jest sposobem banalnym do zrealizowania. Poniżej krótka charakterystyka i podsumowanie każdego problemu. Idealny scenariusz zakłada 10 graczy rozgrywających perfekcyjnie zrównoważony mecz. Są jednak czynniki, które należy wziąć pod uwagę:

- mecze rozgrywane w porach dnia o niskiej aktywności graczy,
- zebrane grupy graczy o różniących się od siebie współczynnikach *MMR*,
- brak chętnych graczy na co najmniej jedną pozycję/rolę w grze.[9]

Dzięki powyższemu system musi optymalizować wyszukiwanie graczy do rozgrywki traktując jeden z aspektów wyszukiwania z większą wagą niż inny. Efektem tego może być sytuacja, w której idealny mecz się nie odbędzie przez to, że:

- przynajmniej jeden z graczy zacznie grę na niewłaściwej lub nieznannej dla siebie pozycji,
- mecz, gdzie skład rywali z linii będzie nieuczciwy – np. gracz nie grający nigdy na górnej alei zostanie na nią rzucony na gracza, który gra tylko na niej,
- gry mogą wymykać się spod kontroli ze względu na wybrane postacie – gracz, który teoretycznie na danej roli spisywałby się lepiej podejmie bohatera, którym zagra pierwszy raz w życiu i przegra na słabszego mechanicznie gracza ale z postacią, którą ten opanował do perfekcji,
- słabszy mecz lub dzień, któregoś z graczy.[9]

Liga Legend

Wielokrotnie już wspomniana na łamach tej pracy dyplomowej gra *League of Legends* to idealny przykład sukcesu swojego gatunku. Darmowa sieciowa gra wieloosobowa, przyciągająca miesięcznie 100 milionów graczy z całego świata, zarabiająca rocznie ok. 2 mld USD z tytułu mikropłatności.

Wszystko zaczęło się od pomysłu na grę dwóch kolegów z akademickiego pokoju Uniwersytetu Południowej Kalifornii, Brandona Ryze'a Becka i Marca Tryndamere'a Merrilla. Wraz ze Stephenem Guinsoo Feakiem, projektantem mapy do modyfikacji Warcraft III o nazwie *DotA* oraz Stephenem Pendragon'em Mesconem, który odpowiadał za pion wsparcia wspomnianej modyfikacji, w 2006 roku założyli firmę *Riot Games*.^[1]



Rysunek 13. Logo gry League of Legends

O pracach nad grą *League of Legends* po raz pierwszy poinformowano opinię publiczną 7 października 2008 roku a zamknięte beta testy gry trwały od 10 kwietnia do 22 października 2009 roku. Po tej ostatniej dacie *LoL* trafił do otwartej bety aż do momentu wydania 5 dni później. *Riot Games* samodzielnie publikuje i obsługuje grę wraz ze wsparciem swoich klientów. Obecnie *Riot Games* dysponuje 24 biurami na całym świecie, w których pracuje ponad 1700 pracowników. Wszelkiego rodzaju analitycy, animatorzy, artyści 3D i artyści koncepcyjni, osoby odpowiedzialne za tworzenie oprogramowania, dyrektorzy artystyczni, projektanci gry i systemów w niej, pracownicy działu audio, menadżerowie marki i zawartości gry. To tylko niektóre ze stanowisk jakie obecnie oferuje firma rozwijająca i utrzymująca tylko jedną grę. Wszystko po to by zapewnić graczom z całego świata rozrywkę na jak najwyższym poziomie. Warto dodać w tym miejscu, że gra wyszła na platformę Microsoft Windows, dopiero w marcu 2013 roku *Riot Games* wypuściło wersję klienta beta na OS X.^[1]

Oceny *League of Legends* wg czołowych portali opiniotwórczych o tematyce gier:

Oceniający	Ocena
Metacritic	78/100
1UP.com	A-
Eurogamer	8/10
Game Revolution	B+
GameSpy	4/5
GameZone	9/10
IGN	9.2/10

Tabela 5. Oceny LoL

Początkowo portal IGN dał *LoL* ocenę 8/10 podkreślając przyjemny w odbiorze projekt gry, pomysły projekt bohaterów, ich dostosowanie do świata gry oraz efekty wizualne. Na minus eksperci ocenili mylący graczy i opinię publiczną sposób wydania gry. Krytycy ocenili, że gra została wydana zbyt wcześnie i wciąż brakuje w niej kilku funkcjonalności. Na koniec recenzenci ocenili, że gracze obeznani już z mechaniką gry mają małą cierpliwość do nowych graczy z zastrzeżeniem, że system matchmakingu, który w dniu recenzji nie był jeszcze wprowadzony, rozwiązał by sprawę. W 2014 roku portal ponownie ocenił produkcję *Riot Games* zmieniając ocenę z 8.0 na 9.2 (w dziesięciostopniowej skali) stawiając ją za przykład doskonałości, chwalać za różnorodność postaci, nagradzając system rankingowy i intensywność rozgrywek strategicznych o charakterze zespołowym.[1]

Dodatkowym czynnikiem wpływającym na popularyzację omawianego tytułu i zadowolenie graczy jest dział jego promocji. Twórcy gry dbają o to by gracz się nie nudził regularnie wypuszczając materiały dodatkowe o tematyce *LoL* takie jak piosenki, w których podmiotami lirycznymi są postacie z gry, zapowiedzi postaci i wydarzeń w formie filmów animowanych, zwiastuny nowych postaci. Prawdziwą perłą w koronie działu promocji są produkcje z serii *League of Legends Cinematic*. W krótkich, kilkuminutowych, dynamicznych filmikach twórcy ukazują bohaterów swojej gry walczących ze sobą nawzajem. Czasem samodzielnie, jeden na jednego a czasem w kilkusobowych drużynach. Produkcje tego typu wpływając na wyobraźnię graczy, cementują ich przywiązanie do gry.

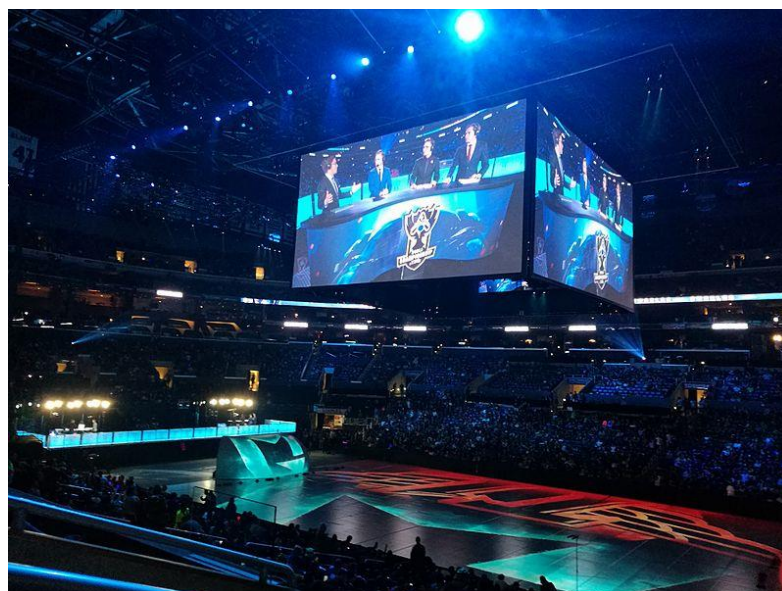
Gra w dalszym ciągu prężnie się rozwija, twórcy udoskonalają i balansują rozgrywkę poprzez uaktualnienia 2 razy każdego miesiąca. Regularnie wypuszczają nowych bohaterów, uaktualniają te stare i zapomniane oraz organizują wydarzenia dla graczy często wspierane przez topowe firmy z branży gier i IT.[31]

Kluczowym aspektem jeśli chodzi o grę jest e-sport. Na całym świecie od wielu lat trwa dyskusja na temat tego czy gry komputerowe to pełnoprawny sport. Dnia 15 kwietnia 2013 roku *League of Legends* oficjalnie została uznana w Stanach Zjednoczonych za pełnoprawny sport dając argument sprzymierzeńcom sportów elektronicznych w prowadzonej dyskusji. *League of Legends* jest jedną z największych gier e-sportowych na świecie. Kanał gry w serwisie twitch.tv jest pierwszym tego typu kanałem, który dobił do 1 miliarda wyświetleń. Na całym świecie co roku odbywają się liczne turnieje sponsorowane przez *Riot Games*. Szacuje się, że na nagrody we wszystkich oficjalnych turniejach twórcy marki przeznaczyli już blisko 30 mln USD. Istnieją również profesjonalne ligi e-sportowe, w których najlepsze drużyny rywalizują ze sobą co tydzień na wzór rozgrywek piłkarskich. Najlepsze z nich zrzeszone są pod szyldem *LolSports.com*. Zwycięzcy najsilniejszych lig wraz z innymi drużynami, które uzyskają prawo uczestnictwa biorą udział w jednym z największych wydarzeń e-sportowych roku, Mistrzostwach Świata w *League of Legends*. O prestiżu rozgrywek świadczy fakt, iż 6 grudnia 2018 roku MŚ w *LoL* zostały uznane **Najlepszym Wydarzeniem E-sportowym (Best Esport Event)** na gali *The Game Awards*. [32]

E-sport

E-sport (ang. Esport) również znane pod nazwą sportów elektronicznych, forma rywalizacji przy użyciu gier komputerowych. Rywalizacja odbywa się w formie rekreacyjnej i zarobkowej. W większości przypadków e-sport przyjmuje formę zorganizowanych zawodów/turniejów gier komputerowych lub konsolowych między graczami i ich zespołami.

Mecze rozgrywane są na żywo i streamowane w portalach *Youtube*, *Twitch* czy *Hitbox* oraz przez telewizję np. *ESPN*. W Polsce przetarg na pokazywanie rozgrywek **LEC** (**League European Championship**) wygrała stacja *Polsat*.^[33] Najnowszymi produkcjami zyskującymi popularność w świecie e-sportowym są gry *Fortnite* i *PUBG*. Największa pula nagród na jednym turnieju e-sportowym należy do gry *Dota2* i jej turnieju *The International 2018* i wynosi 25,5 mln USD. W przypadku Polski najwyżej w rankingu puli nagród, *ex aequo* z 35 innymi wydarzeniami, na 43 miejscu są *ESL One Katowice 2018 (Dota2)* i *PGL Major Kraków 2017 (CS:GO)* z pulą nagród 1 mln USD każdy.^[34]



Rysunek 14. League of Legends World Championship 2016, Staples Center, Los Angeles, USA

Początki

Pierwszym znanym zorganizowanym wydarzeniem e-sportowym w historii można uznać turniej w *Spacewar* na Uniwersytecie Stanforda, dnia 19 października 1972 roku. Główną nagrodą był roczny abonament gazety *Rolling Stone*, zwycięzcą został Bruce Baumgart. Mistrzostwa w *Space Invaders* zorganizowane przez firmę *Atari* w 1980 roku były pierwszymi rozgrywkami gier wideo na dużą skalę. Przyciągnęły ponad dziesięć tysięcy uczestników w Stanach Zjednoczonych, ustanawiając współzawodnictwo w grach wideo jako nowe hobby wielu młodych Amerykanów.^[34]

Latem 1980 roku Walter Day założył organizację zajmującą się księgowaniem rekordów w grach, nosiła ona nazwę *Twin Galaxies*. Organizacja pomogła w promocji gier i nagłaśniała swoje zapisy poprzez ich publikacje a w 1983 roku utworzyła *Krajowy zespół ds. gier wideo w Stanach Zjednoczonych*. Zespół ten pomagał w licznych turniejach, takich jak *Video Game Masters Tour* dla *Księgi rekordów Guinnessa* oraz sponsorował m.in. *North American Video Game Challenge*.^[34]

W latach 1982 - 1984 w amerykańskiej telewizji emitowany był program *Starcade*. Wyemitowano 133 odcinki, podczas których gracze rywalizowali między sobą bijąc wzajemnie swoje rekordy w grach zręcznościowych. Kolejnym krokiem w kierunku popularyzacji rywalizacji poprzez gry było dołączenie do scenariusza programu *That's Incredible!* rozgrywek gier wideo. Gry nabierały na znaczeniu i wkrótce dostały się również na duży ekran, przykładem tego jest film *Tron* Stevena Lisberga z 1982 roku.[34]

Gra walki *Street Fighter II* z 1991 roku spopularyzowała koncepcję bezpośredniej, turniejowej rywalizacji między graczami. Wcześniej gry wideo najczęściej opierały się na wysokich wynikach w celu określenia zwycięzcy. Zmieniło się to w *Street Fighter II*, gdzie gracze zamiast tego rzucali sobie wyzwanie bezpośrednio by określić lepszego z nich. Popularność gier walki takich jak *Street Fighter* w latach 90. doprowadziła do powstania w 1996 roku międzynarodowego turnieju e-sportowego *EVO (Evolution Championship Series)*. [34]

Duże turnieje e-sportowe w latach 90. obejmowały Mistrzostwa Świata na konsoli *Nintendo* w roku 1990 wraz z finałami w *Universal Studios Hollywood* w Kalifornii. W 1994 roku *Nintendo* ponownie zorganizowała mistrzostwa o nazwie *Nintendo PowerFest '94* na konsolę *Super Nintendo Entertainment System*. W finale w San Diego brało udział 132 graczy. E-sportowymi programami TV w tym okresie były m.in. brytyjskie programy *GamesMaster* i *Bad Influence!*, australijski teleturniej *A*mazing* i kanadyjski teleturniej *Video&Arcade Top 10*. Wiele gier zyskało na pojawieniu się sieci *Internet*. Turnieje organizowane pod koniec lat 90 to między innymi *Cyberathlete Professional League* (gry z serii *CS*, *Quake* i *Warcraft*), *QuakeCon* czy też *Professional Gamers League*. [34]

Turnieje rangi światowej i profesjonalne ligi

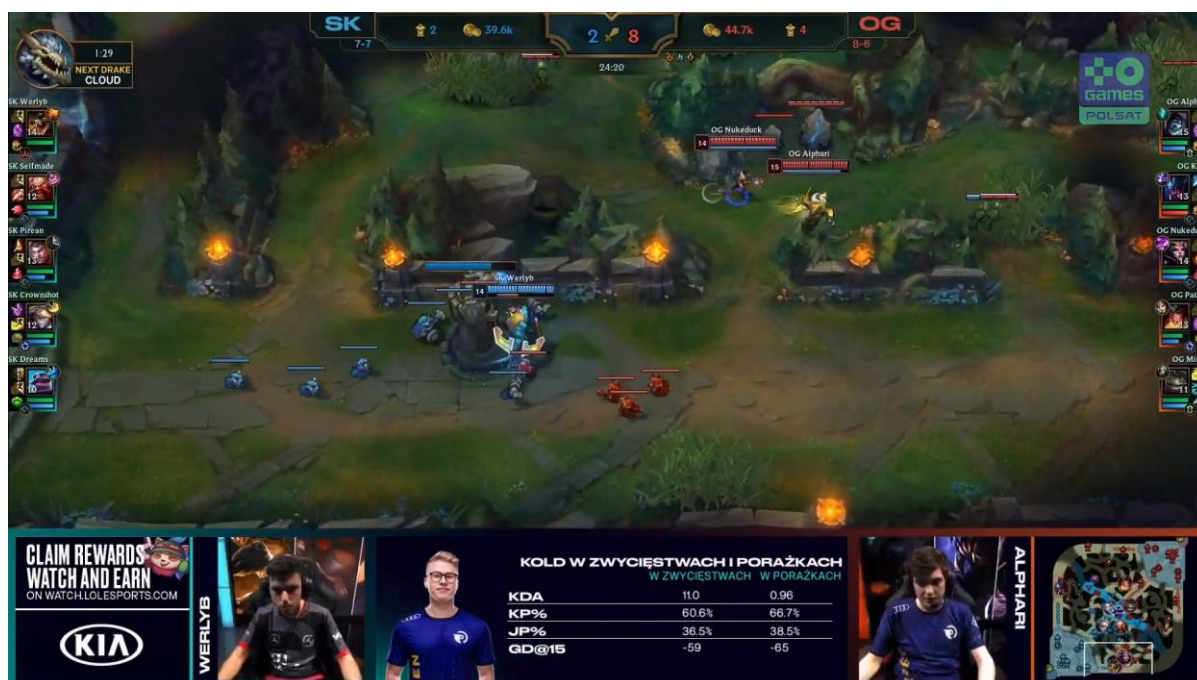
W drugiej dekadzie XXI wieku e-sport ogromnie się rozwinął, powodując duży wzrost oglądalności oraz wzrost nagród pieniężnych. Mimo, że pod koniec ubiegłego wieku odbywały się duże turnieje to po roku 2000 znacznie wzrosła ich ilość. Od około 10 turniejów w 2000 roku do około 260 w 2010 roku. W tym okresie powstało wiele znanych rozgrywek jak *World Cyber Games (WCG)*, *Intel Extreme Masters (IEM)* z wielkim finałem w Katowicach i *Major League Gaming (MLG)*. [34]

Po roku 2000 nastał dobry czas dla telewizji e-sportowych. Największy zasięg miały one w Korei Południowej, zawody w *StarCraft* i *Warcraft III* były regularnie transmitowane przez dedykowane 24-godzinne kanały telewizji kablowej *Ongamenet* i *MBCGame*. We Francji e-sportem zaczął zajmować się kanał *GameOne* a w Stanach *ESPN*. Ci ostatni nie byli jedyną telewizją, która zdecydowała się na ten krok. Nie minęło dużo czasu a do stawki w USA dołączyły telewizje *DirecTV* i *CBS*. [34]

Wraz z rozwojem *Internetu* nastał jednak czas streamingu. Pojawienie się usług streamingowych i ich popularyzacja pomogły w rozwoju e-sportu w tym okresie. Platformy te głównie były i są wykorzystywane w celu transmisji/oglądania transmisji wydarzeń e-sportowych. *Twitch.tv*, internetowa platforma streamingowa uruchomiona w 2011 roku, regularnie transmituje popularne zawody e-sportowe w ponad 100 grach. W 2013 roku widzowie platformy obejrzeli 200 milionów godzin zmagani profesjonalnych graczy, a dwiema najpopularniejszymi grami na platformie były *League of Legends* i *Dota 2*. Podczas jednego

dnia turnieju *The International Twitch.tv* odnotowała 4,5 miliona unikalnych wyświetleń, a każdy z widzów oglądał transmisję średnio dwie godziny. Wraz ze wzrostem liczby zawodów e-sportowych odnotowano też wzrost liczby firm produkujących gry komputerowe. Firmy te zaczęły korzystać z potencjału jaki posiada e-sport. Po wielu latach ignorowania sceny sportów elektronicznych, *Nintendo* gościło *Wii Games Summer 2010*. W ciągu miesiąca turniej zgromadził ponad 400 tysięcy uczestników co uczyniło go największym turniejem w historii firmy. W 2014 roku *Nintendo* zorganizowało na targach *Electronic Entertainment Expo (E3)* turniej *Super Smash Bros*. Na konsolę *Wii U*, który był transmitowany online w serwisie *Twitch.tv*. Tego samego roku twórcy gry *Halo* ogłosili plany wznowienia swojej produkcji jako e-sportu wraz z utworzeniem *Halo Championship Series* i pulą nagród w wysokości 50 tysięcy USD.[34]

Od 2013 roku uniwersytety i koledze w Stanach Zjednoczonych, takie jak *Columbia College*, *Robert Morris University Illinois* i *University of Pikeville* uznały zawodników sportów elektronicznych za sportowców klasy uniwersyteckiej oraz oferują im stypendia sportowe. W 2017 roku *Tespa*, oddział zajmujący się sportami elektronicznymi na uczelniach w USA w *Blizzard Entertainment*, zaprezentowała swoją nową inicjatywę przyznawania stypendiów i nagród dla najlepszych amatorskich klubów e-sportowych na uczelniach. Dzięki temu studenci amerykańskich uczelni rywalizują w turniejach o nagrody o wartości 1 mln USD. Fizyczna oglądalność zawodów e-sportowych i zakres wydarzeń zwiększył się wraz ze wzrostem oglądalności online. W 2013 roku Mistrzostwa Świata 3 sezonu *League of Legends* odbyły się we w pełni wyprzedanym *Staples Center* w Los Angeles. *Mistrzostwa Świata League of Legends 2014* w Korei Południowej zgromadziły ponad 40 tys. fanów na stadionie *Seul World Cup Stadium*, arenie na której odbywały się MŚ w piłce nożnej w 2002 roku. W 2017 roku gdy MŚ 7 sezonu *LoL* odbywały się w Chinach, fani zapełnili większość trybun Stadionu Narodowego w Pekinie (91 tysięcy miejsc).[34]



Rysunek 15. Statystyki gracza o pseudonimie KOLD w profesjonalnej lidze LEC transmitowanej przez polską stację TV – Polsat Games

Przy rosnącym zainteresowaniu e-sportem oraz wzrostem oglądalności niektóre firmy zaczęły dążyć do stworzenia lig, które byłyby wzorowane na normalnych dyscyplinach sportowych. Dzięki temu obecnie zespoły uczestniczą w normalnym sezonie ligowym, w którym rozgrywają mecze każdy z każdym a następnie te, które zajmą czołowe miejsca (1-4 lub 1-8) walczą w fazie play off.[34]

Blizzard Entertainment ogłosił utworzenie *Overwatch League* w oparciu o swoją grę *Overwatch*, w 2016 roku, obejmującym dwanaście drużyn, począwszy od stycznia 2018 roku. Gracze, którzy otrzymają kontrakt na grę w drużynie, mają zagwarantowane roczne wynagrodzenie i świadczenia socjalne, a także biorą udział w podziale dochodów zespołu z nagród. Pierwszy sezon *OL* miał łączną pulę nagród w wysokości 3,5 mln USD, z 1 mln dolarów przyznanych wygranej drużynie.[34]

Klasyfikacja jako sport

Kwalifikowanie gier jako sportu to kontrowersyjny punkt wielu dyskusji. Podczas gdy niektórzy wskazują na wzrost popularności e-sportów jako uzasadnienie wyznaczania niektórych gier jako sportowych, inni twierdzą, że gry wideo nigdy nie osiągną statusu prawdziwego sportu. Jednak popularność nie jest jedynym powodem identyfikowania e-sportu jako sport. Są ludzie, którzy twierdzą, że staranne planowanie i przygotowanie do zawodów oraz umiejętne wykonanie założeń taktycznych powinno być tym, co klasyfikuje aktywność jako sport[16].

W 2013 r. kanadyjski gracz *League of Legends*, Danny Shiphtur Le, został pierwszym zawodnikiem, który otrzymał amerykańską wizę *P-1A*. Kategorię przeznaczoną dla zawodników uznanych na arenie międzynarodowej.[35] W 2014 roku tureckie ministerstwo ds. młodzieży i sportu zaczęło wydawać licencje e-sportowców graczom, którzy zostali uznani przez nich za profesjonalistów. W 2016 roku francuski rząd rozpoczął prace nad projektem regulującym i uznającym e-sport jako sport.[34]

Aby promować e-sport jako legalny sport kilka wydarzeń e-sportowych odbywało się równoległe z tradycyjnymi międzynarodowymi zawodami sportowymi. *Igrzyska Azjatyckie* w 2007 roku były pierwszymi rozgrywkami wielosportowymi z włączeniem gier e-sportowych, jako oficjalne dyscypliny medalowe. Późniejsze edycje igrzysk również zawierały w swoim programie e-sport jako oficjalne wydarzenia medalowe lub imprezę wystawienniczą. Ponadto *Igrzyska Azjatyckie*, które są największą tego typu imprezą w regionie, będą również obejmować e-sport jako wydarzenia medalowe w 2022 roku. Zawodnicy z krajów azjatyckich będą rywalizować m.in. w *Hearthstone*, *Starcraft II* i *League of Legends*. Dodatkowo w ramach *Igrzyskach Azji Południowo-Wschodniej* w 2019 roku przewidziano sześć zestawów medali na konkurencje e-sportowe.[34]

Trening, który zawodnicy muszą przejść, by przygotować się do zawodów, jest podobny do tego sportowego. Wymaga poświęcenia ogromnej ilości czasu zawodnika i sztabu szkoleniowego. Ludzie tradycyjnie uprawiający sport opierają się na doskonaleniu swojej sprawności fizycznej. Ćwiczą pamięć mięśniową, wykonują setki ćwiczeń, stosują dietę. Treningi e-sportowców opierają się na treningu umysłu takim jak studiowanie strategii gry i nowych aktualizacji gry. Team Liquid, profesjonalna drużyna *LoL*, ćwiczy 50 godzin tygodniowo. Jednak jak uważa Pani Urszula Klimczak, psycholog pracująca dla najlepszych drużyn e-sportowych w Europie i Stanach Zjednoczonych, stereotyp e-sportowca mocno

zmienił się na przestrzeni ostatnich 2-3 lat. W wywiadzie udzielonym portalowi *sport.pl* mówi:

Kiedys mówiąc „profesjonalny gracz” ludzie widzieli nastolatka z nadwagą, który je fast foody, popija colą, a gdy wstaje około południa, to ma czerwone oczy. [...]Większość profesjonalistów na scenie ma pełną świadomość, jak istotna jest dieta i sen, oraz że ich przykład to wzór dla młodszych, którzy się wzorują na nich.[18]



Rysunek 16. Jarosław Pasha Jarząbkowski w czasach gry w zespole Virtus Pro

Widać zatem, że jeśli chodzi o trening e-sportowy to odbywa się on tak jak w sporcie tradycyjnym, na wielu płaszczyznach.

Jako dyscyplina olimpijska

Igrzyska Olimpijskie są również postrzegane jako potencjalna metoda legitymizacji sportów elektronicznych. Podczas zorganizowanego przez MKOL szczytu w październiku 2017 roku komitet olimpijski potwierdził rosnącą popularność e-sportów. Uznał konkurencyjność dyscyplin elektronicznych i stwierdził, że e-sport "można uznać za działalność sportową, zaangażowani w niego gracze przygotowują się i trenują z intensywnością porównywalną do sportowców uprawiających tradycyjne sporty".[17]

Pozostają dwie trudności z przedstawieniem e-sportu jako wydarzenia olimpijskiego, zgodnie z wypowiedziami prezydenta MKOL Thomasa Bacha:

- MKOL będzie zmuszone do ograniczania tych gier, które są zbyt brutalne,
- brak globalnego organu sankcjonującego e-sport w celu jego dalszej koordynacji.

W związku z powyższym zasugerowano by MKOL zatwierdził więcej e-sportów skupionych wokół gier symulujących prawdziwe sporty, takich jak *NBA 2K* lub seria *FIFA*. Nie przeszkodziło to jednak MKOL w przeprowadzeniu badań nad możliwością integracji sportów elektronicznych z przyszłymi *Igrzyskami Olimpijskimi*. [17]

W lipcu 2018 roku MKOL i Światowe Stowarzyszenie Międzynarodowych Federacji

Sportowych (GAISF) zorganizowały symposium i zaprosiły główne osobistości sportów elektronicznych by uzyskać głębsze zrozumienie e-sportu, ich wpływu i przyszłego rozwoju. Celem tego było wspólne rozważenie kwestii w jaki sposób obie strony mogą współpracować ze sobą by osiągnąć korzyści dla obu rodzajów sportu w nadchodzących latach.

MKOL przetestował potencjał e-sportu poprzez gry wystawowe. Dzięki wsparciu *MKOL Intel* sponsorował e-sportowe wydarzenia wystawowe dla *StarCraft II* i *Steep* przed *Zimową Olimpiadą 2018* w Pyeongchang, a pięciu południowokoreańskich graczy e-sportu było częścią sztafety olimpijskiej.[34]

Przywódcy Japonii angażują się w pomoc e-sportom by *te dostały się na Letnią Olimpiadę* w 2020 roku w Tokio. Komitet Olimpijski w Tokio zaplanował także zorganizowanie szeregu wydarzeń e-sportowych, które miałyby doprowadzić do gier w 2020 r.[41]

Komitet organizacyjny *Olimpiady Letniej* w 2024 roku w Paryżu dyskutował z *MKOL* i różnymi profesjonalnymi organizacjami e-sportowymi, aby rozważyć e-sport dla tego wydarzenia, powołując się na potrzebę uwzględnienia preferencji młodszych pokoleń. Ostatecznie komisja ds. organizacji uznała, że jeszcze za wcześnie by e-sporty były brane pod uwagę jako wydarzenie medalowe. Jednocześnie komisja nie wykluczyła innych działań związanych z e-sportami podczas igrzysk.[42]

Gry - przykłady

Obecnie jest wiele popularnych gier, w których pro-gracze zmagają się ze sobą. Turnieje które pojawiły się w połowie lat 90. zbiegły się z popularnością gier walki i strzelanek *FPS*, które nadal mają dużą rzeszę fanów. Najbardziej popularne gry z początku drugiej dekady XXI wieku to *CS:GO*, *Call of Duty*, *League of Legends*, *Super Smash Bros.*, *StarCraft II* i *Dota 2*. Dodatkowo gra *Hearthstone* spopularyzowała także cyfrowy gatunek kolekcjonerskich gier karcianych (*DCCG*) od czasu wydania w 2014 roku.

Niektóre gatunki gier e-sportowych z przykładowymi tytułami:

Gatunek gier	Gry
Gry walki	Street Fighter, Super Smash Bros., Tekken
FPS	Doom, Quake, CS, CS:GO, CoD, Halo, Battlefield, Overwatch, Rainbow Six: Siege
RTS	Starcraft: Brood War, Starcraft II, Warcraft III
Sportowe	Fifa, NBA 2K, PES, Rocket League
Wyścigowe	TrackMania, iRacing
MOBA	Dota 2, LoL, Smite, HotS
Inne	Hearthstone, Pokemon, Tetris, World of Tanks, World of Warcraft

Tabela 6. Przykładowe gatunki gier e-sportowych

Turnieje - przykłady

Turnieje sportów elektronicznych to dziś w większości fizyczne wydarzenia. Zawodnicy występują przed publicznością na żywo a turniej może być częścią większego zgromadzenia takiego jak *Dreamhack*, lub osobnym wydarzeniem takimi jak *WCG*. Turnieje przyjmują kilka formatów. Najczęściej polegają na pojedynczej lub podwójnej eliminacji, bywają czasami łączone z rywalizacją w grupach. Odbývają się pod czujnym okiem sędziów zawodów i osób odpowiedzialnych za monitorowanie oszustw.

W przypadku dobrze znanych gier suma nagród może wynieść kilkadziesiąt milionów dolarów rocznie. Od września 2016 roku *Dota 2* przyznała około 86 mln USD w nagrodach w ramach 632 zarejestrowanych turniejów a 23 graczy wygrało ponad 1 mln USD. *League of Legends* sponsorowało nagrody za około 30 mln USD w 1749 zarejestrowanych turniejach, ale oprócz nagród pieniężnych *Riot Games* zapewnia pensje dla graczy z lig profesjonalnych spod szyldu *LolSports.com*. W sierpniu 2018 roku odbył się coroczny międzynarodowy turniej *Dota2 The International 2018*, który pobíł rekord największej puli nagród w historii turnieju e-sportowego. Pula nagród wyniosła ponad 25 mln USD.[34]

Często twórcy gier zapewniają nagrody pieniężne bezpośrednio na turniej, ale sponsoring może pochodzić również od firm trzecich, zazwyczaj sprzedających sprzęt komputerowy, napoje energetyczne lub oprogramowanie. Turnieje e-sportowe to też doskonała okazja do organizacji konwentów tematycznych, mangi i anime oraz doskonała okazja dla firm zajmujących się produkcją sprzętu komputerowego do wystawienia swoich stoisk.

Lista wybranych turniejów e-sportowych wraz krajami, w których się odbywa i rokiem ustanowienia zawodów:

Nazwa	Miejsce	Rok	Opis
Battle.net World Championship Series (BlizzCon)	Na całym świecie	2012	StarCraft II, World of Warcraft, Overwatch, Hearthstone
Call of Duty World League	Australia, Canada, Europa, Meksyk, USA	2016	Call of Duty: Black Ops III na PS4
Cod Championship	USA	2013	Mistrzostwa świata
Capcom Cup	USA	2013	Turniej w grę Street Fighter finansowany przez firmę Capcom
CompeteLeague	Kanada, USA, Meksyk, Europa	2016	Największa amatorska liga e-sportowa LoL w Europie i Ameryce Północnej
Cybergamer	Australia, Nowa Zelandia, Oceania	2007	Główne wydarzenie e-sportowe regionu Oceanii
Dragon Ball FighterZ World Tour	Na całym świecie	2018	Globalna sieć turniejów DB FighterZ
Dreamhack	Szwecja	1994	Największy festiwal komputerowy na świecie łączony z zawodami e-sportowymi
ESL	Online	2000	Organizacja zajmująca się

			turniejami online i LAN, twórcy IEM
Evolution Championship Series	USA	1996	Największy turniej gier walki w USA
eXTREMESLAND	Azja południowo- wschodnia, Australia, Nowa Zelandia i Oceania	2016	Azjatycki turniej CS:GO. Finały odbywają się w Chinach
FIFA Interactive World Cup	Na całym świecie	2004	Coroczne mistrzostwa świata w grę FIFA
Fortnite World cup	Na całym świecie	2019	Turniej Fortnite z pulą nagród wynoszącą 100 mln USD
Global Starcraft II League	Korea Południowa	2010	Główny turniej StarCraft II w Korei Południowej
IMBA esports	Australia	2017	Australijska Liga e-sportowa
LoL Champions Korea	Korea Południowa	2013	Główne rozgrywki LoL w Korei Południowej
LoL India Champions Cup	Bangladesz, Indie, Pakistan, Sri Lanka	2018	Turniej organizowany w Azji południowej, LoL
Sea Tour (Garena Premier League)	Indonezja, Malezja, Filipiny, Singapur, Tajlandia	2015	LoL w Azji południowej
LoL Pro League	Chiny	2013	LoL w Chinach
LoL Rift Rivals	Wszystkie regiony z oficjalną ligą <i>Lol esports.com</i>	2017	Międzykontynentalne rozgrywki LoLa
LoL World Championship	Na całym świecie	2011	MŚ w LoL
Overwatch League	USA	2017	Główne rozgrywki ligowe w grę Overwatch
SMITE World Championship	USA	2014	Mistrzostwa Świata w SMITE
Tekken World Tour	Na całym świecie	2017	Seria turniejów gry Tekken
The International	Na całym świecie	2011	Największy turniej na świecie w grę Dota 2
Wargaming.net League	Na całym świecie	2013	Liga World of Tanks
LEC (EU LCS)	Europa	2013	Liga LoL w Europie
LCS (NA LCS)	USA	2013	Liga LoL w USA
Ultraliga	Polska	2018	Liga LoL w Polsce
IEM	Na całym świecie	2007	MŚ w grach komputerowych z finałem w Katowicach

Tabela 7. Turnieje rangi światowej

Zespoły i stowarzyszenia

W esporcie profesjonalni gracze w większości przypadków grają w zespołach. Zespoły takie jak *Team SoloMid*, *Cloud9*, *Fnatic*, *G2 Esports* czy *SK Telecom T1* składają się z kilku profesjonalnych graczy oraz trenerów, menadżerów, dietetyków, psychologów i analityków. Gracze jedzą, mieszkają i trenują wspólnie w tzw. *gaming house'ie*.

Oprócz nagród pieniężnych z wygranych w turniejach, gracze otrzymują również pensje i premie. Sponsorowanie drużyny może pokryć koszty podróży turniejowych lub sprzętu treningowego. Znanymi firmami wspierającymi finansowo i sprzętowo e-sport są firmy *Logitech* i *Razer*. Zespoły prezentują tych sponsorów na swojej stronie internetowej, na koszulkach, w których występują na scenie jak i na portalach społecznościowych organizacji i/lub graczy. W 2016 roku największe zespoły miały ponad milion obserwujących w mediach społecznościowych (*Facebook*, *Instagram*, *Twitter*). Stowarzyszenia zrzeszające graczy obejmują *Koreańskie Stowarzyszenie e-Sportu (KeSPA)*, *Międzynarodową Federację e-sportową (IeSF)* oraz *Światowe Stowarzyszenie Graczy E-sportowych (WESA)*. [34]

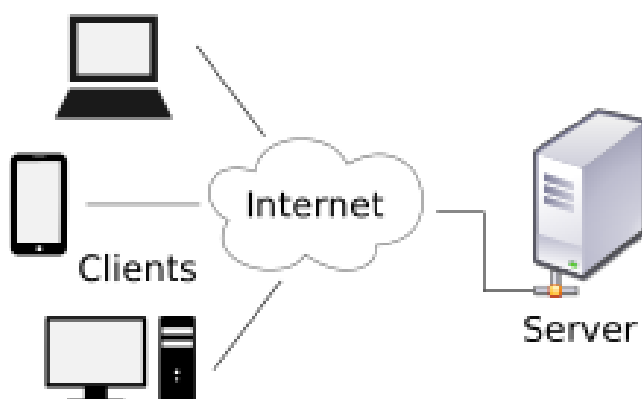
Niektórzy byli sportowcy zainwestowali w e-sport. Przykładami tego są koszykarze Rick Fox i jego zespół *Echo Fox*, Jeremy Lin i *Team VGJ*, Shaquille O'Neal i *NRG Esports*. Niektóre stowarzyszenia drużyn piłkarskich, takie jak *FC Schalke04* w Niemczech, francuskie *Paris Saint-Germain*, tureckie drużyny *Besiktas JK*, *Fenerbahce SK* i *Galatasaray* czy też *Panathinaikos FC* w Grecji prowadzą i sponsorują własne drużyny e-sportowe. [34] W Polsce przykładem tego są drużyna *Legii Warszawa* w grze *FIFA* oraz *Wisła Płock* i *Piast Gliwice* w grze *League of Legends*.

Architektura Klient-Serwer

Modelem wykorzystywanym w przypadku wszystkich gier online, nie tylko gier *MOBA*, jest model Klient-Serwer. Jest on rozproszoną architekturą aplikacji. Umożliwia podział zadań i obciążeń pomiędzy dostawcami zasobów lub usług zwanymi serwerami, a usługobiorcami zwanymi klientami.

Często klienci i serwer komunikują się za pośrednictwem sieci komputerowej na oddzielnym sprzęcie ale zarówno klient jak i serwer mogą znajdować się w tym samym systemie. W przypadku gier *MOBA* zazwyczaj druga opcja nie ma swojego zastosowania. Poza grami internetowymi przykładami aplikacji komputerowych korzystających z modelu klient-serwer są poczta e-mail oraz sieć WWW.[36]

Host serwera uruchamia jeden lub więcej programów serwera, które udostępniają swoje usługi jednemu lub więcej klientom. Klient nie udostępnia żadnych zasobów ale żąda od serwera danych jego funkcji lub usług. Klienci inicjują zatem połączenie zwane sesją komunikacyjną (ang. *session*) z serwerami oczekującymi na przychodzące żądania (ang. *requests*).[36]



Rysunek 17. Zobrazowanie architektury typu klient-serwer

Tworzenie własnej gry, silniki sieciowe i badanie ich jakości

Ogólne tworzenie gier mocno wyewoluowało na przestrzeni ostatniej dekady. Kiedyś developerzy zmuszeni byli pisać setki linii kodu tylko i wyłącznie za pomocą wybranego środowiska programistycznego. Teraz jest to dużo prostsze.

Pisząc grę, sieciową czy nie, deweloperzy mają do wyboru kilka silników tworzenia gier takich jak Unreal Engine czy Unity 3D. Mogą łatwo tworzyć grafiki i animacje za pomocą programów takich jak Adobe Photoshop, Autodesk 3ds MAX czy Autodesk Maya. To znacznie skraca pracę twórcy/twórców gry.

Porównanie silnika Unity 3D z silnikiem Unreal Engine

Unity 3D	Unreal Engine 4
Wspiera 25 platform w tym te najlepsze do monetyzacji produktu, wspiera język C#, interfejs <i>Drag and Drop</i> , zoptymalizowany w kierunku wielu urządzeń silnik graficzny	Lepszy silnik graficzny niż w Unity, wspiera język C++, Asset Store z wieloma gotowymi wzorcami gier, wspiera ponad 10 platform, Epic Games (twórcy silnika) pobierają 5% od wszystkiego co zarobi gra

Tabela 8. Porównanie Unity 3D z Unreal Engine 4

Jednak co dalej? Programista/twórca posiada gotową grę i co ma z nią dalej zrobić zwłaszcza, że powinna być umieszczona w sieci by gracze mogli w nią grać?

Z pomocą przychodzą gotowe rozwiązania silników/usług sieciowych do tworzenia gier jak np. *Networking Unity*. Ale czy jest to najlepsze dostępne rozwiązanie? Czy nie ma nic lepszego od domyślnego narzędzia z dokumentacji?

W tej pracy zostanie przybliżone czytelnikom więcej możliwości rozwiązania tego zagadnienia, przedstawione między innymi zostanie czym jest *Photon* i *GameSparks*. Zostaną przedstawione i porównane poszczególne rozwiązania zastosowane w obu środowiskach, jak i otrzymane z ich pomocą rezultaty.



Rysunek 18. Logo Unity 3D

3. Specyfikacja wymagań

3.1. Projekt gry

Praktyczną częścią pracy dyplomowej magisterskiej jest utworzenie okienkowej gry komputerowej typu *Multiplayer Online Battle Arena* w architekturze Klient-Serwer. Gra posłuży za tester wybranych funkcji sieciowych, które to zostaną omówione w Rozdziale 4 pt. Metodyka pracy. W grze zostaną zaimplementowane podstawowe funkcjonalności takie jak sterowanie, znajdowanie ścieżek (*pathfinding*) czy wystrzelenie pocisku.

Domyślnie zaimplementowanym matchmakerem jest matchmaker Unity Networking. Gracz po uruchomieniu programu może dołączyć do gry na kilka różnych sposobów:

- LAN Host – użytkownik aplikacji tworzy rozgrywkę pod swoim adresem IP, jednocześnie biorąc w niej udział.
- LAN Client – gracz dołącza do gry po dokładnym adresie IP.
- Opcja wyszukiwania gry w matchmakerze. Gracz wybiera konkretną grę, do której chce dołączyć z listy gier.

W celu utworzenia gry użytkownik aplikacji ma do wyboru trzy opcje:

- LAN Host – użytkownik aplikacji tworzy rozgrywkę pod swoim adresem IP, jednocześnie biorąc w niej udział.
- LAN Server Only – użytkownik tworzy rozgrywkę pod swoim adresem IP, nie biorąc w niej udziału.
- Opcja tworzenia rozgrywki w matchmakerze. Użytkownik może nadać konkretną nazwę rozgrywce, co ułatwia jej znalezienie chętnym graczom.

3.2. Wymagania niefunkcjonalne

Bezpieczeństwo – gra zabezpieczana po stronie usług sieciowych.

Dostępność – gra dostępna na platformę Microsoft Windows jako aplikacja okienkowa, użytkownik ma do niej stały dostęp 24 godziny na dobę, 7 dni w tygodniu.

Rozszerzalność – ciągła możliwość rozwoju gry. Możliwość implementacji dodatkowych funkcji mających na celu zwiększenie zawartości gry i jej rozwój.

Użyteczność – gra prosta w obsłudze, służąca za tester wybranych usług sieciowych.

Wydajność – w celu zapewnienia komfortu obsługi, gra nie korzysta ze zbędnych mechanizmów, zapewniając szybki dostęp i łączenie z siecią.

Ograniczenia licencyjne – gra stworzona na darmowej licencji Unity 3D, nie jest przeznaczona do umieszczenia w sklepach online lub do innej, dalszej dystrybucji.

Ograniczenia sprzętowe – przyjmuje się, że ograniczenia sprzętowe gry są takie jak ograniczenia sprzętowe silnika, w którym gra powstała. System operacyjny Microsoft Windows 7 64 bitowy SP1+ lub wyższy, procesor wspierający instrukcje SSE2, karta graficzna wspierająca DirectX w wersji 10 (model shaderów 4.0).

4. Metodyka pracy

4.1. Wykorzystane narzędzia

Poniżej przedstawione i omówione zostaną wszystkie narzędzia, technologie, *framework*'i oraz usługi sieciowe wykorzystane w tej pracy.

4.1.1. Unity 3D

Silnik gier przeznaczony do projektowania i tworzenia gier dwu- i trójwymiarowych oraz innych aplikacji. Działa na systemach operacyjnych MS Windows, macOS i Linux. Pozwala tworzyć programy na urządzenia mobilne, konsole i komputery stacjonarne.[37]

Pierwszy raz zaprezentowany w 2005 roku na konferencji Apple's Worldwide Developers Conference. Jego głównym językiem jest C#, dawniej wspierał też *Boo* i *UnityScript* będące podobny w swojej składni do języka *JavaScript*. Wraz z wprowadzeniem wersji 5.0 i późniejszych, możliwość pisania skryptów w tych językach została usunięta.[37]

Do wersji 4.6 był podzielony na wersję licencji płatnej i darmowej zawierającej ograniczony zbiór funkcjonalności. Wraz z wcześniej wspomnianą wersją 5.0 prawie wszystkie funkcje silnika zostały udostępnione programistom w darmowej wersji z ograniczoną liczbą zarobków użytkownika nieprzekraczającą stu tysięcy USD rocznie.[37]

Unity wspiera następujące silniki graficzne:

1. *Direct3D* na platformę Windows i Xbox One;
2. *OpenGL* na Linux, macOS, i Windows;
3. *OpenGL ES* na Android i iOS;
4. *WebGL* w sieci oraz własne silniki graficzne na konsolach gier video.

Ponadto Unity wspiera aplikację niskiego poziomu o nazwie *Metal* na iOS i macOS oraz *Vulkan* na Android, Linux i Windows, tak samo jak *Direct3D 12* na Windows i Xbox One.[37]

W grach dwuwymiarowych Unity pozwala na importowanie różnorodnych materiałów jak i zaawansowanych silników renderujących świat 2D. Dla gier 3D, Unity umożliwia różne rodzaje kompresji tekstur, *mipmapping*, ustawienie rozdzielczości dla każdej platformy, którą wspiera silnik gry. Zapewnia obsługę mapowań takich jak mapowanie odbicia, mapowanie wypukłości, mapowanie paralaksy, zezwala na dynamiczne cieniowanie za pomocą mapowania cieniem oraz na pełnoekranowe efekty przetwarzania końcowego.[37]

Od 2016 roku Unity oferuje również zestaw usług opartych na chmurze obliczeniowej, w skład których wchodzi Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Everyplay, Unity IAP (*In app purchase* - dla Apple i sklepu Google), Unity Multiplayer, Unity Performance Reporting, Unity Collaborate and Unity Hub.[37]

4.1.2. Visual Studio

Visual Studio jest zintegrowanym środowiskiem pracy deweloperów (*IDE*) zaprojektowanym przez giganta Microsoft. Służy do tworzenia oprogramowania dla programów komputerowych, stron, aplikacji i serwisów internetowych oraz aplikacji mobilnych.[38]

Visual Studio wykorzystuje platformy programistyczne Microsoft, takie jak:

1. Windows API,
2. Windows Forms,
3. Windows Presentation Foundation,
4. Windows Store,
5. Microsoft Silverlight.

Może produkować zarówno kod natywny (dla środowiska .NET C#) jak i kod zarządzany.[38]

Visual Studio zawiera edytor kodu obsługujący IntelliSense (komponent uzupełniania kodu) oraz refaktoryzację (zmianę) kodu. Zintegrowane narzędzie debugujące działa zarówno jako debugger na poziomie kodu źródłowego, jak również na poziomie komputera. Inne wbudowane narzędzia obejmują profiler kodu, projektanta formularzy do budowania aplikacji GUI, projektanta stron internetowych, projektanta klas i projektanta schematów baz danych. Akceptuje wtyczki, które zwiększają funkcjonalność na niemal każdym poziomie – w tym dodawanie obsługi systemów kontroli źródeł (takich jak Subversion i Git) oraz dodawanie nowych zestawów narzędzi, takich jak edytory i narzędzia do projektowania grafik do języków specyficznych dla danej domeny lub zestawów narzędzi dla innych aspektów cyklu życia tworzonego oprogramowania. [38]

Visual Studio obsługuje 36 różnych języków programowania i umożliwia edytorowi kodu i debuggerowi obsługę (w różnym stopniu) prawie dowolnego języka programowania, pod warunkiem, że istnieje usługa specyficzna dla języka. Języki wbudowane to *C*, *C++*, *C++/CLI*, *Visual Basic .NET*, *C #*, *F #*, *JavaScript*, *TypeScript*, *XML*, *XSLT*, *HTML* i *CSS*. Obsługa innych języków, takich jak *Python*, *Ruby*, *Node.js* i *M* jest dostępna za pośrednictwem wtyczek zewnętrznych. Języki *Java* i *J#* były obsługiwane w przeszłości.[u]

Najbardziej podstawową edycją Visual Studio jest edycja Community, dostępna bezpłatnie. Aktualnie obsługiwana stabilna wersja Visual Studio to 2017. Microsoft ogłosił prace nad wersją 2019 w dniu 6 czerwca 2018 roku, obecnie dostępna jest wersja wstępna programu.[38]



Rysunek 19. Logo Visual Studio

4.2. Wykorzystane technologie

4.2.1. C# (C Sharp)

C# to wieloparadygmataowy, uniwersalny język programowania zorientowany obiektowo, zaprojektowany w latach 1998-2001 na platformę .NET firmy *Microsoft*. Obejmuje silne typowanie, zakresy leksykalne jak i programowanie funkcyjne i komponentowe.[39]

Posiada wiele cech wspólnych z takimi językami programowania jak *C++*, *Java*, *Delphi* czy *Object Pascal* m.in. obiektowość, właściwości, indeksery, delegaty, zdarzenia czy *Garbage Collector* (oczyszczanie pamięci). Najnowsza wersja języka to 7.3, wydana 7 maja 2018 roku. Jest głównym językiem wykorzystywanym do pisania skryptów w silniku gier Unity 3D.

Cele języka C# wg standardu ECMA:

1. *Język ma być prostym, nowoczesnym, zorientowanym obiektowo językiem programowania ogólnego;*
2. *Język i jego implementacje powinny zapewniać wsparcie dla zasad inżynierii oprogramowania takich jak silne sprawdzanie typów, sprawdzanie granic tablic, wykrywanie prób użycia niezainicjalizowanych zmiennych i automatyczne usuwanie śmieci. Ważna jest wytrzymałość oprogramowania, trwałość i wydajność programisty;*
3. *Język przeznaczony jest do używania w tworzeniu komponentów oprogramowania odpowiednich do wdrożenia w środowiskach rozproszonych;*
4. *Bardzo ważnym elementem dla kodu źródłowego i programistów jest przenośność.*
5. *Wsparcie dla internacjonalizacji;*
6. *Język jest przeznaczony do pisania aplikacji dla systemów hostowych i systemów wbudowanych, od bardzo dużych, które wykorzystują zaawansowane systemy operacyjne, aż po bardzo małe, posiadające dedykowane funkcje;*
7. *Aplikacje pisane przy pomocy języka C# mają być ekonomiczne pod względem wymagań pamięci i mocy obliczeniowej ale język nie ma na celu bezpośredniej rywalizacji o wydajność i rozmiar z językiem C lub Asemblerem.[39]*

```
class Foo
{
    const double X = 3;

    Foo()
    {
        const int Y = 2;
    }
}
```

Rysunek 20. Przykładowy listing kodu w języku C#

4.3. Wykorzystane usługi sieciowe

Na potrzeby tej pracy omówione zostaną trzy zestawy rozwiązań sieciowych – *UNet*, *Photon Unity Networking* i *Gamesparks*. Poniżej analiza podanych usług sieciowych. Między innymi na podstawie tego podrozdziału i zawartych w nim informacji wyciągnięte zostaną wnioski służące ocenie przydatności omawianych usług.

4.3.1. Unity Networking - UNet

UNet to domyślne narzędzie silnika gier *Unity 3D* służące do tworzenia gier wieloosobowych w sieci. To zbiór funkcji sieciowych odpowiedzialnych za działanie programu online, oparte na architekturze *klient-serwer*. Prostota tego rozwiązania jest jednym z jego największych atutów, jest bardzo dobrym sposobem na rozpoczęcie nauki tworzenia sieciowych gier wieloosobowych.[13]

Unity Networking posiada swoją własną aplikację wysokiego poziomu (*High Level API*, w skrócie: *HLAPI*). Dzięki tej *HLAPI* twórca zyskuje dostęp do funkcji sieciowych, które obejmują większość typowych dla gier wieloosobowych wymagań, bez konieczności martwienia się szczegółami implementacji *niższego poziomu*. Owe *API* wysokiego poziomu pozwala programistom m.in. na:

- kontrolę stanu sieci w grze za pomocą *Menadżera sieci (Network Manager)*,
- obsługę gry w trybie *client hosted*, w którym hostujący również jest graczem,
- serializację danych,
- wysyłanie i odbieranie wiadomości sieciowych,
- wysyłanie poleceń sieciowych od *klientów* do *serwerów*,
- tworzenie *zdalnych wywołań procedur (remote procedure call, RPC)*,
- wysyłanie *zdarzeń sieciowych (network events)* z *serwerów* do *klientów*.[13]

Unity Networking jest zintegrowany z silnikiem gier i jego edytorem. Umożliwia on pracę z komponentami i pomocami wizualnymi. Celem tego jest zbudowanie gry dla wielu graczy oraz wsparcie twórcy gry przez cały okres jej produkcji i wydania. UNet zapewnia m.in.:

- komponent *NetworkIdentity* dla obiektów sieciowych,
- komponent *NetworkBehaviour* dla skryptów sieciowych,
- konfigurowalną, automatyczną synchronizację transformacji obiektów,
- automatyczną synchronizację zmiennych skryptowych,
- obsługę obiektów sieciowych umieszczonych w scenach *Unity 3D*,
- inne komponenty sieciowe,
- *matchmaking*,
- serwer przekaźnikowy,
- umożliwia grę przez *Internet* bez dedykowanego serwera.[13]

4.3.2. Photon i PUN

4.3.2.1. Photon

Photon to szybki, niezawodny, skalowalny i niezależny silnik sieciowy oraz platforma dla gier wieloosobowych. Używany przez programistów do tworzenia i uruchamiania gier wieloosobowych czasu rzeczywistego.[14]

Aplikacje tworzone przy pomocy *Photon* potrzebują unikalnego identyfikatora w postaci *AppId* by móc się z nim połączyć. Każdy deweloper dostaje taki identyfikator po dokonaniu rejestracji aplikacji w serwisie internetowym *Photon*. Korzystanie z niego jest bardzo proste i ogranicza się często do skorzystania z kilku linijek kodu.[14]

Usługi *Photon Engine* można podzielić ze względu na funkcje typu wielu graczy, funkcje komunikacyjne oraz funkcję self-hostingu. W pierwszej grupie wyróżnia się *Photon REALTIME*, *PUN*, *BOLT* oraz *QUANTUM*. [14]

1. *Photon REALTIME* to narzędzie stworzone dla twórców gier synchronicznych jak i asynchronicznych bez podziału na to czy tworzący są małym studiem typu *indie* czy wielką korporacją AAA.
2. *Photon PUN* pozwala użytkownikowi cieszyć się z bliskiej integracji rozwiązań *Photon* z *Unity* umożliwiając przez to łatwiejszy proces tworzenia i uruchamiania gier. Przede wszystkim pozwala na łatwy eksport gry na każdą z wspieranych przez *Unity* platform, włącznie z konsolami.
3. *Photon BOLT* to idealny zbiór funkcjonalności do gier typu *FPS/TPS*, gier akcji oraz zyskujących ostatnio na popularności gier typu *Battle Royal*. Wspiera *Unity*. Autorski sposób poruszania się w połączeniu z potężną gamą systemów zdarzeń sprawia, że coraz większa grupa deweloperów korzysta z *BOLT'a* w nadziei na stworzenie kolejnego *PUBG*, *Fortnite'a* bądź *APEX Legend*.
4. *Photon QUANTUM* to deterministyczny silnik przeznaczony do szybkich gier akcji czasu rzeczywistego, pozwalający osiągnąć jak najlepsze rezultaty połączeń. Zaprojektowany by służył programistom w tworzeniu gier online typu *MOBA*, *RTS'ów*, gier sportowych lub gier walki. Korzystają z niego m.in. studia *Mediatonic* czy *FGOL* będący własnością firmy *Ubisoft*.

Druga grupa to funkcje pozwalające na komunikowanie się graczom. W tej kategorii *Photon* prezentuje potencjalnym użytkownikom dwa rodzaje usług poprzez komunikację czatową i/lub głosową. Są nimi odpowiednio *Photon Chat* oraz *Photon Voice*. [14]

W ostatniej grupie zwanej self-hostingiem *Photon* daje użytkownikom możliwość zbudowania ich własnego rodzaju gier wieloosobowych wraz z własnymi, w pełni autorytatywnymi i miarodajnymi serwerami. Pozwala na ich hosting i działanie na zasadach użytkownika. Przykładem gier opartych na tym rozwiązaniu jest *Family Guy Online* od *A.C.R.O.N.Y.M./FOX* oraz *The Witcher: Battle Arena* stworzone przy współpracy studia *Fuero Games* z *CD Projekt Red*. [14]

4.3.2.2. PUN

PUN (ang. *Photon Unity Networking*) czyli *Photon* dla silnika Unity 3D do gier wieloosobowych. Jednym z jego najbardziej kluczowych atutów jest łatwo przystosowujący się system znajdowania gier, w którym twórca ma możliwość konfigurowania go w taki sposób by mógł wyszukać grę jeszcze szybciej. Pozwala na komunikację klient – serwer.[14]

PUN może być eksportowany na każdą platformę wspieraną przez Unity i jest dostępny w dwóch wersjach, *PUN Free* – wersja darmowa, zawierająca pliki testowe, pliki demonstracyjne, głównie nadająca się do przeprowadzania testów aplikacji bądź gry oraz *PUN Plus* stworzony do umieszczania gotowych produktów na serwerze, będący wersją płatną *PUN*. [14]



Rysunek 21. Logo PUN

Atutami *Photon Unity Networking* są:

1. Chmura czasu rzeczywistego – gry tworzone przy pomocy *PUN* są umieszczane na globalnej, rozproszonej chmurze obliczeniowej Photon Cloud w celu zagwarantowania graczom na całym świecie niskich opóźnień i możliwie najkrótszego czasu przesyłu danych w obie strony (klient - serwer).
2. Gry wieloosobowe (ang. *Multiplayer*) – zapewnia połączenie, dopasowywanie ustawień i rozgrywkę, *PUN* jest solidną podstawą do każdego rodzaju gier wieloosobowych opartych na Unity 3D. Ściąga z dewelopera ciężar zarządzania zapleczem gry jak statystyki dzięki czemu twórca może skupić się na zawartości swojego projektu.
3. Wielopatformowość – użytkownik ma możliwość wyeksportowania swojej gry na urządzenia mobilne, komputery stacjonarne, strony www lub konsole – *Photon* to standardowa wielopatformowa usługa dla gier wieloosobowych, dla wielu deweloperów obecny numer jeden wśród usług dla gier wytwarzanych przy pomocy Unity 3D.
4. Najwyższa skalowalność – gry stworzone w oparciu o *PUN* skalują się płynnie i automatycznie w chmurze *Photon Cloud*: od kilku do kilkudziesięciu tysięcy

- jednoczesnych użytkowników.
5. System Matchmakingu – pozwala na dopasowanie graczy do gier losowo lub poprzez sparаметryzowane wyszukiwania. Pozwala na utworzenie listy otwartych pokoi, z których gracz może sobie wybrać ten, do którego ma aktualnie ochotę dołączyć.
 6. Personalizacja – *PUN* obsługuje wszelkiego rodzaju gry oparte na pokojach na dowolnej platformie do gier. Nie ważne czy jest to *FPS*, strzelanka zręcznościowa, *MOBA*, *RPG* czy *RTS* – *PUN* jest w stanie obsłużyć każdy z tych typów.
 7. Klient-serwer – *PUN* pozwala na bezproblemowe połączenia bez potrzeby zwracania sobie głowy błędami związanymi z protokołami *UDP*, *TCP*, *http* czy też w *Websocket*'ach.
 8. Niezrównana elastyczność – twórca gry jest w stanie dopasować użytkownika iOS Game Center do osoby korzystającej z usług Google Play, uwierzytelnić użytkownika za pomocą Facebooka lub dodać niestandardowe uwierzytelnianie wykorzystując takie serwisy jak *PlayFab* lub jeśli ma taką ochotę może podpiąć własne oprogramowanie uwierzytelniające.
 9. Bezproblemowy system cenowy – uczciwy i przejrzysty, w 100% oparty na liczbie współbieżnych użytkowników (*Concurrent Users* - *CCU*). Zapewnia uczciwą elastyczność płatniczą: Użytkownik zaczyna od bezpłatnego planu *PUN Free* i ma możliwość przejścia na wersję wyższą (płatną) lub zejścia z wyższej na darmową kiedy chce. W tym drugim przypadku użytkownik nie cierpi ze względu na przekroczenie *CCU* i natychmiastowe cięcia z tym związane.

PUN FREE



No-cost package with various demos, pre-made scripts and reference documentation. Exports to basically all platforms.

GET PUN

PUN PLUS



Same content as PUN FREE, plus a 100 concurrent user plan for the Photon Cloud (approx. 40k MAU, valid 60 months).

GET PUN

for just \$95

Rysunek 22. Dostępne wersje PUN

Szacuje się, że ponad 334 tysięcy deweloperów tworzy gry sieciowe przy pomocy Unity 3D i *PUN* co czyni z *Photon*'a numer jeden wśród usług sieciowych dla programistów i twórców gier.[14]

Przykładami gier, które powstały przy pomocy PUN są m.in.:

1. *Flashing Lights: Police – FIRE – EMS* od Nilsa Jakrinsa i Excalibur Games
2. *Disco Dodgeball* od 82apps
3. *Shadowgun Legends* studia Madfinger Games
4. *Super RocketBall* od Deep B Bhuvu

4.3.3. GameSparks

Gamesparks to platforma sieciowa, oparta na chmurze obliczeniowej Microsoft Azure, rozwiązanie dla twórców gier, pomagające im w budowie ich komponentów aplikacji/gier działających po stronie serwera.[15]

Wspiera zarządzanie kontami graczy m.in. kilka rodzajów rejestracji i logowania. Zbiera wybrane przez twórcę gry dane na temat poczynąń gracza, które później deweloper może wykorzystać w celu poprawienia jakości swojego produktu.



Rysunek 23. Logo Gamesparks

Dane są przypisane do każdego konta gracza, usprawniając przejrzystość ich odczytu i sprawdzenia każdego przypadku osobno. Daje dostęp do tabel najlepszych graczy, systemu osiągnięć zachęcającego graczy do rozwijania swojej przygody z produktem czy też jest w stanie odpowiadać za rozdysponowanie lub sprzedaż wirtualnych walut w grze.[15]



Rysunek 24. Logo Gamesparks

Gamesparks można podzielić na kilka frakcji usług sieciowych. Pierwszą frakcją są funkcje socialne i funkcje wieloosobowe, drugą frakcję stanowią funkcje meta-gry i aspekty ekonomiczne tworzonej gry. Dalej, w trzeciej frakcji, *Gamesparks* oferuje operacje czasu rzeczywistego. Czwarta grupa usług to tzw. *Gamesparks core*. Osobną, piątą kategorię usług stanowi integracja z wieloma frameworkami i urządzeniami docelowymi tworzonej gry. Ostatnią, szóstą frakcją usług są funkcje infrastruktury i operacji. Na następnej stronie zostały przedstawione poszczególne frakcje wraz z ich usługami.[15][40][43]

Social i multiplayer	Meta-gra i ekonomia	Operacje Live Game	Gamesparks Core	Integracja	Infrastruktura i operacje
Tablice wyników	Waluty	Zarządzanie graczem	Niestandardowe wywołania API	Silniki gier	Dynamicznie skalowalna pojemność w Chmurze
Wyzwania i multiplayer turowy	Dobra wirtualne	Segmentacja	Spersonalizowana logika po stronie serwera	Platformy i sklepy	Zarządzanie infrastrukturą wysokiej jakości
Multiplayer czasu rzeczywistego	System osiągnięć	Analityka	Database-as-a-Service	Konsole gier	Dedykowane klastry
Matchmaking, lobby i rywalizacja	IAP i wykrywanie oszustw	Kampanie i automatyzacja marketingu	Binary Asset Management	Sieci społecznościowe	Wybór regionu
Zespoły	Zarządzanie katalogami i sklepem	Eksperymentowanie i A/B testy	Powiadomienia i wiadomości	Zarządzanie zawartością	Premium SLA
Czat		Personalizowany panel sterowania Backoffice'em	Uwierzytelnienie		
		Role i uprawnienia	Testowanie		
		Promocje weekendowo sezonowe	System wersji i publikacji		
			Planowanie		

Tabela 9. Tabela przedstawiająca podział na frakcje funkcjonalności Gamesparks

❖ Social i multiplayer

1. Tablice wyników

Gamesparks umożliwia zaprojektowanie dowolnej tablicy wyników porównującej wyniki graczy, z pełną elastycznością w stosunku do sposobu wyświetlania ich w grze. Tabele wyników w *GameSparks* są bardzo wydajne i zbudowane w celu skalowania do milionów użytkowników podczas klasyfikacji w czasie rzeczywistym.

2. Wyzwania i multiplayer z podziałem na tury

Deweloperzy mają możliwość zwiększenia rywalizacji pomiędzy graczami, dodając do swojej gry funkcję gry wieloosobowej w oparciu o *Gamesparks*. Twórca ma możliwość zapewnienia dodatkowej rywalizacji organizując grę w oparciu o tury (gracz do gracza) lub globalne wyzwania, aby zwiększyć zaangażowanie graczy w grę.

3. Multiplayer czasu rzeczywistego

Zoptymalizowany pod kątem niskich opóźnień i wysokiej skalowalności gry, *Gamesparks* pozwala z łatwością przekształcić wcześniej stworzone gry w całkowicie wciągające tytuły wieloosobowe.

4. Matchmaking, lobby i rywalizacja

W *Gamesparks* twórca gry ma możliwość wyboru w jaki sposób matchmaker ma dopasować graczy do siebie. Może zrobić to ściśle wg wytycznych lub luźno dopasować graczy w czasie rzeczywistym na podstawie dowolnego atrybutu. Dzięki różnym rodzajom dopasowania graczy do rozgrywki, twórca napędza współzawodnictwo między graczami.

5. Zespół

Dzięki zespołom w *Gamesparks* użytkownik może zaimplementować własne listy znajomych lub klany i wykorzystać je do zwiększenia zaangażowania społecznego graczy.

6. Czat

Czym byłby tryb wieloosobowy bez możliwości komunikowania się z tymi, z którymi się gra? Funkcja czatu umożliwia dodawanie rozmów w grze pomiędzy jednym lub większą liczbą graczy i zapewnia pełną elastyczność w zakresie sposobu prezentacji w grze. [40][43]



Rysunek 25. Gamesparks – Logo rywalizacji

❖ Meta-gra i ekonomia

1. Waluty

W *Gamesparks* programista może zarządzać maksymalnie 6 różnymi walutami w grze (w tym XP – punktami doświadczenia). Może używać walut, aby nagradzać graczy za osiągnięcia w grze i tworzyć systemy transakcyjne.



Rysunek 26. Gamesparks – logo walut

2. Dobra virtualne

Twórca gry ma możliwość zarządzania listami zapasów we wszystkich sklepach, zmieniając produkty i ceny w czasie rzeczywistym i wspierając złożone relacje między wieloma towarami i walutami w grze.

3. System osiągnięć

Zachęcanie graczy do angażowania się i zatrzymywania ich w stworzonej przez dewelopera grze poprzez wykorzystanie systemu nagród i osiągnięć. *Gamesparks* pozwala na dynamicznie przyznawanie wirtualnych dóbr i walut w miarę postępów w grze.

4. IAP i wykrywanie oszustw

Zarządzanie IAP w *Gamesparks* jest całkowicie oparte na różnych platformach i integruje się ze sklepami z aplikacjami Android, Apple i Windows Phone.

5. Zarządzanie katalogami i sklepem

Kiedy gracz kupuje jakąś rzecz w grze na jednej platformie, powinien mieć ją w swoim ekwipunku, gdy gra na innej platformie. Zarządzanie zakupami *Gamesparks* w aplikacji jest całkowicie oparte na integracji różnych platform i jest gotowe do integracji z systemami Android, Apple i Windows Phone App Stores. [40][43]

❖ Operacje Live Game

1. Zarządzanie graczem

Pełny dostęp do historii gracza i podsumowania jego aktywności w grze. *Gamesparks* zapewnia pełny przegląd graczy wraz z możliwością nagradzania najlepszych lub usuwania niechcianych graczy np. za postępowanie niezgodne z regulaminem gry.

2. Segmentacja

W grach opartych na silniku *Gamesparks* użytkownik ma możliwość pogrupowania graczy na segmenty i dalszą analizę ich zachowania. Dzięki temu użytkownik ma możliwość przygotowania i zastosowania konkretnych strategii rozwoju gry i jej konfiguracji dla każdego z tych segmentów w celu poprawienia grywalności lub naprawienia błędów w grze.

3. Analityka

Gamesparks pozwala na przechowywanie każdej najmniejszej informacji na temat wykonanej przez gracza akcji, każdego jego postępowania. System ten jest w pełni rozszerzalny, oparty o interfejs API umożliwiający konfigurowanie i śledzenie własnych zapytań. Pulpit nawigacyjny (dashboard) zapewnia natychmiastowy podgląd wydajności gry użytkownika.

4. Kampanie i automatyzacja marketingu

Deweloper ma opcję zarządzania promocją tytułu jak i przeprowadzanie kampanii marketingowej, wykorzystującej zaawansowaną segmentację. W *Gamesparks* deweloper może stworzyć spersonalizowane, kontrolowane przez niego ekrany operatora, aby usprawnić przebieg pracy nad grą i zwiększyć jej wydajność.

5. Eksperymentowanie i A/B testy

Konfiguracja eksperymentów przeprowadzanych na grach i Alfa/Beta testy wraz z możliwością wypróbowania różnych opcji konfiguracji dla losowo wybranych grup graczy przez określony czas. Przegląd i analiza wyników tych eksperymentów, w celu podjęcia decyzji i dostosowania ustawień konfiguracji gry w celu zwiększenia zaangażowania w nią graczy.

6. Personalizowany panel sterowania Backoffice'em

W *Gamesparks* w łatwy sposób użytkownik może stworzyć pozycje Community Manager'ów i operatorów gry w postaci GameMaster'ów, którzy zamiast niego będą zajmowali się graczami i ich problemami, dzięki temu twórca gry będzie miał więcej czasu na pracę nad jej zawartością.

7. Role i uprawnienia

Gamesparks zapewnia wiele rodzajów dostępu użytkowników do platform bez ryzyka naruszenia ich bezpieczeństwa, integralności konfiguracji lub danych.

8. Promocje weekendowo sezonowe

Obsługa wydarzeń i promocji weekendowych dla graczy oraz przeprowadzanie kampanii marketingowej gry na ich podstawie, wykorzystując zaawansowaną segmentację i zarządzanie graczami. Podobnie jak w punkcie 4. [40][43]

❖ **Gamesparks Core**

1. Niestandardowe wywołania API

Gamesparks zapewnia programiście możliwość zbudowania swojego własnego API po stronie serwera, w celu dodania do strony dodatkowych funkcji.

2. Spersonalizowana logika po stronie serwera

Tworzenie i zarządzanie bogatą logiką po stronie serwera za pomocą internetowego narzędzia do programowania w chmurze oraz debuggera. Pozwala dodawać logikę do istniejących funkcjonalności, korzystając z obszernej biblioteki zdarzeń.

3. Database-as-a-Service

Pozwala na budowę własnych kolekcji środowiska wykonawczego lub kolekcji metadanych. Zyskanie pełnego dostępu do danych gry za pośrednictwem interfejsu eksploratora NoSQL.

4. Binary Asset Management

Funkcja *GameSparks* służąca do zarządzania treścią do pobrania w formie dodatku do gry (DLC) lub do przechwytywania plików binarnych z urządzenia gracza by udostępniać je innym graczom. Umożliwia zarządzanie wszystkimi zasobami binarnymi gry po stronie serwera.

5. Powiadomienia i wiadomości

Zarządzanie całą komunikacją z graczami z jednego serwera. Ustawianie i konfiguracja wiadomości w grze i powiadomień typu push. Pozwala na przetłumaczenie wiadomości na dowolny język.

6. Uwierzytelnienie

W *Gamesparks* istnieje wiele opcji uwierzytelniania gotowych do integracji z wszystkimi niestandardowymi integracjami, a także wiodącymi sieciami społecznościowymi, takimi jak Facebook, Twitter, Google, Apple, Microsoft, Amazon i Steam, umożliwiającym korzystanie z funkcji społecznościowych, zakupów w aplikacji i powiadomień push.

7. Testowanie

Gamesparks pozwala na wysyłanie żądań API w ich surowej postaci, aby zobaczyć odpowiedzi na dowolną komendę. Udostępnia możliwość skopiowania podłączonego urządzenia w celu testowania dowolnej ilości konfiguracji gry.

8. Systemy wersji i publikacji

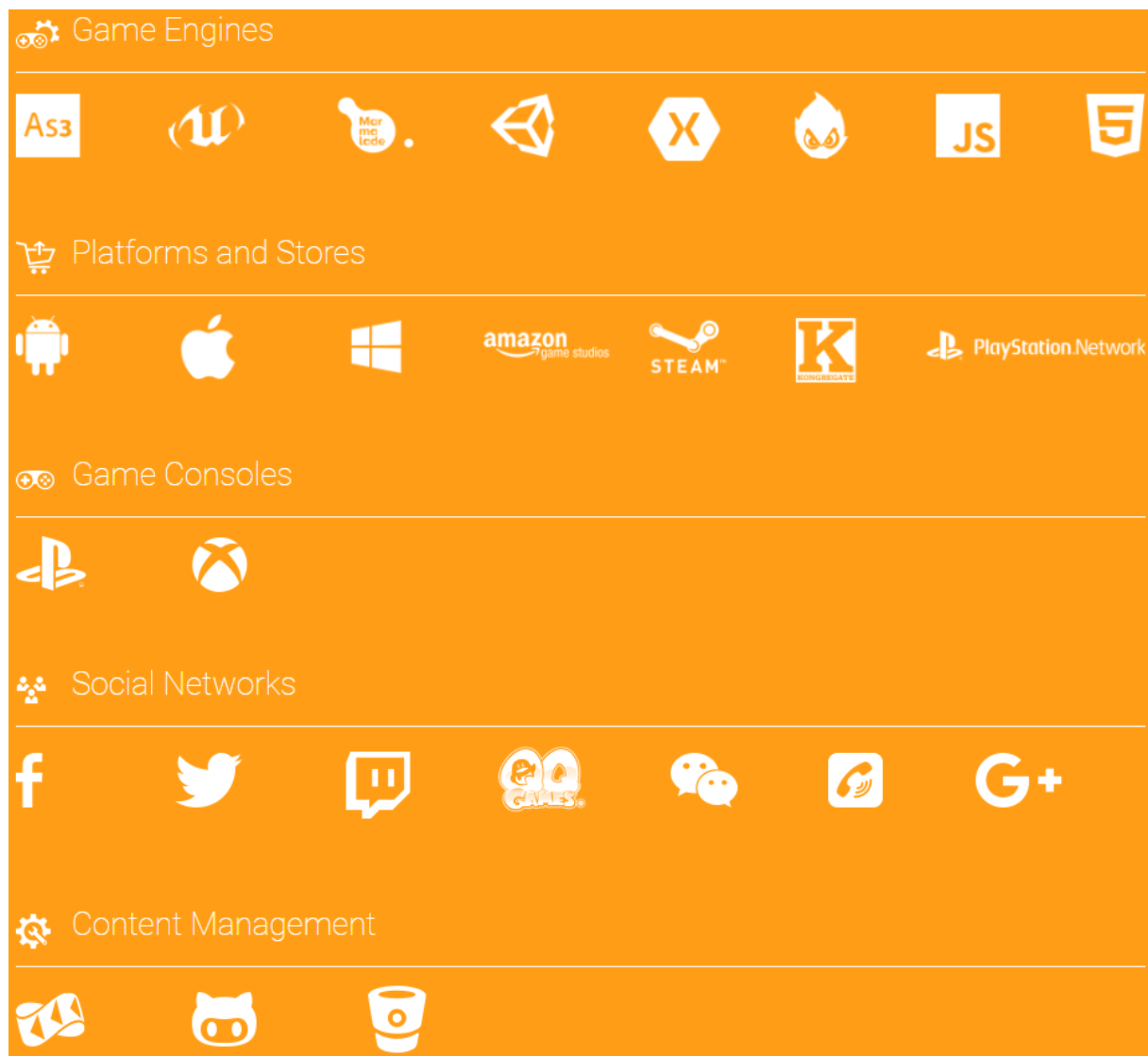
Twórcy gry korzystając z *Gamesparks* mogą utworzyć wiele wersji konfiguracji swojego produktu w dowolnym punkcie pracy nad nim. Pozwala to na bezpieczne zrezygnowanie z części niechcianych konfiguracji, niepotrzebnych kopii zapasowych lub ze skorzystania z wielu kompilacji, w celu dopięcia szczegółów i naprawy błędów przed zbliżającą się kampanią promocyjną produktu. [40][43]

9. Planowanie

Twórca zostaje wyposażony w narzędzie umożliwiające planowanie masowych zadań dla graczy, które będą im uruchamiane w określonym czasie oraz pozwalające wysyłać wiadomości do nich globalnie lub ukierunkowane, ograniczone przez dowolną konfigurację. [40][43]

❖ Integracja

Gamesparks zapewnia jedno zintegrowane środowisko, kompatybilne ze wszystkimi dużymi platformami odbiorczymi. Dostarcza szeroką gamę gotowych integracji ze wszystkimi głównymi silnikami gier takimi jak Unity 3D czy Unreal Engine, platformami jak Steam, sklepami jak Google Play i sieciami społecznościowymi jak Facebook czy Twitter oraz niestandardowymi dodatkami dostępnymi na żądanie. [40][43] Poniżej przedstawiona została plansza, na której można zobaczyć większość z nich:



Rysunek 27. Dostępne formy integracji z Gamesparks

❖ Infrastruktura i operacje

1. Dynamicznie skalowalna pojemność w Chmurze

Programiści nie muszą martwić się więcej tym, że zabraknie im miejsca na grę, aplikację, serwis. *Gamesparks* zapewnia im kontrolę nad wszystkim dzięki dynamicznie skalowalnym, najlepszym w swojej klasie wdrożeniom IaaS (AWS i Azure) oraz Centrum wsparcia 24/7.

2. Zarządzanie infrastrukturą wysokiej jakości

Gamesparks umożliwia korzystanie z pomocy eksperckiego zespołu wsparcia, dostarczając twórcy i jego grze aktualizacje oprogramowania i zarządzanie infrastrukturą 24/7.

3. Dedykowane klastry

Deweloper ma dowolność w działaniu, z *Gamesparks* może sam zaprojektować swoje rozwiązanie, aby zapewnić dodatkowe bezpieczeństwo i zwiększoną wydajność swojej aplikacji dzięki dedykowanym klastrum.

4. Wybór regionu

Pozwala na zwiększenie szybkości i wydajności gry dzięki rozmieszczeniu wybranych klastrów geograficznych za pomocą selektora regionu *Gamesparks*.

5. Premium SLA

Zapewnia indywidualnie dostosowane poziomy wsparcia dla konkretnej firmy w zależności od problemu, z poziomami usług wsparcia zarówno dla twórców gry jak i dla jej graczy. [40][43]

Przykładami gier stworzonych przy pomocy Gamesparks są:

- **7 Days to Die** - gra komputerowa z gatunku survival horror, stworzona przez studio The Fun Pimps Entertainment.



- **Pac-Man Pop** – produkcji BANDAI NAMCO, polecane przez App Store.



- **Lara Croft: Relic Run** – dostępne na platformy Android, iOS i Windows Phone, gra stworzona w Unity 3D, ponad 10 milionów pobrań na Google Play i AppStore.



- **Rocky, Mobile Game** – gra mobilna, w której wcielamy się w legendarnego Rockiego, polecane przez App Store.



5. Wykonanie prototypu gry

5.1. Organizacja projektu

Podstawowym krokiem w celu wykonania badań w pracy dyplomowej jest określenie kryteriów oceny każdego zestawu omawianych usług.

W tej pracy oceniane są:

- dostęp do informacji na temat danych zestawów funkcji sieciowych oraz jakość tych informacji,
- ilość zapewnianych usług,
- trudność w implementacji poszczególnych funkcjonalności, prostota,
- panel sterowania – ilość opcji i ich przydatność,
- podatność na błędy i opóźnienia,
- pozostałe cechy.

Pierwszym krokiem w celu wykonania powyższych badań było przedstawienie funkcjonalności omawianych usług sieciowych. Krokiem następnym jest implementacja zawartości gry komputerowej.

Rzeczą podstawową każdej gry jest arena zmagania graczy potocznie zwana *mapą*. Twórcy gier mają wybór, co do sposobu w jaki taką mapę zdobyć. Mogą stworzyć ją samodzielnie przy pomocy programów graficznych lub silnika Unity 3D, albo mogą skorzystać z gotowych map dostępnych w sklepie Unity. Jako że Asset Store będzie przedstawiony w jednym z następnych podrozdziałów, służąc do znalezienia innych składowych wykonywanego projektu, mapa zostanie stworzona przy pomocy Unity 3D. Po krótkim omówieniu, we wspomnianym sklepie Unity zostaną wybrane rzeczy przydatne w grze MOBA takie jak struktury czy niebo/tło.

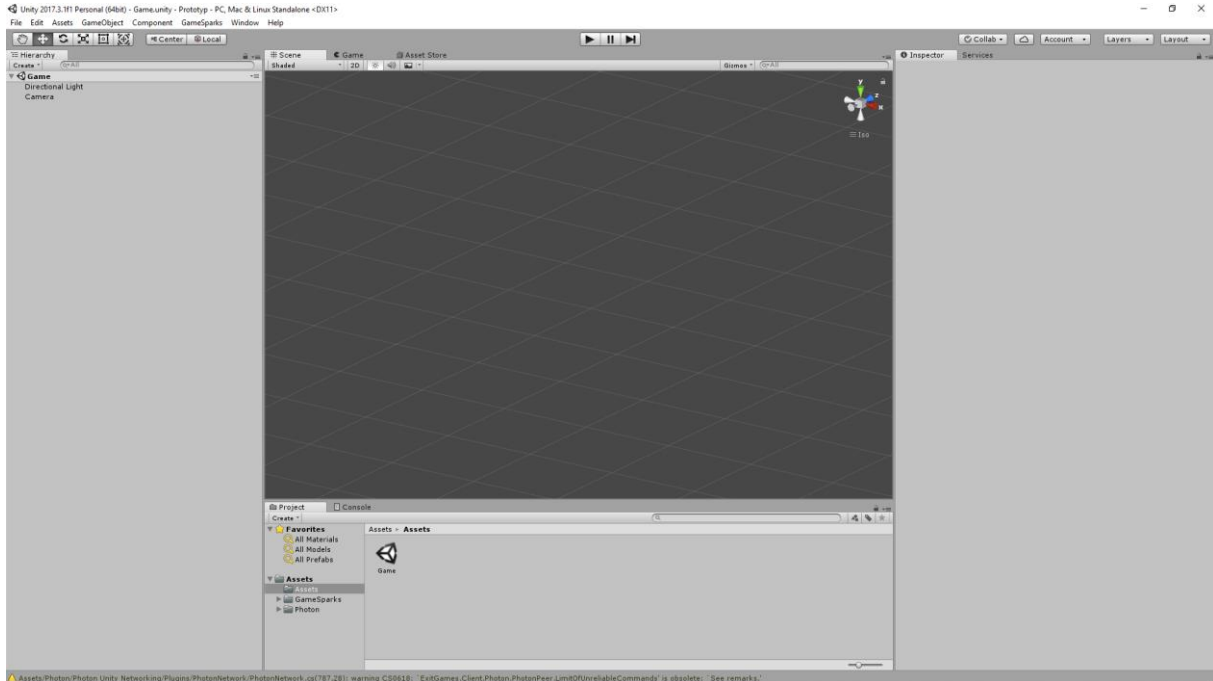
Kolejnym krokiem będzie wybór postaci, implementacja jej ruchu i ataku oraz mechanizm znajdowania ścieżek na mapie. Na potrzeby prototypu nie jest wymagana zbyt duża ilość zawartości grywalnej w projekcie, dlatego atak ograniczy się do implementacji metody zwracającej komunikat w konsoli. Jest to pierwsza z możliwości przyszłego rozwoju gry.

Dalej zostaną zaimplementowane wybrane funkcjonalności usług sieciowych omówionych w Rozdziale 4 tej pracy dyplomowej. Z racji tego, że sposób łączenia gracza oraz połączenia z siecią są do siebie zbliżone, zaimplementowany zostanie zestaw usług *Unity Networking (UNet)*, a pozostałe usługi (*PUN, Gamesparks*) zostaną na jego podstawie omówione.

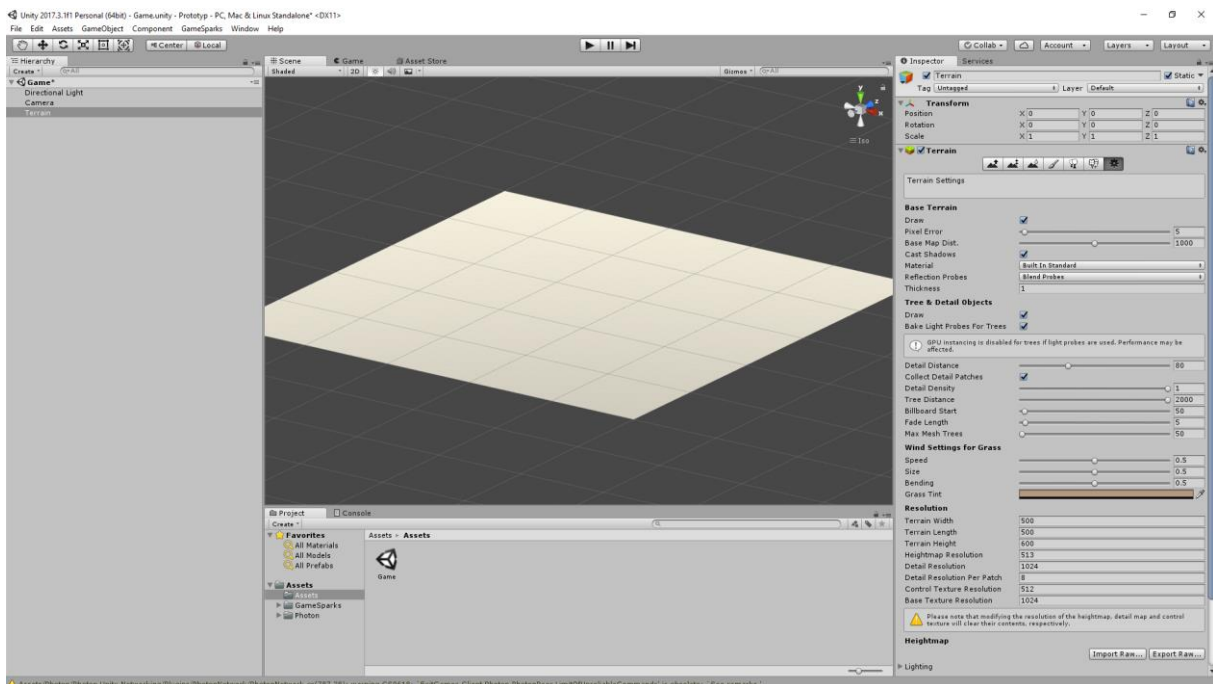
5.2. Implementacja funkcjonalności

❖ Mapa, opcja *Terrain*, Asset Store

Podstawowym krokiem przy tworzeniu oprogramowania za pomocą Unity 3D jest utworzenie projektu. Gdy już twórca otworzy Unity, zaloguje się do swojego konta i nada nowemu projektowi nazwę, ukaże się przed nim główny ekran silnika gry. Wygląda on mniej więcej tak:



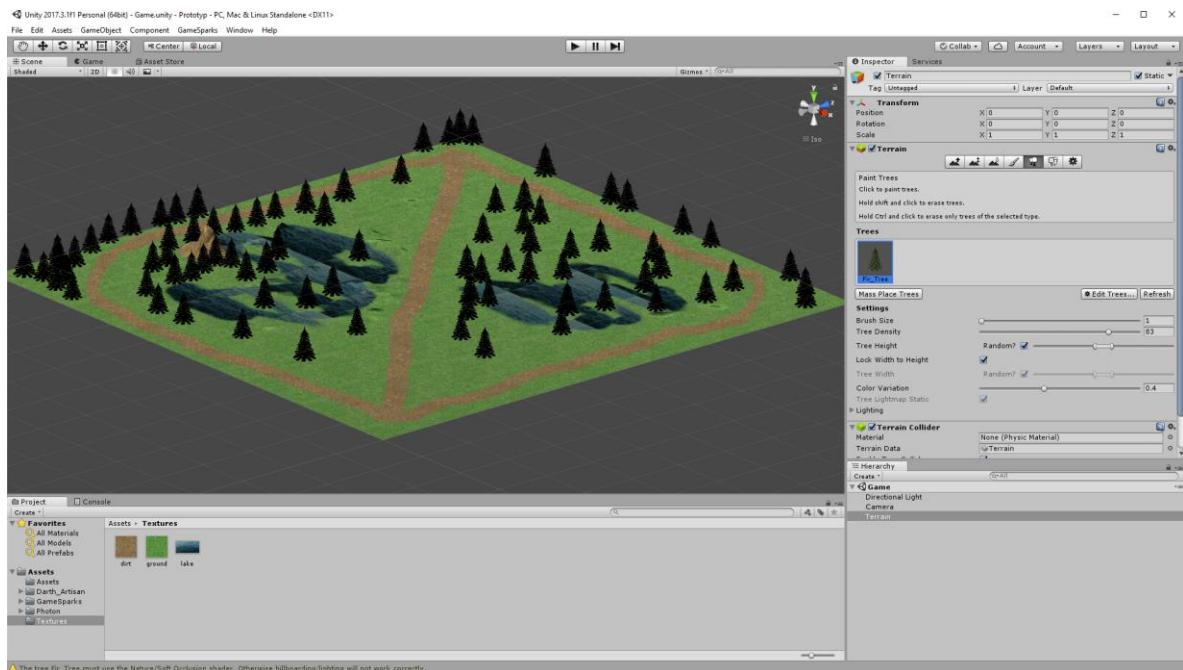
W celu utworzenia mapy gry należy w górnym pasku opcji kliknąć **GameObject**, następnie rozwinąć listę opcji **3D Object** i wybrać **Terrain**. W aktualnie tworzonej scenie utworzony zostanie kwadrat, który można dowolnie edytować.



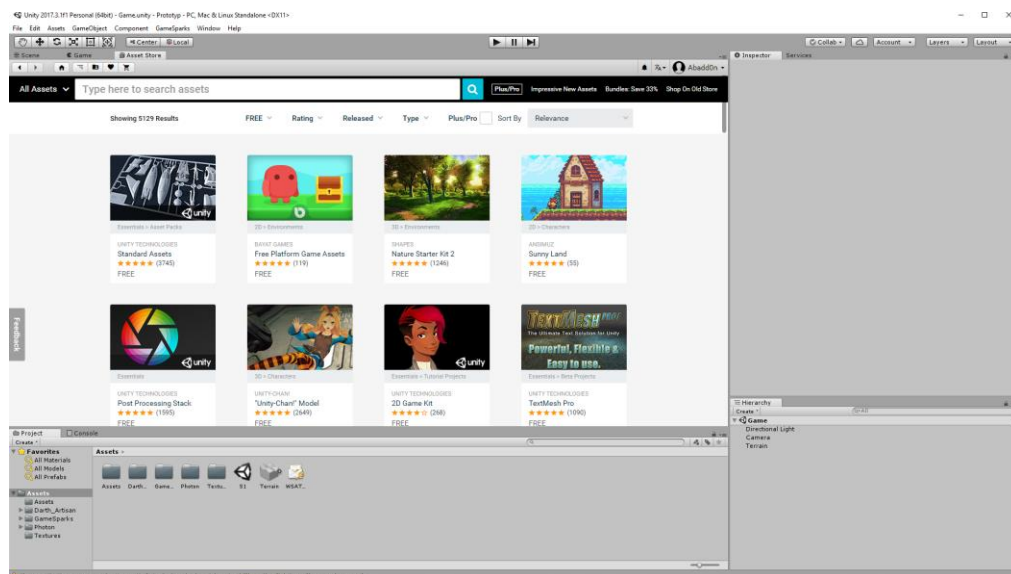
Po prawej stronie w **inspektorze** dostępny jest komponent *Terrain*. Posiada on kilka przydatnych opcji edycji stworzonej płaszczyzny, m.in. są to:

- Ustawienia terenu.
- Wznoszenie/obniżanie terenu.
- Malowanie tekstury i/lub drzew.

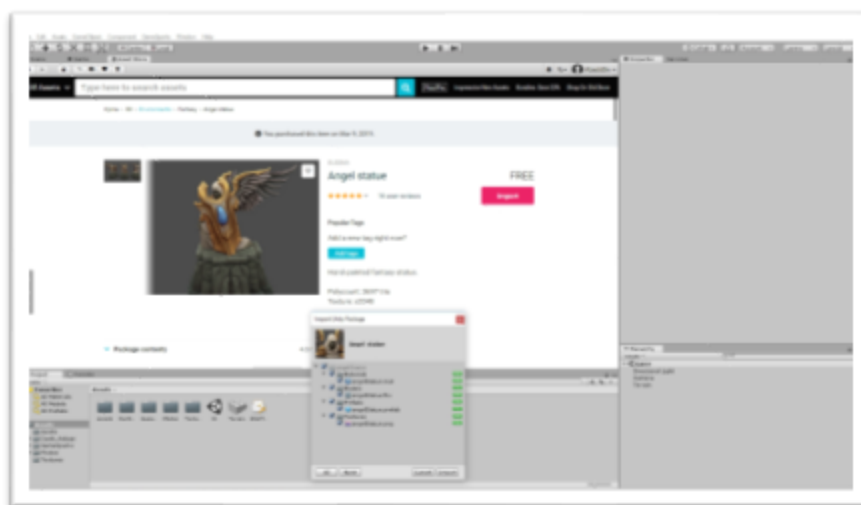
Powyższe opcje pomogą w wypełnieniu mapy szczegółami. Za ich pomocą należy wymodelować kontury mapy, jej wzniesienia i zagłębienia, nadać jej teksturę i wypełnić ją drzewami lub innymi szczegółami, które posłużą za jej obramowanie. Obszerny opis wszystkich opcji komponentu *Terrain* znajduje się w dokumentacji do Unity 3D na stronie producenta silnika gier. Po wymodelowaniu mapa wygląda następująco:



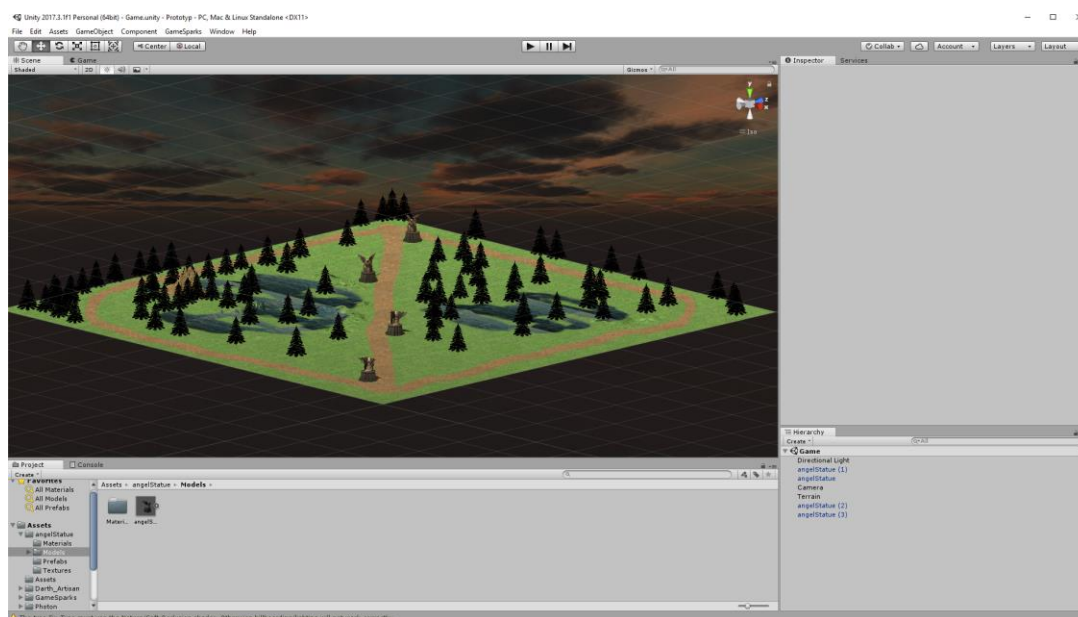
Gdy teren zostanie już wymodelowany według preferencji tworzącego grę następuje kolejny krok tworzenia mapy. Klikając kombinację klawiszy **ctrl + 9** użytkownikowi Unity ukaże się sklep zasobów silnika zwany Asset Store'm:



Asset Store to wbudowany w Unity 3D magazyn gotowych do użycia elementów składowych gry. W sklepie tym, tworzący grę mogą znaleźć wiele przydatnych rzeczy zaczynając od zestawów tekstur, a kończąc na bardzo złożonych modelach postaci z podpiętymi animacjami i sterowaniem. Są tu rzeczy płatne i darmowe, pojedyncze elementy jak i całe zestawy grafik. Skrypty, ustawienia kamery, światła, wzorce map. Wszystko to można znaleźć w jednym miejscu. Asset Store daje możliwość wyszukania potrzebnego elementu i pozwala zdefiniować kilka parametrów wyszukiwania poprzez filtry takie jak nazwa produktu, jego cena, rozmiar, data wydania lub aktualizacji. Aby pobrać wybrany element należy kliknąć w jego nazwę, a następnie w widoku, który ukaże się korzystającemu z Unity, należy kliknąć **Download** w celu pobrania i **Import** w celu zaimportowania produktu do projektu. W tej drugiej opcji Unity 3D da twórce gry możliwość wyboru, które elementy z listy wszystkich elementów mają zostać zaimportowane do projektu. Należy ponownie kliknąć **Import**:



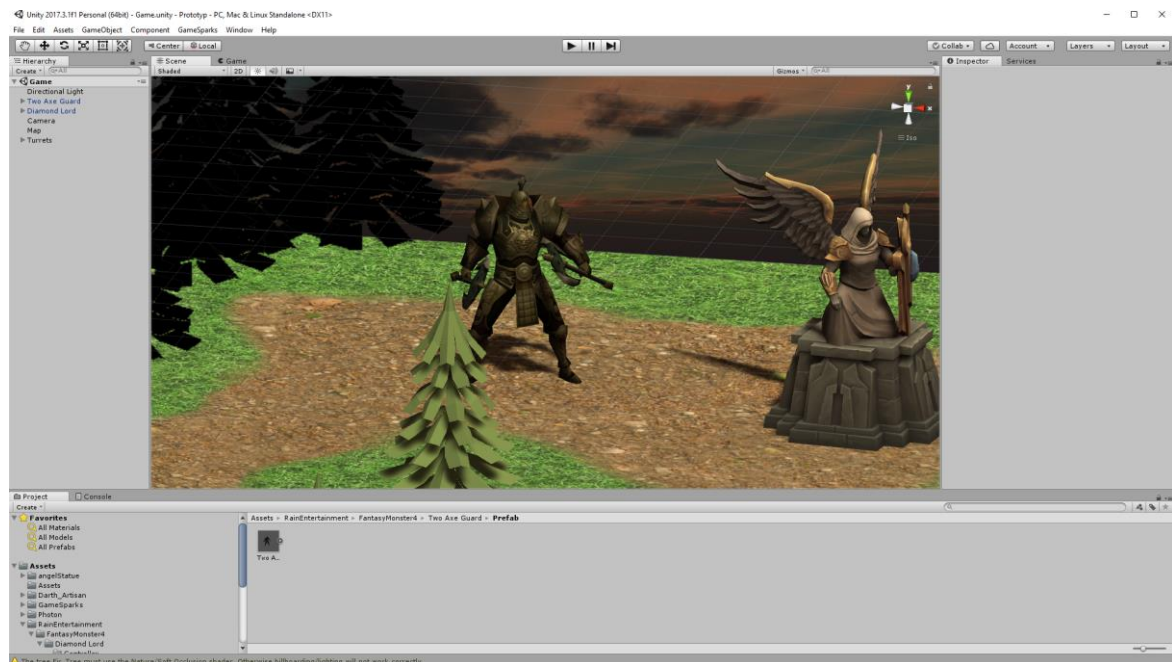
Jako struktury w projekcie gry skorzystano z posągu anioła *Angel statue* dostępnego w Unity Asset Store za darmo. Jako tekstury tła skorzystano z zestawu tekstur *Sky5X One*.



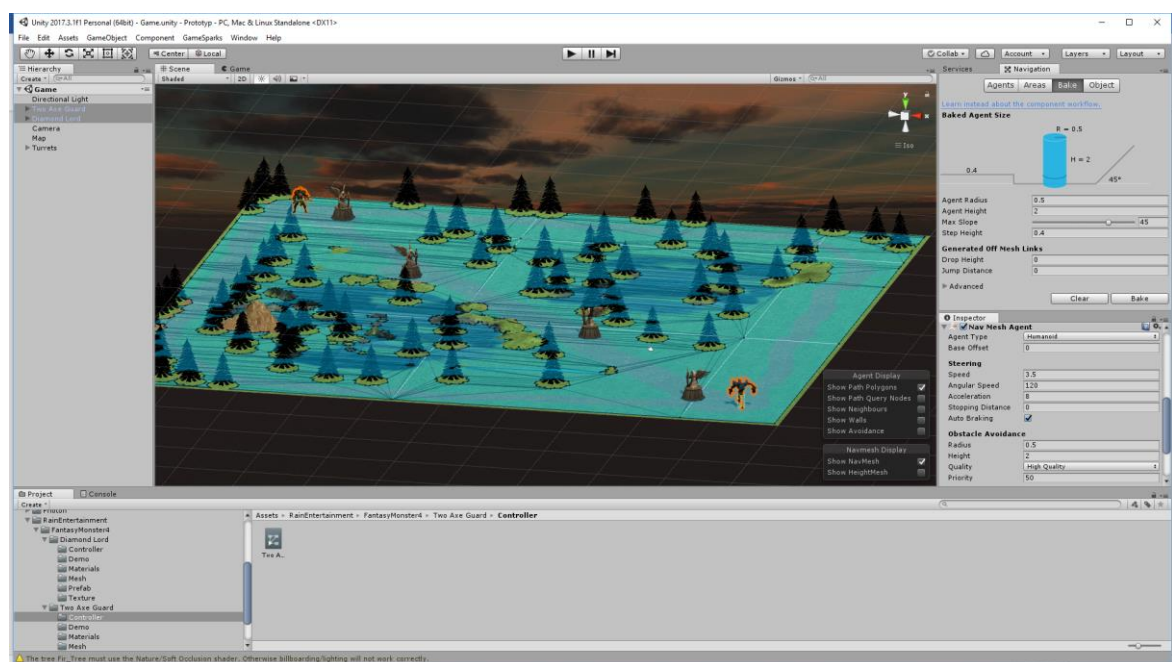
Rysunek 28. Mapa po zaimportowaniu potrzebnych elementów

❖ Postać – pathfinding, kamera, sterowanie

W celu stworzenia postaci do gry również skorzystano ze sklepu zasobów. Wybrano *Diamond Lord* produkcji *RainEntertainment*.



Zaimportowane postacie umieszczono w miejscach, które posłużą graczom do rozpoczęcia rozgrywki. Teraz należy ustalić po jakim terenie mogą się one poruszać. Służy do tego znajdowanie ścieżek, bardziej znane ze swojej angielskiej nazwy czyli pathfinding. Należy zabezpieczyć teren, po którym gracz nie powinien móc się poruszać jak np. woda czy drzewa. O ile te ostatnie posiadają odpowiedni **Collider**, który blokuje graczowi tę możliwość tak w przypadku zaimportowanej tekstury wody takiej opcji z automatu nie ma.



W celu zarządzania pathfindingiem użytkownik powinien wejść w opcję **Window**, a następnie **Navigation**. W powyższym oknie należy wybrać zakładkę **Bake** i kliknąć ponownie

przycisk **Bake**. Po wykonanej akcji teren, po którym porusza się **agent** zostanie podświetlony na niebiesko. Agentem jest wszystko, co twórca oznaczy poprzez opcję **Component** -> **Navigation** -> **Nav Mesh Component**. Odpowiada on za obiekt, który jest w stanie się poruszać po mapie. Twórca gry ma możliwość dostosowania kilku składających się na ruch funkcji, jak prędkość poruszania się lub rozmiar collider'a. Gdy użytkownik Unity skończy oznaczać teren poruszania się wg własnych preferencji oraz gdy ustawi odpowiedni rozmiar każdego agenta ten etap tworzenia gry zostanie zakończony. Efekty będą widoczne po skonfigurowaniu sterowania postaci.

Następnym krokiem tworzenia gry jest konfiguracja kamery. Domyślnie obiekt kamery jest już umieszczony w projekcie wraz z jego utworzeniem. Niestety, jest to kamera unieruchomiona. Oznacza to, że gracz nie ma nad nią żadnej kontroli. By to naprawić trzeba napisać odpowiedni skrypt w C#. Należy kliknąć w projekcie prawym przyciskiem myszy w polu **Assets**, dalej **Create** i z listy wybrać **C# Script**. Należy nadać mu nazwę wedle preferencji użytkownika a następnie ustawić w nim kamerę. Poniższy listing kodu C# zarządzający ruchem kamery, plik został nazwany *CameraController.cs*:

```
float speed = 100.0f;
//int boundary = 1;
//int width;
//int height;

void Start()
{
    //width = Screen.width;
    //height = Screen.height;
}

void Update()
{
    if (Input.GetMouseButton(0))
    {
        if (Input.GetAxis("Mouse X") < 0)
        {
            transform.position -= new Vector3(Input.GetAxisRaw("Mouse X") * Time.deltaTime * speed,
            0.0f, Input.GetAxisRaw("Mouse Y") * Time.deltaTime * speed);
        }

        else if (Input.GetAxis("Mouse X") > 0)
        {
            transform.position -= new Vector3(Input.GetAxisRaw("Mouse X") * Time.deltaTime * speed,
            0.0f, Input.GetAxisRaw("Mouse Y") * Time.deltaTime * speed);
        }
    }
}
```

Zmienna *speed* typu *float* odpowiadają za prędkość przewijania kamery. Cała mechanika przewijania opiera się o kliknięcie i przesunięcie myszy przytrzymując lewy jej przycisk (*GetMouseButton(0)*). Do określenia pozycji przewijania wykorzystuje się wektory położenia względem osi X i Y. Gotowy skrypt należy zapisać i przenieść myszką do obiektu kamery. Jest to przykład kontroli kamery bez zablokowania możliwości przewinięcia za mapę, gdyż nie jest to wymagane w owej pracy. Jest to jednak jedna z pierwszych rzeczy, które można by dodać w grze w przyszłości. Kolejnym jest możliwość dodania ruchu kamery poprzez przesunięcie kursora myszy na krawędź okna aplikacji. W omawianym pliku C# jest już napisany taki skrypt, w razie potrzeby można go odznaczyć komentarzami pamiętając o wyłączeniu przedstawionego rozwiązania.

Kolejnym krokiem w tworzeniu gry jest sterowanie ruchem bohatera. Ponownie należy utworzyć skrypt, tym razem będzie on nazywał się *Player.cs*. Funkcje, które musi spełniać ten skrypt to ruch postacią na mapie oraz strzał w kierunku przeciwnika gdy zostanie on kliknięty przez gracza. Duża część pracy została zaoszczędzona dzięki mechanizmowi znajdowania ścieżek. Dzięki jego obecności do napisania sterowania postacią wystarczy znajomość instrukcji *if...else...* oraz zmiennej typu *Boolean*.

Poniżej listing kodu funkcji *Update ()*:

```
void Update()
{
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;

    if (Input.GetButtonDown("Fire2"))
    {
        if (Physics.Raycast(ray, out hit, 200))
        {
            if (hit.collider.CompareTag("Enemy"))
            {
                targetedEnemy = hit.transform;
                enemyClicked = true;
            }
            else
            {
                walking = true;
                enemyClicked = false;
                navAgent.destination = hit.point;
                navAgent.Resume();
            }
        }
    }

    if (enemyClicked)
    {
        MoveAndShot();
    }

    if (navAgent.remainingDistance <= navAgent.stoppingDistance)
    {
        walking = false;
    }
    else
    {
        walking = true;
    }

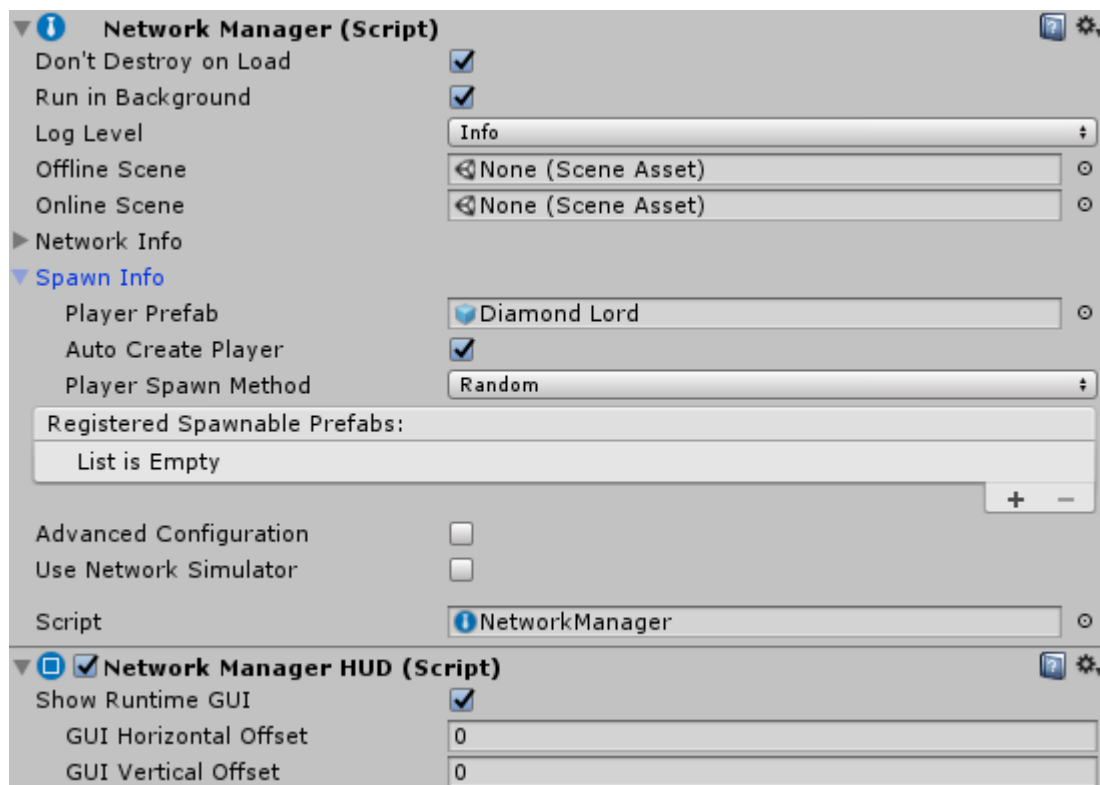
    //anim.SetBool("IsWalking", walking);
}
```

W powyższej funkcji pierwsza instrukcja *if...* odpowiada za sprawdzenie czy został wciśnięty prawy przycisk myszy. Jeśli wartość sprawdzenia jest prawdą w drugiej instrukcji dochodzi do sprawdzenia możliwości poruszenia się po planszy w zakresie do 200 jednostek. Jeżeli to również ma wartość *true* to w trzeciej instrukcji program sprawdza co zostało kliknięte. Jeśli był to przeciwnik (collider oznaczony jako *Enemy*) funkcja nakazuje postaci strzelać do wroga, jeśli nie jest to przeciwnik to postać poruszy się w wyznaczone miejsce.

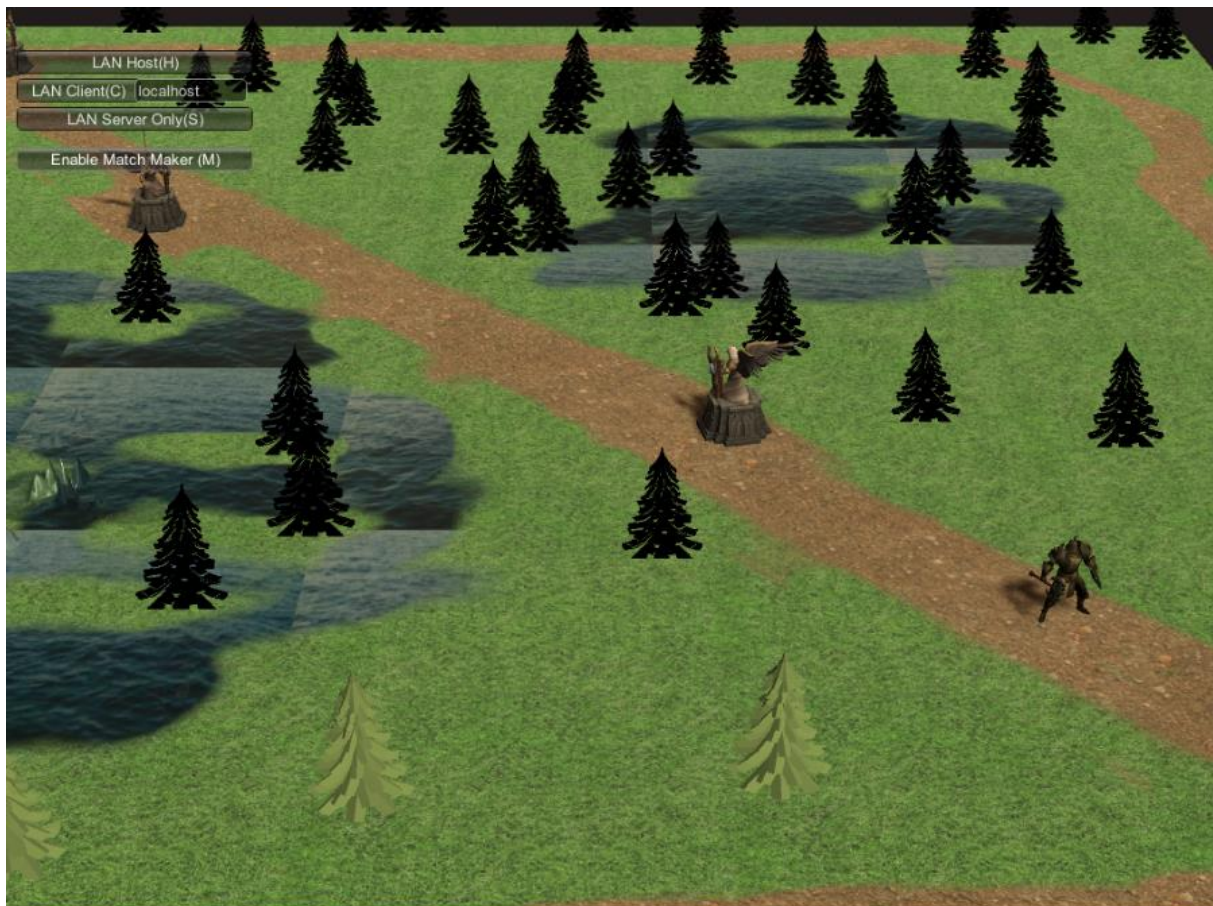
❖ Funkcje sieciowe

W tym rozdziale zostaną zaimplementowane wybrane rodzaje usług sieciowych. Główną funkcjonalnością tworzonej gry jest jej menadżer sieciowy. To on będzie zarządzał połączeniem oraz wszystkim co nastąpi dalej.

W oknie **Hierarchy** należy kliknąć prawym przyciskiem myszy i utworzyć pusty obiekt (**Create Empty**) a następnie nadać mu nazwę *NetworkManager*. Następnie z opcji **Component** -> **Network** w górnym pasku Unity należy wybrać komponent o nazwie **NetworkManager**. Kolejnym krokiem jest stworzenie interfejsu dostępu do tworzonej aplikacji po sieci. W tym celu należy skorzystać z **Component** -> **Network** -> **NetworkManagerHUD**. Kolejnym etapem tworzenia aplikacji sieciowej jest nadanie identyfikatora obiektom klienta, w przypadku tworzonej gry są to bohaterowie gdyż każdy z graczy będzie kontrolował tylko jednego z nich. Do postaci należy dołączyć **Component** -> **Network** -> **NetworkIdentity**, następnie w zakładce *NetworkIdentity* należy zaznaczyć opcję *Local Player Authority*. Dzięki temu gra rozpoznaje czy wybrany bohater jest sterowany przez gracza (zaznaczona powyżej opcja) czy jest sterowanym komputerowo botem (druga możliwa opcja – *Server Only*). Następnie twórca gry musi przenieść postać gracza z dodanymi w procesie tworzenia komponentami do osobnego folderu. Dzięki temu model postaci z dołączonymi skryptami sieciowymi będzie mógł posłużyć za wzorzec i być powielanym w sieci przy dołączaniu każdego z graczy do gry.

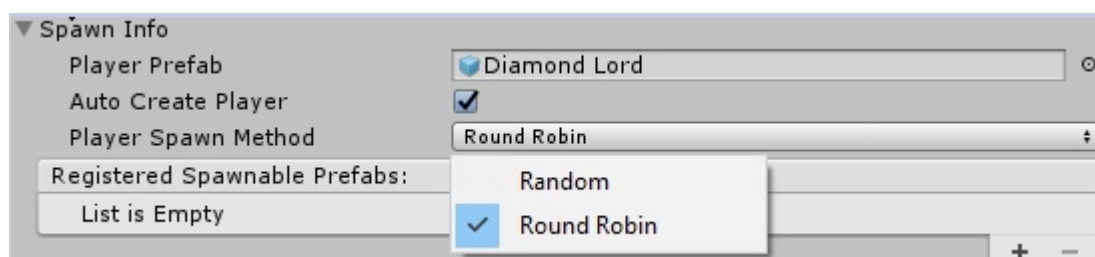


Rysunek 29. Wyeksportowaną do osobnego folderu postać należy przeciągnąć na miejsce opcji Player Prefab w Spawn Info



Rysunek 30. W lewym górnym rogu widoczny komponent *NetworkManagerHUD*. Pozwoli on graczom łączyć się z rozgrywką.

Gdy postać gracza została dodana do menadżera sieciowego przyszedł czas na oznaczenie miejsc, w których gracze będą pojawiali się w grze. Miejsca te zostaną nazwane *SpawnA* oraz *SpawnB*. W żargonie graczy punkty te noszą nazwę *punktów spawnu* lub *punktów odrodzenia*. Do ich stworzenia zostaną wykorzystane dwa puste obiekty w oknie *Hierarchy*. Po utworzeniu tych obiektów należy nadać im odpowiednie współrzędne wg preferencji. W tej pracy są to przeciwległe końce mapy, jest to charakterystyczne rozmieszczenie graczy w grze *MOBA*. Do poinformowania serwera o miejscach rozpoczęcia rozgrywki wg powyższych wytycznych posłuży komponent *NetworkStartPosition*. Należy dodać go do obu obiektów, *SpawnA* i *SpawnB*. Dzięki temu, że w grze jest więcej niż jeden punkt rozpoczęcia, *NetworkManager* pozwoli osobie tworzącej grę wybrać sposób pojawiania się postaci w danym punkcie spawnu. Do wyboru dostępne jest losowe wybranie punktu odrodzenia lub mogą być one obsługiwane na zmianę.



Rysunek 31. Dwie dostępne opcje rozpoczęcia rozgrywki z zaznaczoną opcją na przemian

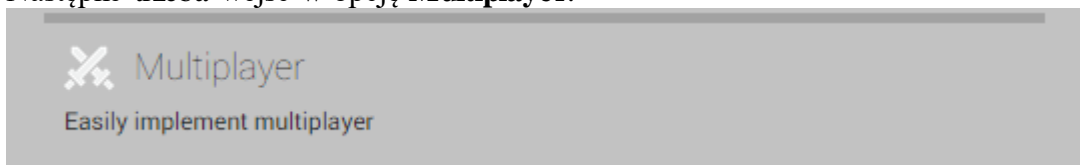
Tak utworzona gra nie może jeszcze zostać buildowana do poziomu aplikacji okienkowej. Wszystko przez to, że nie zostało wprowadzone zabezpieczenie związane ze śledzeniem ruchu postaci w grze lokalnie. Gdyby użytkownik zbudował grę w tym momencie na każdym z kont klientów serwer utworzyłby po dwie instancje gracza i każdy z graczy mógłby sterować nimi co uniemożliwiłoby rozgrywkę. Należy zatem poprawić skrypt odpowiadający za ruch postaci. Pierwsze co należy zrobić to zaimportować bibliotekę *UnityEngine.Networking*. Zawiera ona kilka pól, które okażą się bardzo przydatne, m.in. jednego z nich należy użyć w celu zabezpieczenia problemu śledzenia ruchu po sieci.

```
if (!isLocalPlayer)
{
    return;
}
```

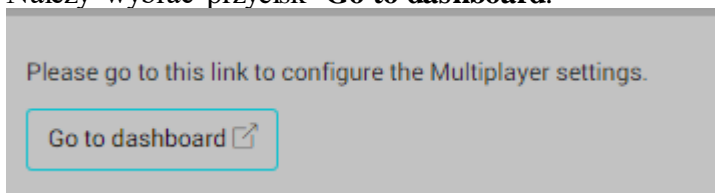
Dzięki temu, że powyższa zmienna typu *bool* o nazwie *isLocalPlayer* zwraca wartość *true* gdy obiekt reprezentuje gracza na maszynie lokalnej, powyższy fragment kodu wystarczy by sprawdzić czy dany gracz kontroluje tylko jedną z postaci. Dzięki temu klient rozpozna, którą postacią będzie sterował gracz i nie pozwoli poruszyć mu inną. Teraz wystarczy dodać komponent, który zsynchronizuje ruch postaci w kliencie gry i gotowe, problem rozwiązany. Posłuży do tego komponent *NetworkTransform* dodany do obiektów, które piszący grę chce zsynchronizować. Z racji tego, że w pisanej grze obiektem tym są postaci graczy i że we wcześniejszym kroku postać została zapisana jako wzorzec, wystarczy dodać wyżej wymieniony komponent w zaledwie jednym miejscu – do postaci.

Tak przygotowana gra może zostać już umieszczona na serwerze sieciowym. W UNet odbywa się to w następujący sposób:

1. Należy w liście opcji **Window** otworzyć okno **Services**.
2. Następnie trzeba wejść w opcję **Multiplayer**.

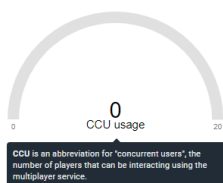


3. Jeśli użytkownik nie jest zalogowany do konta Unity 3D w przeglądarce lub jeśli użytkownik nie zaakceptował wcześniej zasad licencyjnych *EULA*, zostanie on poproszony o zapoznanie się z zasadami korzystania z usługi i zaakceptowanie ich.
4. Należy wybrać przycisk **Go to dashboard**.



5. Użytkownik zostanie przeniesiony do panelu zarządzania aplikacją. W tym miejscu może określić liczbę jednoczesnych użytkowników korzystających z jego gry, może sprawdzić i dostosować liczbę aktualnie korzystających osób z jego innych projektów, może dostosować maksymalną liczbę osób w jednym pokoju sieciowym i ustawić wartość pasma dla pojedynczego klienta.

UNET ID: 7802152
 Last updated: a day ago (March 9, 2019 11:09:47 +0100)

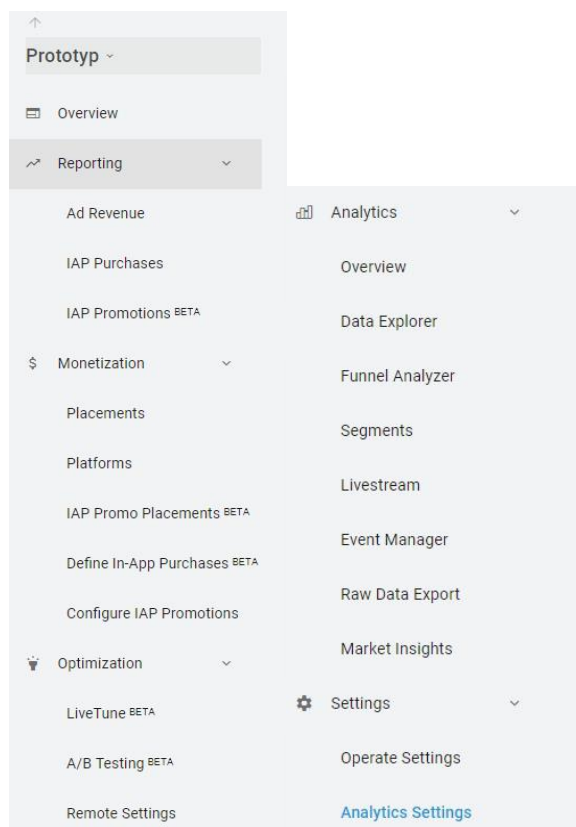


Global CCU Limit	20	CCUs
Currently used by all projects	0	CCUs
Used by this project	0	CCUs
Bandwidth per client (can be changed in live mode)		
	4608	B/s
Max players per room		
	20	players
Save		

Rysunek 32. Zdjęcie przedstawiające panel zarządzania aplikacją w Unity Networking

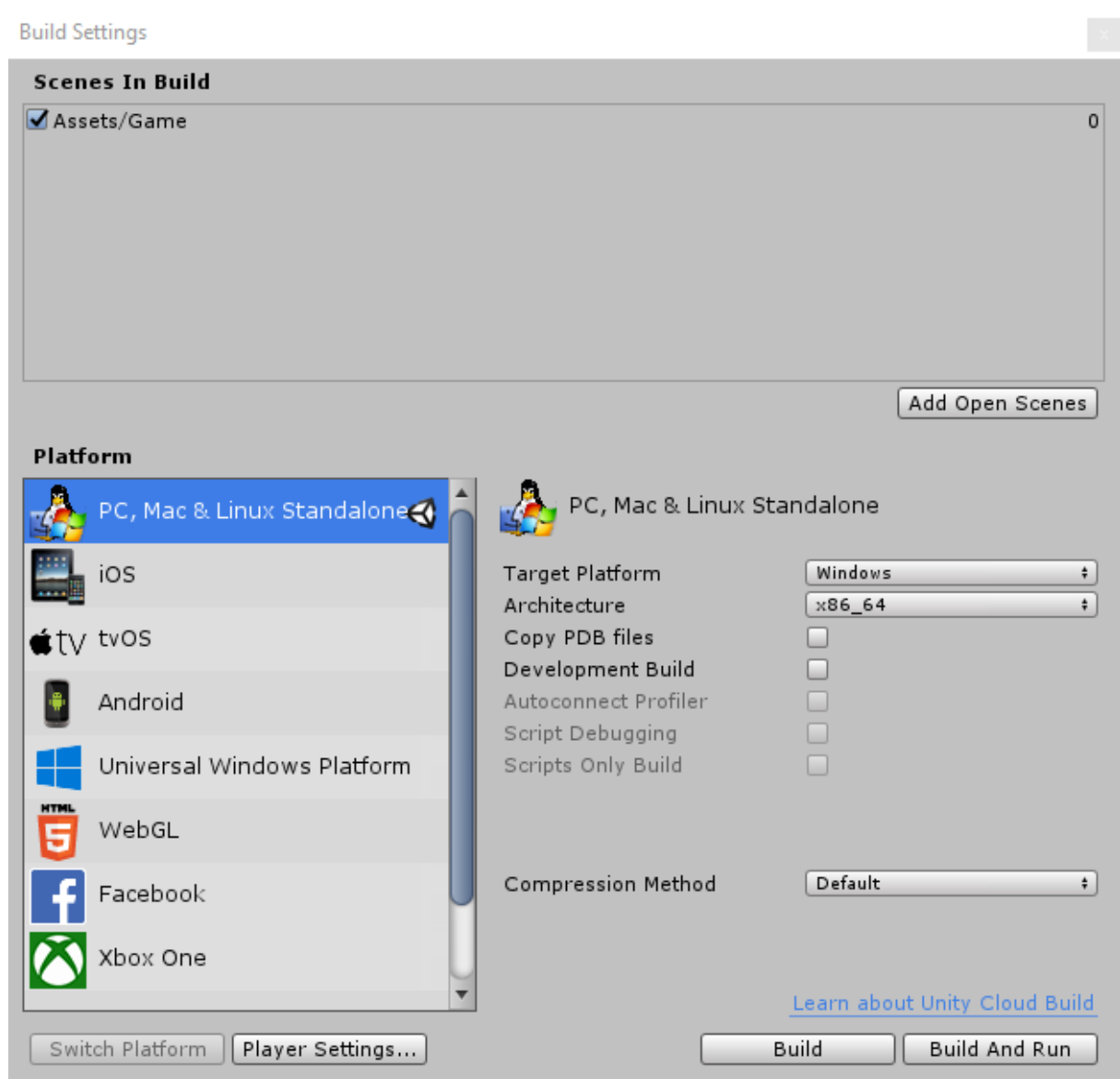
We wspomnianym panelu osoba tworząca grę może również przejść do ogólnego widoku wszystkich swoich projektów. Może również:

- Włączyć i zarządzać narzędziem reklam i monetyzacji,
- Skorzystać z narzędzia do analizy, gdzie w formie tabel i wykresów ma przedstawione informacje na temat sesji, przychodów i graczy, ilu ich jest miesięcznie, tygodniowo, dziennie oraz ilu jest nowych,
- Przeprowadzenia transmisji wideo na żywo np. z prezentacji nowych funkcjonalności.
- Informacje na temat ilości dokonanych przez użytkowników mikropłatności w aplikacji,
- Narzędzie do przeprowadzenia testów Alfa oraz Beta,
- Konfigurację opcji takich jak nazwa projektu i jego identyfikator oraz konfigurację narzędzia do analizy danych gry i rynku.



Rysunek 33. Na powyższych obrazkach widać funkcje, które zapewnia programiście Unity Networking.

Gdy liczba użytkowników zostanie ustawiona programista może zbudować swój projekt.

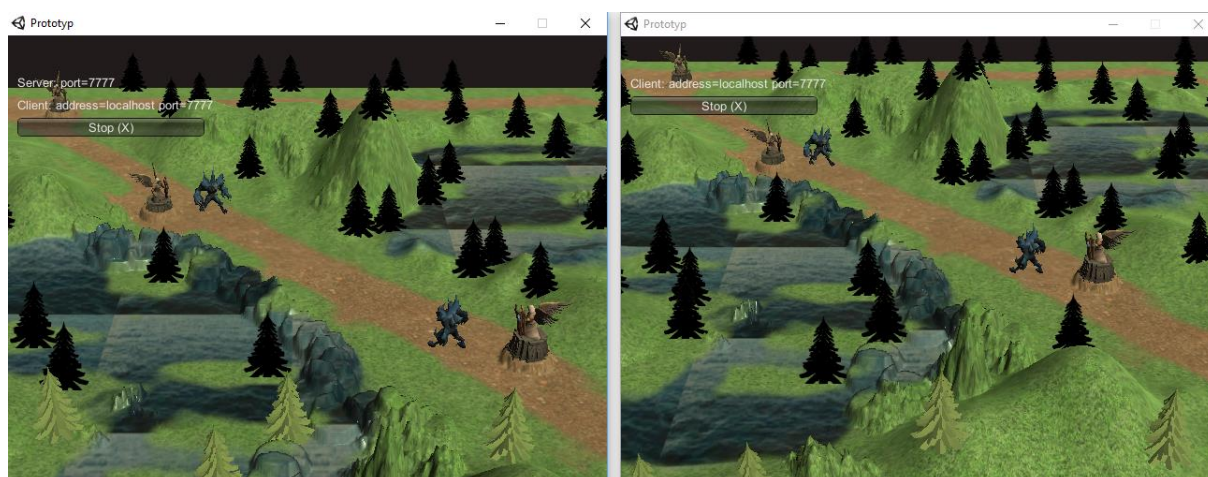


Rysunek 34. W tym momencie można zaobserwować jak wiele korzyści może wynikać z korzystania z silnika gier Unity 3D. Wystarczy zaimportować wybrane sceny gry do okna Scenes In Build oraz wybrać platformę, na którą gra ma zostać stworzona. Dzięki temu programista jest w stanie zaoszczędzić bardzo dużo cennego czasu, który może w przyszłości przeznaczyć na rozbudowę grywalnej zawartości swojego projektu.

Po kliknięciu **Build** Unity zajmie się zbudowaniem stworzonej gry, nieco wcześniej użytkownik silnika zostanie poproszony o wskazanie miejsca/folderu/katalogu, w którym ma odbyć się powyższy proces. Po chwili, którą silnik gier przeznaczy na budowę stworzonego projektu, należy sprawdzić czy wszystko działa jak należy. Proces ten potrwa tym dłużej im więcej zawartości jest w grze. Najprostszym sposobem sprawdzenia czy powyższe operacje się powiodły jest uruchomienie dwóch klientów aplikacji w oparciu o zaimplementowaną metodę *matchmakingu*. W jednym z klientów gracz A tworzy pokój gry, a w drugim gracz B dołącza do niego. Jeśli na jednym i drugim ekranie widać dwie niezależnie poruszające się postacie oznacza to sukces.



Rysunek 35. Klient A (zdjęcie po lewej stronie) tworzący pokój gry o nazwie Abadd0n oraz Klient B (zdjęcie po prawej stronie) widzący stworzony pokój. Na zdjęciach poniżej widać, że połączenie klientów zakończyło się powodzeniem. Sukces!



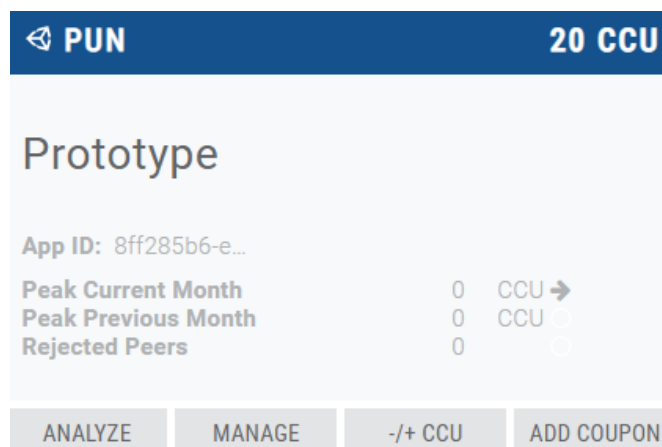
Wadą powyższego rozwiązania jest potrzeba resetowania połączenia z poziomu Unity i panelu sterowania *UNet* w momencie gdy straci się połączenie z serwerem. Podczas pisania tej pracy nastąpiło to co najmniej kilkakrotnie. Kolejnym minusem jest fakt, iż twórcy Unity 3D powoli wycofują się z *UNet* i pracują nad nowym rozwiązaniem sieciowym, gdyż jak twierdzą „nie spełnia ono wymagań jakie przed Unity stawiają twórcy nowych gier”. Finalnie usługa ma zostać wyłączona w roku 2022. W przypadku gier komercyjnych nie będzie opłacało się firmom zajmującym się produkcją gier tworzenia ich oprogramowania na silniku, który zaraz zostanie wyłączony. Niemniej jednak *UNet* można zakwalifikować jako bardzo dobre narzędzie do rozpoczęcia swojej przygody z usługami sieciowymi oraz do napisania pierwszej gry online.

Co się tyczy pozostałych dwóch zestawów usług to składnia pisania programu jest bardzo podobna lub wręcz taka sama. Jest to spowodowane tym, że skrypty zarządzające siecią różnią się od siebie tylko bibliotekami klas, z których czerpią. W przypadku *UNet* były to głównie klasy *NetworkBehavior* i *NetworkManager*. Więcej informacji na temat wspomnianych klas można znaleźć w dokumentacji Unity 3D.

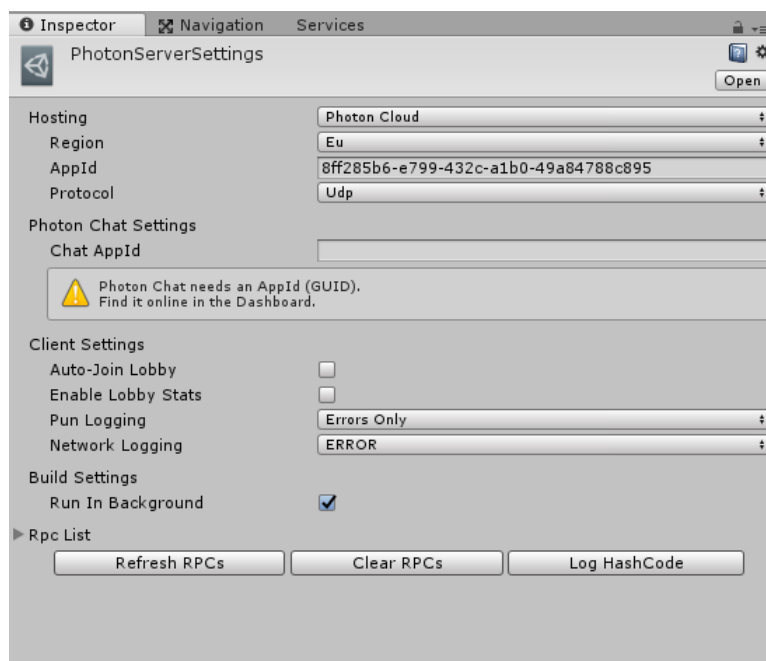
W przypadku *PUN* klasą, której programista będzie używał najczęściej do pisania swoich skryptów jest *PhotonNetwork*. Photon nie ma gotowego ekranu HUD, który posłużyłby za narzędzie matchmakingu stąd należy stworzyć taki osobno w Unity. Odbывается to poprzez **GameObject -> UI -> Canvas** oraz poprzez stworzenie i dodanie odpowiedniego skryptu. Przykładem metody dobierania przeciwników jest *PhotonNetwork.JoinRandomRoom()*; pozwalające graczowi na dołączenie do losowego pokoju i *JoinRoom(roomName)*; pozwalająca mu dołączyć do pokoju o określonej przez niego nazwie.

W *Gamesparks* w celu zarządzania dobieraniem graczy użytkownik jest zmuszony do implementacji w grze systemu logowania. Posiada on jednak narzędzie, które pozwala mu na test połączenia z grą na serwerze w postaci *GameSparkSettings* -> *Test Configuration*. W przypadku *Photon Unity Networking* i *Gamesparks* łączenie z grą na serwerze odbywa się dokładnie tak samo jak miało to miejsce w przypadku *UNet*. Należy pamiętać jednak, że domyślnie w Unity nie ma paczek z zestawem narzędzi *SDK* ani *PUN* ani *Gamesparks*. Należy je wcześniej pobrać i zaimportować do projektu tak samo jak miało to miejsce w przypadku pod rozdziału z omawianym *AssetStorem*.

W celu uzyskania połączenia z *PUN* należy założyć konto na stronie *photonengine.com*, następnie załogować się do niego. W prawym górnym rogu należy kliknąć na ikonkę konta. Menu, które się rozwinięło ma trzy opcje, należy skorzystać z pierwszej z nich (*Your Application*). W oknie, które pokaże się po wejściu w wyżej wymienioną opcję użytkownik ma możliwość utworzenia swojej gry. Dzięki temu zostanie mu nadany identyfikator, który posłuży do synchronizacji gry z serwerem.



Rysunek 36. Na powyższym zdjęciu widać utworzoną grę na serwerze. Poniżej w Inspektorze poprawna konfiguracja gry z serwerem



Analogicznie jest w przypadku *Gamesparks*. Użytkownik tworzy konto w witrynie *gamesparks.com* i loguje się do swojego konta. W oknie, które się ukaże należy wybrać opcję *Configure Your Games* dzięki czemu tworzący grę zostanie przeniesiony do panelu sterowania, konkretnie do okna utworzenia swojej gry na serwerze. Po podaniu nazwy projektu oraz jego opisu, programista będzie miał możliwość zaznaczenia funkcjonalności, które uważa za przydatne w swoim projekcie. Innymi słowy programista nie musi korzystać ze wszystkich dobrodziejstw *Gamesparks* i ma opcję wyłączenia niechcianych funkcjonalności już na starcie. W tym miejscu użytkownik również podaj platformy z jakim chce zintegrować swój produkt oraz ma możliwość zdefiniowania do sześciu walut w grze:

Platform Features

What GameSparks platform features would you like to enable?

☒ ⚡ Events Create User Events and use them to define custom data structures that you want to pass into the platform.

☒ 📊 Leaderboards Build traditional in-game Leaderboards to rank players or Leaderboards to control when Achievements are awarded or Challenges are won.

☒ 👥 Teams Define player-based Teams and use them for team-based Leaderboards, team Achievements, and to build social features and cooperation into your game.

☒ 💎 Virtual Goods Create Virtual Goods for any in-game assets that can be awarded, accumulated, or bought, including XP points and in-game currencies.

☒ 🏆 Achievements Set up and manage Achievements for your game and use them to reward players when they trigger certain targets such as a score or particular rank.

☒ 🎮 Multiplayer Set up Challenges to encourage your players to play with each other. Configure Matches to control how players are matched on customized player data.

☒ 💬 Messages Configure in-game messages sent to players through the use of socket-sent messages or push notifications.

☒ 📁 Downloadables Create and manage binary data content in the platform.

☒ 📋 Properties Properties are your own configuration objects you use within Cloud Code or expose to clients. Property Sets let you combine and segment Properties, delivering different configurations among players.

☒ </> Cloud Code Create custom logic to control the behavior of configurable components you build into your game.

Integrations

What providers would you like your game to integrate with?

☐ G Google

☐ 🍏 Apple / iOS

☒ 🌐 Microsoft

☐ 📦 Amazon

☐ f Facebook

☐ 🐦 Twitter

☐ PSN

☐ 🎮 Steam

☐ 📺 Twitch

☐ Kongregate

☐ Viber

☐ 🗣️ We Chat

☐ 🐼 QQ

☐ Nintendo

Dalej użytkownik ma jeszcze możliwość wyboru ustawień geograficznych swojego projektu. Ta opcja pozwala na określenie kontynentów lub nawet konkretnych państw, którym tworzący grę udostępni swój produkt. Gdy już nastąpi koniec rejestracji gry na serwerze twórca może przejść do właściwej części panelu sterowania swoją grą, który wygląda mniej więcej tak:

Game

Prototype

Master Degree

Pricing Tier: Development

Your game was created successfully.

GameSparks API Key: [X37836d9b0h0e](#) GameSparks API Secret: [FvUj9d6uzc6SbYcs7gCvY9u3C1etHsUP](#) [All Credentials](#)

Snapshots

Create Snapshots of your game to store the game's existing configuration at a particular point in time. Use Snapshots to continue game development without risk—revert to the saved Snapshot at any time should anything go wrong. If you want to publish a game Snapshot to Live, your game must be on a [pricing plan](#) that includes publishing to live.

Create

There are no snapshots for this game.

Notifications

Daily log summaries.

Preview

FATAL0

ERROR0

WARN0

INFO0

DEBUG0

Live

Game is not published to LIVE stage

View Logs

User Management

Collaborators Groups

Email

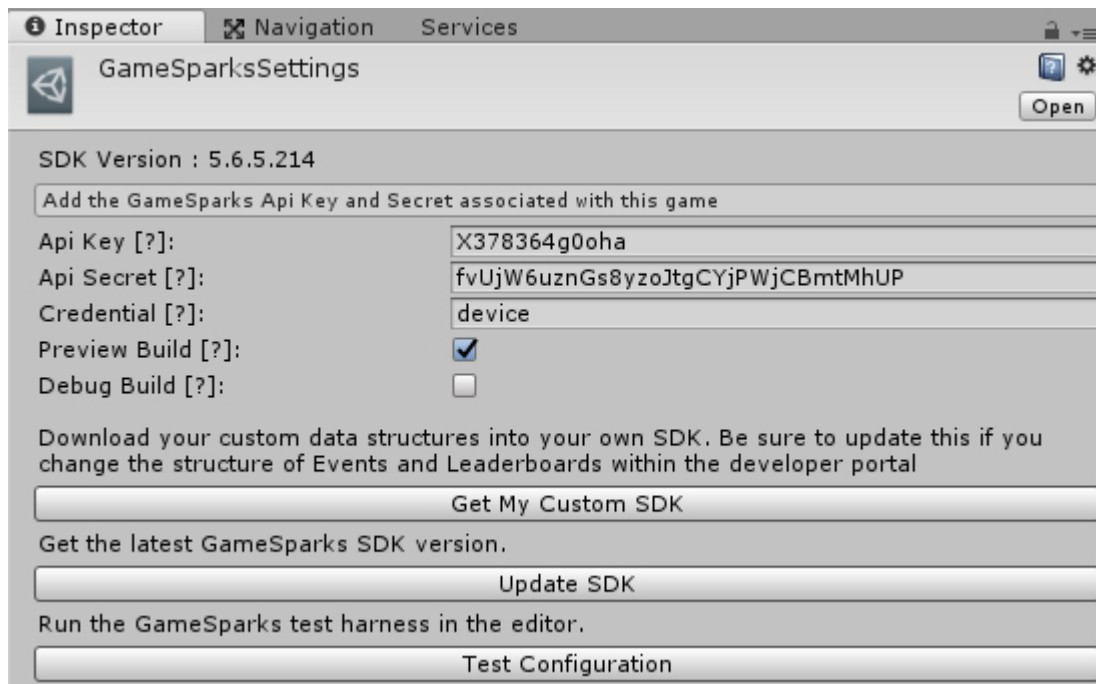
Groups

lpystak27@gmail.com

Owner

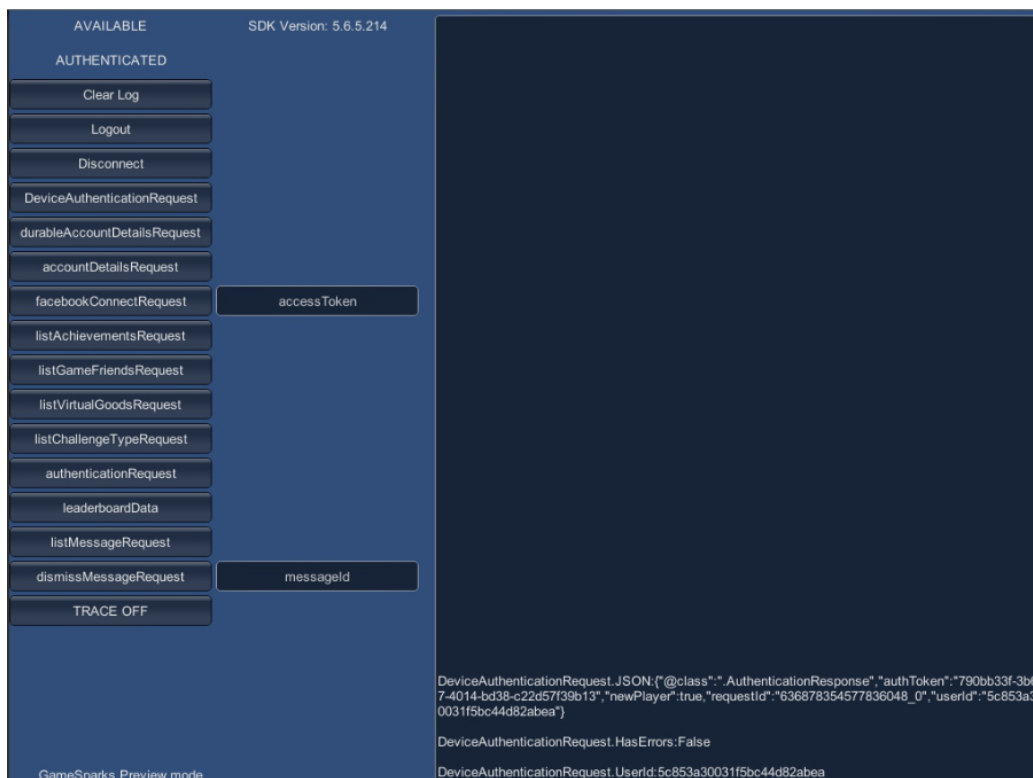
You cannot add collaborators unless you have at least one group. [Add Group](#)

Należy pobrać *GameSparks API Key* oraz *GameSparks API Secret* a następnie skonfigurować połączenie w *Inspektorze* w Unity:



Rysunek 37. GameSparksSettings

Na powyższym zdjęciu widać miejsca, w których należy umieścić klucze aplikacji otrzymane w panelu sterowania *Gamesparks*. Jest tu również wcześniej wspomniana opcja testu połączenia. Poniższe zdjęcie obrazuje ilość jej funkcjonalności.



Rysunek 38. Jak widać na załączonym zdjęciu, po kliknięciu *DeviceAuthenticationRequest* nastąpiło połączenie z serwerem. Połączenie z serwerem zostało nawiązane

6. Wyniki

6.1. Analiza otrzymanych wyników

❖ Dostęp do informacji na temat funkcji sieciowych i jakość tych informacji.

Każda z grup usług posiada swoją dokumentację i forum, oba źródła informacji w języku angielskim. Co się tyczy wszystkich trzech forów użytkowników, niekiedy zdarzają się błędne sugestie w stosunku do pytań osób zakładających konkretny temat.

Jeśli chodzi o informacje na temat funkcjonalności *UNet* to część dokumentacji skupiona wokół usług sieciowych napisana jest w sposób przejrzysty, gwarantujący zrozumienie tematu. Zawiera krótką informację na temat tego co znajduje się w omawianych rozdziałach, informuje, w którym miejscu użytkownik powinien zacząć czytać jeśli już wcześniej miał styczność z usługami sieciowymi. Pozwala wyjaśnić czym jest *HLAPI* Unity, wokół czego działa, jakie funkcjonalności gwarantuje i zawiera przykładowe projekty angażujące takie funkcjonalności jak matchmaking. Korzystający z *UNet* jest przeprowadzany przez ustawienia projektu gry wieloosobowej, jest mu wyjaśnione w sposób prosty za co odpowiada dany komponent, skrypt, metoda czy zmienna. Zawiera przykładowe linie kodu w języku C# wraz z ich opisem w komentarzach.

W swojej dokumentacji *PUN* zawiera większość pozytywnych cech charakteryzujących *UNet*, takich jak przejrzysty sposób omawianych komponentów i metod w *SDK*, informuje co znajduje się w konkretnej podstronie dokumentu. Ponadto użytkownikowi dokumentacji przedstawione są demonstracyjne wersje programów i kursy pozwalające szybciej zapoznać się z usługami. Objasnia kluczowe funkcjonalności takie jak logowanie, rejestracja, matchmaking. Udostępnia narzędzie do radzenia sobie z błędami poprzez informowanie użytkownika o najczęściej kierowanych pytaniach do działu wsparcia (wraz z odpowiedziami) oraz informacje na temat znanych problemów, jeśli aktualnie takie są.

Gamesparks również zawiera większość informacji dostępnych w *UNet* a także kursy wyjaśniające działanie poszczególnych funkcjonalności. W tym przypadku są to jednak lekcje w formacie wideo. Schemat dokumentacji usług *Gamesparks* dzieli się m.in. na sekcje dla początkujących, właściwą dokumentację dla programistów, kursy wideo, porady i sztuczki, forum i dział wsparcia.

❖ Ilość zapewnianych usług.

Ilość zapewnianych usług w przypadku *UNet*, *PUN* i *Gamesparks* jest zbliżona do siebie. Różnica polega na rodzajach poszczególnych metod. *UNet* stanowi standard usług potrzebnych do utworzenia aplikacji sieciowej natomiast pozostałe dwa zestawy usług rozwijają je i dają więcej możliwości. Przykładem tego jest m.in. wiele rodzajów logowania lub formy dobierania przeciwników w meczu. *UNet* skupia się na tradycyjnym dostępie do gry, *Photon* daje więcej możliwości matchmakingu, a *Gamesparks* ponadto daje wiele możliwości konfiguracji pokoi w grze.

❖ Trudność w implementacji poszczególnych funkcjonalności, prostota.

UNet z racji największej integracji z silnikiem Unity 3D posiada większość swoich skryptów funkcjonalności dołączonych do komponentów typu *przenieś i upuść* (ang. *Drag and drop*). Wystarczy odpowiedni komponent przeciągnąć w wymagane miejsce i sprawdzić czy usługa działa.

Photon Unity Networking nie ma tak dobrej integracji z silnikiem gry, głównie opiera się na pisaniu przez programistę skryptów z korzystaniem metod klasy *PhotonNetwork*. W większości przypadków ogranicza się to do kilku linii kodu na plik C#.

Tak samo jak w *PUN*, *Gamesparks* nie ma natywnej integracji z Unity 3D. Opiera się na wtyczce pozwalającej skonfigurować połączenie i to wszystko. W tym przypadku również wymaga się od programisty pisania różnej wielkości skryptów (w zależności od rozbudowania funkcjonalności). Jednak tutaj sprawa jest dodatkowo utrudniona poprzez fakt dużego nacisku silnika usług na korzystanie z przesyłu żądań (*requests*). Wymaga to większej znajomości tematu sieci.

❖ Panel sterowania – ilość opcji i ich przydatność.

Panel sterowania *UNet* zawiera wiele przydatnych w procesie utrzymania opcji. Są to m.in. narzędzia do analizy i zarządzania reklamami oraz monetyzacją w aplikacji, diagnostyka oparta na chmurze obliczeniowej zawierająca m.in. zgłoszone przez użytkowników programu błędy, powiadomienia z aplikacji i informacje na temat nowych użytkowników. Pozwala ustawić limit użytkowników aplikacji grających w jednej chwili. Łącznie około 20 funkcjonalności.

Panel *PUN* ogranicza się do możliwości dodania nowej gry i ustanowienia identyfikatorów służących do łączenia z serwerem po stronie Unity. *Photon* udostępnia programiście narzędzia do analizy danych gry i jej użytkowników, pozwala ustawiać liczbę jednoczesnych graczy a także pozwala posortować utworzone aplikacje wg kilku kryteriów.

Gamesparks posiada wszystkie wyżej wymienione funkcjonalności dodając do nich jeszcze m.in. możliwość blokowania usług na dany kraj lub kontynent, pozwala z łatwością przeprowadzać Alfa i Beta testy oraz z poziomu panelu sterowania pozwala zaplanować i przeprowadzać akcje promocyjne gry. Rozbudowane narzędzie do analizy pozwala na podgląd wielu przydatnych danych jak choćby rozróżnienie graczy w grze – pozwala określić ilość stałych wracających do aplikacji klientów. *Gamesparks* ponadto umożliwia łatwy eksport analizowanych danych do pliku.

❖ Podatność na błędy i opóźnienia.

W trakcie pisania aplikacji w oparciu o *UNet* kilkakrotnie po zbudowaniu gry nastąpiło wyrzucenie użytkownika z serwera.

Z racji podobnej architektury oprogramowania oraz informacji zaciągniętych z forów użytkowników *UNet* i *PUN* można wywnioskować, że podobne rozłączenia mogą nastąpić również w silniku *Photon*.

Zaobserwowano również drobne opóźnienia na linii klient-serwer po kliknięciu klawiszem ruchu w aplikacji.

W przypadku usług *Gamesparks* należy liczyć się z trudnościami na płaszczyźnie przesyłu żądań między klientami a serwerem.

❖ Pozostałe cechy.

Należy przede wszystkim zwrócić uwagę na prostotę rozwiązań *UNet* i *PUN* oraz na ilość funkcjonalności zestawu usług sieciowych *Gamesparks*. Więcej informacji na temat tych usług znajduje się w Rozdziale 4 i Rozdziale 5 omawianej pracy magisterskiej oraz na stronach dokumentacji każdej z trzech usług:

- *Unity Networking*: docs.unity3d.com/
- *Photon Unity Networking*: doc.photonengine.com/
- *Gamesparks*: docs.gamesparks.com/ oraz gamesparks.com/product/

7. Wnioski

7.1. Wnioski z otrzymanych danych

Na podstawie otrzymanych danych można jasno zobaczyć w jakich przypadkach każdy z trzech omawianych zestawów usług sieciowych znajdzie swoje zastosowanie. Nie da się jednoznacznie odpowiedzieć na pytanie, który zestaw usług jest najlepszy, gdyż jest to zależne od wielu składowych jak np. ilość graczy lub złożoność projektu.

UNet to zbiór usług podstawowych, koniecznych w procesie tworzenia gry sieciowej. Jest zintegrowany z silnikiem gier przez co ze wszystkich trzech jest najprostszy w obsłudze. Wielką zaletą jest możliwość przeniesienia komponentu i bezpośredniego dodania go do obiektu. Posiada dobre źródło informacji na temat swoich komponentów w postaci dokumentacji silnika Unity 3D. W razie potrzeby znalezienia informacji na temat funkcjonalności sieciowych i funkcjonalności gry użytkownik obie szukane informacje może znaleźć w jednym miejscu. Posiada rozbudowany panel sterowania aplikacjami, podstawowe funkcje sieciowe odpowiedzialne za szukanie rozgrywki i nie wymaga od użytkownika rejestracji gry w Unity 3D jak pozostałe dwa zestawy usług.

Wadami *UNet* są braki w materiałach szkoleniowych, rozłączenia z serwerem, niewielkie opóźnienia, ograniczone opcje doboru przeciwników do gry oraz fakt, że producenci silnika gry do 2022 roku zamierzają wycofać wsparcie dla niego.

Niemniej jednak jest to najlepsze narzędzie, by zacząć swoją naukę z usługami sieciowymi i pisanem gier online dla wielu osób.

PUN rozszerza wiele funkcjonalności *Unity Networking*. Posiada wiele więcej opcji wyszukania rozgrywki, pozwala na integrację z serwisami społecznościowymi oraz zawiera dobrze napisaną dokumentację wraz z licznymi materiałami szkoleniowymi. Pozwalają one przeprowadzić użytkownika przez cały proces pisania aplikacji sieciowej. Ma bardzo dobrze rozwinięte narzędzie do pomocy programiście w przypadku błędów oraz pozwala na łatwą konfigurację z serwerem za pomocą identyfikatora z panelu sterowania. Posiada prosty sposób implementacji czatu pisanego i czatu głosowego. Skrypty pisane w oparciu o klasę *PhotonNetwork* nie wymagają poświęcenia dużej ilości czasu na ich naukę, metody są proste a skrypty krótkie.

Wadami *PUN* są najmniejsza integracja z silnikiem Unity i mała liczba funkcjonalności panelu sterowania.

Zaleca się rozpoczęcie pisania gier w *PUN* po uprzednim zapoznaniu się z usługami sieciowymi *UNet* i skryptami Unity 3D ze względu na brak opcji *Drag and Drop*.

Gamesparks jest najbardziej rozbudowanym zestawem usług sieciowych. Udostępnia narzędzie testujące połączenie posiadające wiele rodzajów integracji m.in. z portalami społecznościowymi, ma wiele opcji konfiguracji pokoi gry, opiera się na wysyłanych żądaniach. Trudność pisania skryptów jest podobna jak w *PUN*, a ilość funkcjonalności po stronie serwera jest największa ze wszystkich trzech zestawów usług. Dodatkowym atutem usług *Gamesparks* jest doskonale opracowana dokumentacja zawierająca odnośniki do działu wsparcia, kategorię dla początkujących i filmy instruktażowe z implementowanymi funkcjonalnościami. Pozwala na przeprowadzanie kampanii marketingowej gry, tworzenie

wydarzeń dla graczy, dokładną analizę danych, wiele systemów zapewniających graczom rozrywkę i zaostarzających rywalizację poprzez systemy osiągnięć i wyzwań, w łatwy sposób pozwala na planowanie dodawania większej ilości zawartości grywalnej gry.

Wadą *Gamesparks* może okazać się jednak zbyt duża liczba funkcjonalności jeśli np. użytkownik wcale ich nie potrzebuje. Kolejnym minusem jest potrzeba większej znajomości tematu sieci komputerowych, Internetu i przesyłu żądań.

Gamesparks przeznaczony jest do użycia w przypadku tworzenia projektów na szeroką skalę. Oznacza to setki użytkowników i bardzo dużo funkcjonalności grywalnej tytułu. W innym razie twórca nie będzie w stanie w pełni spożytkować wielu funkcjonalności zbioru omawianych usług jak choćby narzędzia do analizy danych czy segmentacji (za mała ilość graczy). W przypadku mniejszych gier, prototypów lub testów należy skorzystać z *UNet* lub *PUN*.

7.2. Możliwość dalszego rozwoju

Możliwość dalszej rozbudowy pracy należy podzielić na dwie płaszczyzny, rozbudowę funkcjonalności usług sieciowych w projekcie oraz rozbudowę funkcjonalności grywalnych zwanych dalej zawartością gry.

Dalsza możliwość rozwoju pracy w oparciu o usługi sieciowe:

- zakolejkowanie gier, zmiana formatu dołączenia do gry na bazie kolejek normalnej i rankingowej;
- implementacja systemu rejestracji, logowania i/lub autoryzacji dwuetapowej;
- możliwość dołączenia do losowej kolejki gry;
- rozbudowa matchmakera w oparciu o skrypty *PUN* lub *Gamesparks*;
- implementacja czatu i/lub czatu głosowego;
- implementacja listy znajomych;
- możliwość dodania znajomego do pokoju gry;

Dalsza możliwość rozwoju pracy w oparciu o zawartość gry:

- rozbudowa funkcjonalności kamery, dodanie możliwości przesunięcia jej poprzez najazd na krawędź okna aplikacji;
- zablokowanie kamery by nie mogła wyjechać za mapę;
- implementacja przybliżenia/oddalenia kamery;
- obramowanie mapy;
- dodanie innych map, trybów gry;
- animacja ruchu postaci, ataku, przyjęcia obrażeń, śmierci, odrodzenia;
- implementacja systemu ataku, czasu odrodzenia, punktów zdrowia postaci;
- rozbudowa mapy, dodanie większej ilości szczegółów;
- dodanie neutralnych stworów gwarantujących wzmocnienia postaci;
- dodanie stworów w ramach drużyn gwarantujących obronę przed atakami wież;
- dodanie systemu wież, ich ataku w stwory i gracza;
- implementacja minimapy;
- zbudowanie klienta gry na inne platformy niż Microsoft Windows;
- implementacja systemu statystyk;

Podsumowanie

Niniejsza praca magisterska stanowi materiał poglądowy dla niezdecydowanych względem wyboru technologii osób zainteresowanych napisaniem własnej gry typu *multiplayer online* w oparciu o jeden z zestawów usług sieciowych:

- *Unity Networking*,
- *Photon Unity Networking*,
- *Gamesparks*.

Na łamach pracy dyplomowej autor przedstawił szeroki kontekst historyczny gier, ich początki i rozwój wraz z rozwojem technologii. Następnie krótko omówił udział w rynku poszczególnych rodzajów gier i przedstawił obecnie panujące trendy.

W dalszym kroku, w oparciu o publikacje publicystyczno-naukowe, autor pracy zaznajomił czytelników z aspektami gier komputerowych: psychologicznym, socjologicznym, kulturowym i sportowym. Miało to na celu nakreślenia szerszej perspektywy tematu gier.

W pracy dyplomowej zostały również podane informacje na temat obecnie najchętniej wybieranych tytułów gier z podziałem na gatunki. Następnie został poruszony i omówiony temat gier typu *free-to-play*. Czytelnicy poznali jego zalety jak i niebezpieczeństwa związane z omawianym modelem płatności. Autor pracy pozwolił też czytelnikom zapoznać się z gatunkami gier, które najczęściej z niego korzystają oraz zaznajomił czytelnika z opiniami ludzi z branży IT, co ci ostatni myślą na temat opisywanego modelu.

Dalej w pracy poruszane są tematy gier typu *Multiplayer Online Battle Arena*, w skrócie *MOBA*. Podane są przykładowe tytuły związane z omawianym gatunkiem oraz zostaje zaprezentowany tytuł wiodący prym w tym gatunku jak i w e-sporcie na przestrzeni ostatnich 10 lat – gra *League of Legends*. Następnie na jej podstawie praca wyjaśnia czym jest pojęcie *matchmakingu* i jakie znaczenie ma ono obecnie w grach sieciowych, przytaczając solidne argumenty w postaci sprawiedliwości gry, preferencji wyboru i krótkiego czasu oczekiwania na mecz. Uświadamia czytelnika, że ze względu na wiele składowych, znajdowanie przeciwników nie jest elementem łatwym do osiągnięcia i że wielokrotnie w celu optymalizacji *matchmakingu* programiści muszą iść na kompromis z graczami.

Przedstawiony został temat e-sportu (e-sportów), jego początki, pierwsze turnieje i pierwsze transmisje telewizyjne. Następnie autor przedstawił obecny rozwój tej dyscypliny sportu, osiągnięcia związane z liczbami widzów na portalach streamingowych oraz coraz bardziej rosnące zainteresowanie na aspekt sportowy gier komputerowych. Poruszony został temat zarobków graczy, sum przeznaczanych przez wielkie firmy na turnieje rangi mistrzowskiej oraz rozmach z jakim są one organizowane (Stadion Olimpijski w Pekinie, 91 tysięcy miejsc siedzących). Dalej poruszony został drażliwy temat wielu dyskusji, a mianowicie czy e-sport to już sport oraz czy powinien znaleźć się na Olimpiadzie Letniej w Tokio lub w Paryżu. Przytoczone zostały argumenty obu stron biorących udział w dyskusji oraz działania jakie podjął *MKOL* w celu wypracowania obopólnych korzyści i rozwoju sportu tradycyjnego i elektronicznego.

Omówiony został trening graczy e-sportowych wraz z przytoczeniem opinii Pani Urszuli Klimczak, światowej klasy e-sportowej psycholog i *performance coach*, na temat

świadomości graczy odnośnie diety i snu w przygotowaniu do meczu. Podane zostały konkretne tytuły gier e-sportowych z podziałem na gatunki. Tak samo jak gry, przedstawione czytelnikom w formie tabeli zostały największe turnieje gier e-sportowych na świecie wraz z ich krótkim opisem.

Wszystko po to, by nakreślić szeroki zakres wpływu gier na społeczeństwo w ostatnich latach. By uświadomić czytelnikom jak dużą wagę należy przyłożyć w przyszłości do rozwoju usług sieciowych oraz jak ważne są te usługi w wieloosobowych grach online.

W dalszym kroku omówiona została architektura oprogramowania w oparciu o model klient-serwer, rodzaje silników do tworzenia gier komputerowych, specyfikacja wymagań tworzonej gry oraz metodyka pracy. Na tę ostatnią złożyły się opisy silnika gier Unity 3D, zintegrowanego środowiska dla programistów Visual Studio, obiektowego języka programowania C# oraz trzech, wyżej wymienionych zestawów usług sieciowych wraz z przeglądem ich funkcjonalności.

W Rozdziale 5 omawianej pracy dyplomowej autor wykonał niewielki prototyp gry komputerowej typu *MOBA*. Przedstawił schemat postępu prac nad projektem z wypunktowanymi etapami tworzenia mapy, postaci i implementacji wybranych usług sieciowych. Następnie na łamach tej pracy odbyło się porównanie *UNet*, *PUN* i *Gamesparks* co posłużyło autorowi pracy do przedstawienia wyników przeprowadzonych badań w Rozdziale 6 pracy magisterskiej.

Badane były: dostęp do informacji, ilość zapewnianych usług, trudność w implementacji, jakość panelu sterowania oraz podatność na błędy. Na tej podstawie w Rozdziale 7 zostały wyciągnięte wnioski informujące czytelników na temat przydatności badanych usług w projekcie. Podane również zostały zalecenia co do sposobu użytkowania każdego z zestawów usług sieciowych.

Na podstawie zgromadzonej wiedzy autor przedstawił również sposób dalszego rozwoju prototypu w oparciu o usługi sieciowe i zawartość grywalną aplikacji.

Adnotacje i słownik

Aplikacja – w tej pracy w znaczeniu wytwarzanego oprogramowania, gry.

API – interfejs programowania aplikacji.

Assets – zasoby w postaci zawartości do gry, np. bohaterowie, broń, ikony itp. własne bądź pozyskane za darmo ze sklepu zasobów *Unity*, tzw. *AssetStore*.

Asset Store – sklep zasobów dostępny w *Unity 3D* oferujący zawartość do gier i programów za darmo lub za pieniądze. W skład jego zawartości wchodzi wyżej wymienione modele postaci (z lub bez animacji ruchu), animacje, obrazki, ustawienia kamer i światła, skrypty, pliki audio, narzędzia, pliki skryptów szkoleniowych wraz z materiałami wideo, mapy itp.

Atari – amerykańska firma produkująca sprzęt komputerowy, automaty wideo i domowe konsole gier.

Bot – sterowany za pomocą sztucznej inteligencji bohater gry typu *MOBA*

Dżungla – obszar na mapie w grze typu *MOBA* nie będący linią/aleją.

Dżungler – uczestnik rozgrywki grający w dżungli (tak jak inni gracze grają na liniach).

EA – Electronic Arts, firma produkująca gry komputerowe, m.in. *Fifa*, *BF*, *APEX Legends*.

E-sport (ang. **Esport**) – również znane pod nazwą sportów elektronicznych, forma rywalizacji przy użyciu gier wideo. Rywalizacja odbywa się zarówno w formie rekreacyjnej jak i zarobkowej, w większości przypadków e-sport przyjmuje formę zorganizowanych zawodów czy też turniejów gier komputerowych lub konsolowych między profesjonalnymi graczami i zespołami.

F2P (**Free-to-play**) – typ gier i zarazem model płatności. Oznacza grę, która jest darmowa, tj. taka której użytkownik nie musi kupić lub płacić za nią abonamentu by w nią grać. System ten polega na kupowaniu przez użytkownika opcjonalnych dodatków do gier np. w postaci broni, bohaterów, skórek zmieniających wygląd naszego bohatera, awatarów czy wspomagaczy zdobywania punktów doświadczenia czy waluty gry.

FPS – ang. *First Person Shooter*, strzelanka w pierwszej osobie, uczestnik gry widzi poczynania swojej postaci jej oczami.

Framework – innymi słowy platforma programistyczna.

HLAPI – API wysokiego poziomu.

Host – udostępniający usługi.

Klient – program komputerowy odbierający informacje z serwera.

LLAPI – API niskiego poziomu.

League of Legends – wydana w 2009r. gra online z gatunku *MOBA*, Twórcą *LoL*'a jest amerykańska firma *Riot Games*. Dnia 15 kwietnia 2013 roku została uznana w USA za pełnoprawny sport. Gra jest dostępna w 16 wersjach językowych, posiada swój własny system gier rankingowy i jest dostępna na *Windows* oraz *OS X*.

Matchmaker – system dobierania przeciwników do gry.

Matchmaking – proces dobierania graczy do gry oparty w danej grze na pewnych założeniach matchmakera.

Mecz – pojedyncza instancja gry.

Metacritic – portal opiniotwórczy zreszający krytyków filmów, gier i książek.

Microsoft – jedna z największych na świecie firm branży IT, odpowiada m.in. za systemy Windows oraz produkcję tabletów Surface PRO.

M.I.T. – Massachusetts Institute of Technology

MOBA (*Multiplayer Online Game Arena*) – podgatunek strategicznych gier czasu rzeczywistego, najczęściej ograniczona liczbą graczy dzielących się na dwie bądź więcej drużyn walczących na arenie/mapie we wspólnym celu. Termin wymyślony przez firmę *Riot Games* przy okazji premiery ich produktu – *League of Legends*. Protoplastą gatunku była modyfikacja gry *StarCraft: Aeon of Strife*.

Pathfinding – zagadnienie polegające na znalezieniu jak najkrótszej drogi z punktu A do punktu B, w przypadku gry komputerowej jest to sprawa kluczowa ze względu na możliwość postaci poruszania się po mapie.

Pay-to-win – standardowy model płatności za gry komputerowe polegający na wcześniejszym opłaceniu wszystkich kosztów gry w celu skorzystania z niej po raz pierwszy.

PlayStation – Seria konsoli gier wideo, produkowana przez firmę Sony.

Postać – bohater wybrany przez użytkownika, którym będzie operował w trakcie rozgrywki.

PvP – w grach *MOBA* tryb *Player vs Player* oznacza rodzaj rozgrywki nieuwzględniający Botów.

RTS – gra strategiczna czasu rzeczywistego.

Serwer – program komputerowy udostępniający usługi Klientom.

Silniki gier – Zintegrowane środowiska programistyczne przeznaczone do tworzenia dwu- i trójwymiarowych gier, wizualizacji i aplikacji na komputery, smartfony i konsole. Stworzone specjalnie na potrzeby osób bądź zespołów osób tworzących programy interaktywne w celu ułatwienia i przyspieszenia ich pracy. Zazwyczaj zajmują się interakcją pomiędzy poszczególnymi elementami gry takimi jak assety. Mogą służyć również obróbce zdjęć i plików audio, tworzeniu animacji i modeli 3D. Bywają mylone z silnikami graficznymi. Przykładami silników gier są m.in. Unity 3D i Unreal Engine.

Silnik sieciowy – zbiór usług sieciowych, z których użytkownik może korzystać po połączeniu silnika ze swoim programem.

Umiejętność – Atak lub czar użyty przez postać po kliknięciu odpowiedniego przycisku/klawisza przez gracza.

Unity 3D – silnik gier wideo oparty na języku C# i interfejsie *drag and drop*.

Xbox – seria konsol gier wideo, produkowana przez firmę Microsoft.

Bibliografia

1. Wijman, Tom. *Mobile Revenues Account for more than 50% of the global games market as it reaches \$137.9 billion in 2018*. newzoo.com. 2018. Data dostępu: 01.03.2019.

Dostęp w Internecie:

www.newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/

2. Surdyk, Augustyn. *Edukacyjna funkcja gier w dobie "cywilizacji zabawy"*. Zakład Teorii i Filozofii Komunikacji IF UAM. 2008, s.28. Data dostępu: 01.03.2019.

Dostęp w Internecie:

www.repozytorium.amu.edu.pl/bitstream/10593/1585/1/Edukacyjna%20funkcja%20gier%20w%20dobie%20cywilizacji%20zabawy.pdf

3. Orenstein, Hannah. *Video Games Effects: Teen gamers better at Virtual Surgery Than Medical Residents*. The Huffington Post. 2012. Data dostępu 01.03.2019.

Dostęp w Internecie:

www.huffingtonpost.com/2012/11/16/video-game-effects-teen-g_n_2147062.html?guccounter=1&guce_referrer_us=aHR0cHM6Ly9wbC53aWtpcGVkaWEub3JnLw&guce_referrer_cs=_xC5jAbwdoz2Yt8aGHlbWQ

4. Musiolik, Paweł. *Krytyka Star Wars Battlefront 2 kosztowała EA aż 3 miliardy dolarów*. Ppe.pl. 2017. Data dostępu 01.03.2019.

Dostęp w Internecie:

<https://www.ppe.pl/news/119282/krytyka-star-wars-battlefront-2-kosztowala-ea-az-3-miliardy-dolarow.html>

5. Makuch, Eddie. *Freemium is the future, says EA*. Gamespot.com. 2012. Data dostępu: 01.03.2019.

Dostęp w Internecie:

<https://www.gamespot.com/articles/freemium-is-the-future-says-ea/1100-6389123/>

6. Dedmon, Tanner. *League of Legends top Free-to-play revenue charts in 2017*. Comicbook.com. 2018. Data dostępu: 01.03.2019.

Dostęp w Internecie:

www.comicbook.com/gaming/2018/01/30/league-of-legends-top-free-to-play-revenue-charts-in-2017/

7. Słownik gracza. *MOBA, Multiplayer Online Battle Arena*. gryonline.pl. 2013. Data dostępu: 01.03.2019.

Dostęp w Internecie:

www.gry-online.pl/slownik-gracza-pojecie.asp?ID=341

8. League of Legends Wiki. *Elo rating system*. leagueoflegends.fandom.com. 2018. Data dostępu: 01.03.2019.

Dostęp w Internecie:

www.leagueoflegends.wikia.com/wiki/Elo_rating_system/

9. Isto. *Przewodnik systemu dobierania graczy*. support.riotgames.com. 2019. Data dostępu: 14.03.19.

Dostęp w Internecie:

www.support.riotgames.com/hc/pl/articles/201752954-Przewodnik-systemu-dobierania-przeciwnik%C3%B3w

10. *All Time PC Metascore – game releases by metascore*. Metascore.com. Data dostępu: 10.03.2019.

Dostęp w Internecie:

www.metacritic.com/browse/games/score/metascore/all/pc/filtered

11. Onyett, Charles. *Death of the disc-based game*. Ign.com. 2011. Data dostępu: 01.03.2019.

Dostęp w Internecie:

www.ign.com/articles/2011/08/02/death-of-the-disc-based-game

12. Kelly, Neon. *BioWare on subscriptions vs Free-to-play*. Videogamer.com. 2010. Data dostępu: 05.03.2019.

Dostęp w Internecie:

https://www.videogamer.com/news/bioware_on_subscriptions_vs_free-to-play

13. Dokumentacja Unity. Data dostępu: 10.03.2019.

Dostęp w Internecie:

www.docs.unity3d.com/Manual/UNet.html/

14. Dokumentacja Photon. Data dostępu: 10.03.2019.

Dostęp w Internecie:

www.doc.photonengine.com/en-us/pun/current/getting-started/pun-intro

15. Strona główna gamesparks.com. Data dostępu: 10.03.2019.

Dostęp w Internecie:

www.gamesparks.com/

16. Burns, Tom. *‘E-sport’ can now drop ‘e’*. aljazeera.com. 2014. Data dostępu: 03.03.2019.

Dostęp w Internecie:

www.aljazeera.com/indepth/opinion/2014/07/esports-can-now-drop-e-2014724112549724248.html

17. Grohmann, Karolos. *E-sports get closer to being part of the Olympics*. Reuters. 2017.
Data dostępu: 03.03.2019.

Dostęp w Internecie:

www.businessinsider.com/e-sports-gets-closer-to-being-part-of-the-olympics-games-2017-10?IR=T

18. Kubicki, Krzysztof. *Zawodnik gra ciałem ale wygrywa głową*. Emine.pro/Sport.pl. 2019.
Data dostępu: 03.03.2019.

Dostęp w Internecie:

<http://www.sport.pl/sport/7,164252,24428747,zawodnik-gra-cialem-ale-wygrywa-glowa-psycholog-pomaga-esportowcom.html>

19. *History of video games*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/History_of_video_games

20. *Spacewar!*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://en.wikipedia.org/wiki/Spacewar!>

21. *Gra komputerowa*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://pl.wikipedia.org/wiki/Gra_komputerowa

22. *List of video game genres*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/List_of_video_game_genres

23. *List of best-selling video games*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/List_of_best-selling_video_games

24. *Free-to-play*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://en.wikipedia.org/wiki/Free-to-play>

25. Starcraft Wiki. *Aeon of Strife(map)*. Starcraft.fandom.com. 2017. Data dostępu: 10.03.2019.

Dostęp w Internecie:

[https://starcraft.fandom.com/wiki/Aeon_of_Strife_\(map\)](https://starcraft.fandom.com/wiki/Aeon_of_Strife_(map))

26. *DotA*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/Defense_of_the_Ancients

27. *MOBA*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/Multiplayer_online_battle_arena

28. *List of multiplayer online battle arena games*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/List_of_multiplayer_online_battle_arena_games

29. *Matchmaking*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

[https://en.wikipedia.org/wiki/Matchmaking_\(video_games\)](https://en.wikipedia.org/wiki/Matchmaking_(video_games))

30. *League of Legends*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/League_of_Legends

31. Strona główna League of Legends. Data dostępu: 14.03.2019.

Dostęp w Internecie:

pvp.net

32. Lista zwycięzców i nominowanych na gali The Game Awards. Thegameawards.com.2018. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://thegameawards.com/awards/>

33. Piłat, Jarosław. *Transmisja z LEC 2019 na kanałach Polsat Games*. 2019. Data dostępu: 04.03.2019.

Dostęp w Internecie:

<http://cybersport.pl/179640/transmisja-z-lec-2019-na-kanalach-polsat-games/>

34. *Esports*. Wikipedia.org. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://en.wikipedia.org/wiki/Esports>

35. Profil gracza *Shiphtur* na stronie *lol.gamepedia.com*. 2019. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://lol.gamepedia.com/Shiphtur>

36. *Klient-serwer*. *Wikipedia.org*. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://pl.wikipedia.org/wiki/Klient-serwer>

37. *Unity*. *Wikipedia.org*. Data dostępu: 14.03.2019.

Dostęp w Internecie:

[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

38. *Visual Studio*. *Wikipedia.org*. Data dostępu: 14.03.2019.

Dostęp w Internecie:

https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

39. *C#*. *Wikipedia.org*. Data dostępu: 14.03.2019.

Dostęp w Internecie:

[https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))

40. Opis usług *gamesparks*. *Gamesparks.com*. Data dostępu: 12.03.2019.

Dostęp w Internecie:

<https://www.gamesparks.com/product/>

41. Frisk, Adam. *Video Gaming as an Olympic Sport? IOC hosting Esports forum to better understand competitive gaming*. *Globalnews.ca*. 2018. Data dostępu: 14.03.2019.

Dostęp w Internecie:

<https://globalnews.ca/news/4340744/esports-in-olympics/>

42. Morris, Chris. *Video games won't be part of the Paris Olympics*. *Fortune.com*. 2018. Data dostępu: 10.03.2019.

Dostęp w Internecie:

<http://fortune.com/2018/12/10/olympics-video-games-paris-2024/>

43. Dokumentacja *gamesparks.com*. Data dostępu: 14.03.2019.

Dostęp w Internecie:

docs.gamesparks.com

Spis rysunków

Rysunek 1. Spacewar! na komputerze PDP-1 firmy DEC.....	6
Rysunek 2. Automat do gry w Pong.....	6
Rysunek 3. Atari 2600 – jedna z pierwszych popularnych konsol domowych.....	7
Rysunek 4. Sony PlayStation	7
Rysunek 5. Pierwszy Xbox firmy Microsoft.....	7
Rysunek 6. Kadr z filmu umieszczonego w ramach easter egga GTA w grze Max Payne 2.....	10
Rysunek 7. Kadr z gry Starcraft: Aeon of Strife.....	14
Rysunek 8. Ekran ładowania gry DotA.....	14
Rysunek 9. Logo firmy Riot Games.....	15
Rysunek 10. Obrazek poglądowy ukazujący jak wygląda typowa mapa gry MOBA	16
Rysunek 11. Kadr z trybu dla oglądających z gry League of Legends ukazujący walkę drużynową trzech na trzech na górnej alei	18
Rysunek 12. op.gg.....	21
Rysunek 13. Logo gry League of Legends	23
Rysunek 14. League of Legends World Championship 2016, Staples Center, Los Angeles, USA.....	25
Rysunek 15. Statystyki gracza o pseudonimie KOLD w LEC, Polsat Games	27
Rysunek 16. Jarosław Pasha Jarząbkowski w czasach gry w zespole Virtus Pro	29
Rysunek 17. Zobrazowanie architektury typu klient-serwer	34
Rysunek 18. Logo Unity 3D	35
Rysunek 19. Logo Visual Studio	39
Rysunek 20. Przykładowy listning kodu w języku C#	40
Rysunek 21. Logo PUN.....	43
Rysunek 22. Dostępne wersje PUN.....	44
Rysunek 23. Logo Gamesparks	45
Rysunek 24. Logo Gamesparks	45
Rysunek 25. Gamesparks – Logo rywalizacji	47
Rysunek 26. Gamesparks – logo walut.....	48
Rysunek 27. Dostępne formy integracji z Gamesparks	51
Rysunek 28. Mapa po zaimportowaniu potrzebnych elementów.....	57
Rysunek 29. Wyeksportowaną do osobnego folderu postać należy przeciągnąć na miejsce opcji Player Prefab w Spawn Info	61
Rysunek 30. W lewym górnym rogu widoczny komponent NetworkManagerHUD. Pozwoli on graczom łączyć się z rozgrywką.....	62
Rysunek 31. Dwie dostępne opcje rozpoczęcia rozgrywki z zaznaczoną opcją na przemian	62
Rysunek 32. Zdjęcie przedstawiające panel zarządzania aplikacją w Unity Networking	64
Rysunek 33. Funkcje, które zapewnia programiście Unity Networking.	64
Rysunek 34. Wiele korzyści może wynikać z korzystania z silnika gier Unity 3D.	65
Rysunek 35. Klient A tworzący pokój gry o nazwie AbaddOn oraz Klient B widzący stworzony pokój ..	66
Rysunek 36. Utworzona gra. Poniżej w Inspektorze poprawna konfiguracja gry z serwerem	67
Rysunek 37. Po kliknięciu DeviceAuthenticationRequest łączenie z serwerem.....	69

Spis tabel

Tabela 1. Ranking najlepiej sprzedających się gier w historii.....	11
Tabela 2. Najlepsze gry komputerowe wg krytyków i dziennikaży metacritic.com	11
Tabela 3. Tabela przedstawiająca przykładowe gry f2p wraz z ich ocenami na metacritic.com	13
Tabela 4. Role postaci z opisem.....	17
Tabela 5. Oceny LoL.....	23
Tabela 6. Przykładowe gatunki gier e-sportowych.....	30
Tabela 7. Turnieje rangi światowej	32
Tabela 8. Porównanie Unity 3D z Unreal Engine 4	35
Tabela 9. Tabela przedstawiająca podział na frakcje funkcjonalności Gamesparks.....	46