



**Politechnika
Śląska**

PROJEKT INŻYNIERSKI

„Implementacja w środowisku Unity gry 2D typu Block Breaker”

Kamil TLAŁKA
Nr albumu 294188

Kierunek: Informatyka
Specjalność: Grafika komputerowa

PROWADZĄCY PRACĘ
dr Ewa Lach
KATEDRA Grafiki, Wizji Komputerowej i Systemów Cyfrowych

Wydział Automatyki, Elektroniki i Informatyki

KATOWICE 2023

Tytuł pracy:

Implementacja w środowisku Unity gry 2D typu Block Breaker

Streszczenie:

Tematem pracy inżynierskiej jest stworzenie gry na urządzenia mobilne przy użyciu silnika Unity. Gatunek gry to Block Breaker, czyli gry Arcade polegającej na niszczeniu bloków przy pomocy piłki odbijanej przez paletkę u dołu ekranu. Dostępnych jest wiele poziomów zapewniających długą rozgrywkę. Dostępny jest również losowy generator poziomów oraz sklep z elementami kosmetycznymi.

Praca składa się z siedmiu rozdziałów. Pierwsze dwa wprowadzają do tematu, analizują problem oraz podobne aplikacje. Kolejne trzy rozdziały są poświęcone budowie aplikacji, jej wymaganiom oraz opisem działania. Załączony jest również opis wykorzystanych narzędzi. Ostatnie dwa rozdziały podsumowują wykonaną pracę, opisują wykonane testy oraz pokazują możliwość dalszego rozwoju.

Słowa kluczowe:

Block Breaker, gra mobilna, Unity

Thesis title:

Implementation in Unity environment 2D game Block Breaker Type

Abstract:

The subject of the engineering thesis is the creation of a mobile device game using the Unity engine. This application belongs to the type of Block Breaker, which is an Arcade game consisting of destroying blocks with a ball bouncing off the paddle at the bottom of the screen. There are multiple levels providing long gameplay. There is also a random level generator and a cosmetic items shop.

The work consists of seven chapters. The first two focus on the introduction to the work and the analysis of the problem, along with similar applications. The next three chapters are devoted to the construction of the application, its requirements and a description of its operation. A description of the tools used is also provided. The last two chapters summarize the work done, describe the tests carried out and show the possibility of further development.

Keywords:

Block Breaker, mobile game, Unity

Spis treści

ROZDZIAŁ 1 WSTĘP	8
ROZDZIAŁ 2 ANALIZA TEMATU	9
2.1. HISTORIA GIER KOMPUTEROWYCH	9
2.2. POCZĄTEK GIER MOBILNYCH	11
2.3. GRA ARCADE	12
2.4. GRA TYPU BLOCK BREAKER.....	13
2.5. AKCELEROMETR W GRACH.....	13
2.6. PODOBNE TYTUŁY NA RYNKU.....	14
2.6.1. <i>Brick Breaker Star</i>	14
2.6.2. <i>BBTAN</i>	16
ROZDZIAŁ 3 WYMAGANIA I NARZĘDZIA	17
3.1. PRZEGLĄD NARZĘDZI	17
3.1.1. <i>Unity</i>	17
3.1.2. <i>Visual Studio</i>	18
3.1.3. <i>GIMP</i>	18
3.1.4. <i>Stable Diffusion</i>	18
3.1.5. <i>Audacity</i>	19
3.2. WYMAGANIA W STOSUNKU DO GRY	19
3.2.1. <i>Wymagania funkcjonalne</i>	19
3.2.2. <i>Wymagania нефункционалне</i>	20
ROZDZIAŁ 4 SPECYFIKACJA ZEWNĘTRZNA	21
4.1. MENU GŁÓWNE	21
4.2. SKLEP.....	22
4.3. ZASADY	23
4.4. WYBÓR POZIOMÓW.....	23
4.5. OKNO GRY	24
4.6. WYMAGANIA DOTYCZĄCE TELEFONU ORAZ INSTALACJA	26
ROZDZIAŁ 5 SPECYFIKACJA WEWNĘTRZNA	27
5.1. PRZECHOWYWANIE ZMIENNYCH	27
5.2. KLASY INTERFEJSU.....	27
5.2.1. <i>Klasa przejść</i>	27
5.2.2. <i>Klasa odblokowywania poziomów</i>	28
5.2.3. <i>Klasa sklepu</i>	28
5.3. KLASY ODPOWIEDZIALNE ZA ROZGRYWKĘ	29
5.3.1. <i>Klasa cegły</i>	29
5.3.2. <i>Klasa piłki</i>	29
5.3.3. <i>Klasa paletki</i>	30
5.3.4. <i>Klasa usuwania postępu</i>	30
5.3.5. <i>Klasa generowania poziomów</i>	31
5.4. TWORZENIE GRAFIKI I EFEKTÓW DŹWIĘKOWYCH.....	32
5.4.1. <i>Generowanie elementów graficznych</i>	32
5.4.2. <i>Tworzenie efektów dźwiękowych</i>	33

ROZDZIAŁ 6 TESTY I DALSZY ROZWÓJ	35
6.1. TESTOWANIE.....	35
6.2. DALSZY ROZWÓJ	35
ROZDZIAŁ 7 PODSUMOWANIE	37
BIBLIOGRAFIA	39
SPIS RYSUNKÓW	40
DODATKI	41
LISTA DODATKOWYCH PLIKÓW, UZUPEŁNIAJĄCYCH TEKST PRACY	42

Rozdział 1

Wstęp

Wraz z rozwojem komputerów oraz technologii cyfrowej nastąpił również rozwój dostępnej rozrywki, jaką są gry komputerowe. Coraz więcej młodych ludzi wybiera taką formę spędzania wolnego czasu. Można debatować na temat tego, czy taka forma relaksu jest dobra, czy też nie, ale pewne jest to, że nie da się uciec od galopującego rozwoju technologicznego.

Celem pracy dyplomowej jest stworzenie gry komputerowej w środowisku Unity. Program dedykowany jest na urządzenia mobilne z jednym z najpopularniejszych systemów operacyjnych na rynku, systemem Android. Gra będzie polegała na zaangażowaniu użytkownika poprzez ruch telefonem, aby przesuwając platformę do odbijania piłki i celowania w blok, który zapewni zwycięstwo.

Dalsza część pracy wygląda następująco. W drugim rozdziale zostanie przedstawiona bardziej szczegółowa analiza tematu wraz z historią gier komputerowych oraz gier mobilnych, jak również pokazane zostaną podobne, już istniejące, gry komputerowe. W rozdziale trzecim omówione zostaną wymagania dotyczące aplikacji jak również narzędzia wykorzystane do jej stworzenia. Rozdział czwarty został poświęcony specyfikacji zewnętrznej aplikacji. W piątym rozdziale przedstawiony zostanie sposób implementacji gry jak również pokazane tworzenie multimedialnych elementów. Rozdział szósty skupia się na opisie testowania oraz możliwościach dalszego rozwoju aplikacji. Ostatni rozdział zawiera podsumowanie całej pracy.

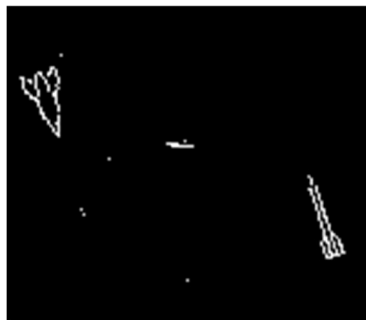
Rozdział 2

Analiza tematu

Gry towarzyszyły ludziom niemal od początku. Jest to fundamentalna część naszej egzystencji [1]. Przed wynalezieniem komputerów istniało mnóstwo wspaniałych form rozrywki, które pozwalały ćwiczyć logiczne myślenie oraz koncentracja, ale również służyły do odstresowania się, oraz dobrej zabawy w pojedynkę lub w gronie znajomych czy też rodziny. Wśród graczy powszechna jest również chęć rywalizacji dzięki czemu powstało również wiele turniejów i tym podobnych wydarzeń gdzie najlepsi mogą ze sobą konkurować. Wraz z postępem technologicznym znaczna część gier przeniosła się do świata wirtualnego, co umożliwia zmierzenie się z przeciwnikami z różnych części świata.

2.1. Historia gier komputerowych

Można zaryzykować stwierdzenie, że historia gier komputerowych jest po części historią rozwoju technologii. Od wynalezienia pierwszego komputera w 1946 roku [2] do wydania gry komputerowej potrzeba było 16 lat, gdyż za pierwszą grę uważany jest Spacewar! (Rys. 1). Została zaprojektowana w 1962 roku przez MIT [3]. Twórcą był Stephen Russell. Oryginalnie program działał na komputerze PDP-1 a jej celem było sterowanie swoim statkiem kosmicznym i próba zestrzelenia statku przeciwnika. Była to pierwsza gra, ale z uwagi na rozmiar oraz cenę sprzętu, którego wymagała, nie była powszechnie dostępna.



Rys. 1 Pierwsza gra komputerowa Spacewar! [3]

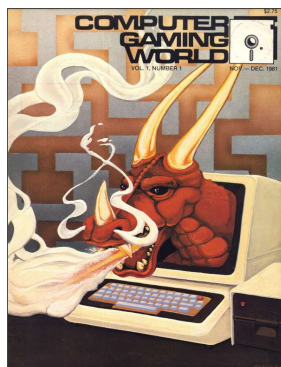
Od tego wydarzenia minęła dekada, zanim pojawiły się pierwsze automaty pozwalające na wejście do wirtualnego świata rozrywki. Jedyne co trzeba było zrobić to wrzucić monetę do maszyny i można było się cieszyć rozgrywką. Niespełna rok później została wydana pierwsza gra, którą można było kupić do swojego domu. Mowa tutaj o PONG-u (Rys. 2) stworzonym przez Nolana Bushnellsa w 1972 roku [4].



Rys. 2 Gra "Pong" [3]

Wydarzenia z ubiegłych lat doprowadziły do masowego rozwoju branży gier video dzięki firmom takim jak Chicago coin, Midway czy Kee Games, które było własnością Atari. Doprowadziło to do wynalezienia pierwszej konsoli, która w przeciwieństwie do dotychczas sprzedawanych urządzeń z wbudowaną w sprzęt grą włączającą się wraz z dopływem zasilania, posiadała wymienne kasety, co umożliwiało znacznie tańszą dystrybucję tego rodzaju oprogramowania. Osoba posiadająca już konsole mogła bez problemu wymieniać gry bez potrzeby zakupu nowego urządzenia. Pierwszy taki system został wprowadzony w 1976 roku jako Video Entertainment System.

Z początkiem lat 80 zaczęły pojawiać się czasopisma takie jak Computer Gaming World (Rys. 3) zawierające kody gier, które można było przepisać ręcznie do komputera i zacząć rozgrywkę [5]. Programy te były tworzone głównie przez hobbistów i wysyłane aby, umieścić je w artykule.



Rys. 3 Okładka pierwszego wydania czasopisma "Computer Gaming World" [5]

Do szybkiego rozwoju gier przyczynił się również IBM, który wypuścił swój pierwszy komputer personalny w 1981 roku [6]. Zaczęto modyfikować istniejące już programy oraz pisać nowe, aby były kompatybilne z tym, jak na swoje czasy, bardzo potężnym komputerem. Na początku był używany głównie przez biznesmenów, którzy przyznali się do grania w gry na ich komputerach. Czasopismo BYTE uznało maszyny IBM za wspaniałe urządzenia, które idealnie nadawały się do gier video [7]. Wypuszczono na nie takie tytuły jak Microsoft Flight Symulator, czyli znaną i kontynuowaną do dziś serią symulatora lotów, czy Microsoft Adventure, która była świetną grą fantastyczną.

Przemysł gier rozwijał się bardzo szybko, jak również sama technologia komputerowa. Wcześniej już wspomniane czasopismo Computer Gaming World, w numerze z 1989 roku, ogłosiło, że VGA (Video Graphics Array), czyli technologia umożliwiająca wyświetlanie większej ilości kolorów, jak i lepszej rozdzielczości, będzie wdrażana do nowych gier, co umożliwi lepszą jakość rozgrywki niż dotychczas [8].

2.2. Początek gier mobilnych

W dzisiejszych czasach „grą mobilną” określa się gry na urządzenia mobilne, przede wszystkim telefony komórkowe. Pionierem w tej dziedzinie jest kultowy Tetris, (Rys. 4) który zadebiutował w 1994 roku. Był on preinstalowaną aplikacją na telefonie Hagenuk MT-



Rys. 4 Gra Tetris na telefonie Hagenuk MT-200

źródło:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Ftekdeeps.com%2Fthe-evolution-of-mobile-games-komorkomania-pl%2F&psig=AOvVaw0is6j8xhIcZ8lw16WrmW1l&ust=1672774127461000&source=images&cd=vfe&ved=0CA0QjRxqFwoTCIinhNvPqfwCFQAAAAAdAAAAABAD>

200 [9]. Nie jest to jednak gra, która najbardziej kojarzy się z początkiem branży, ponieważ prawdziwą sensację sprawił Snake, który został wypuszczony na telefony Nokia trzy lata później. Można go znaleźć na ponad 400 milionach urządzeń [10].

Sukces firmy Nokia spowodował, że powstała duża konkurencja o rynek gier na telefony komórkowe, przez co rynek rozwijał się bardzo szybko. Końcem lat 90 technologia WAP (Bezprzewodowy protokół aplikacji) umożliwiła pobieranie i instalowanie aplikacji mobilnych „przez powietrze”. Powstały również gry bazujące na SMS, których koszt był wliczany do rachunku u operatora.

Pomimo możliwości pobierania i instalowania gier na telefony, nie było to zbyt szybkie ani wygodne. Nie istniał żaden sposób, aby przeglądać duże biblioteki gier i wybierać co przypadło komuś do gustu. Rewolucyjne okazało się wyjście iPhone firmy Apple w 2007 roku oraz wydaniu App Store w 2008 roku. Aplikacja ta służyła do łatwej dystrybucji oprogramowania na telefony mobilne, co bardzo spopularyzowało wszelkiego rodzaju gry mobilne. Alternatywą na telefony z systemem android, dla sklepu firmy Apple, stał się Google Play, który oryginalnie został wypuszczony w 2008 roku jako Android Market. Dwa lata później ukazał się Windows Phone Store pod pierwotną nazwą Windows Phone Marketplace.

Od pojawienia się mobilnych sklepów z oprogramowaniem branża gier mobilnych gwałtownie się powiększa, lecz nie wprowadza już żadnych przełomowych innowacji, które zmieniłyby oblicze rynku, jaki jest znany obecnie.

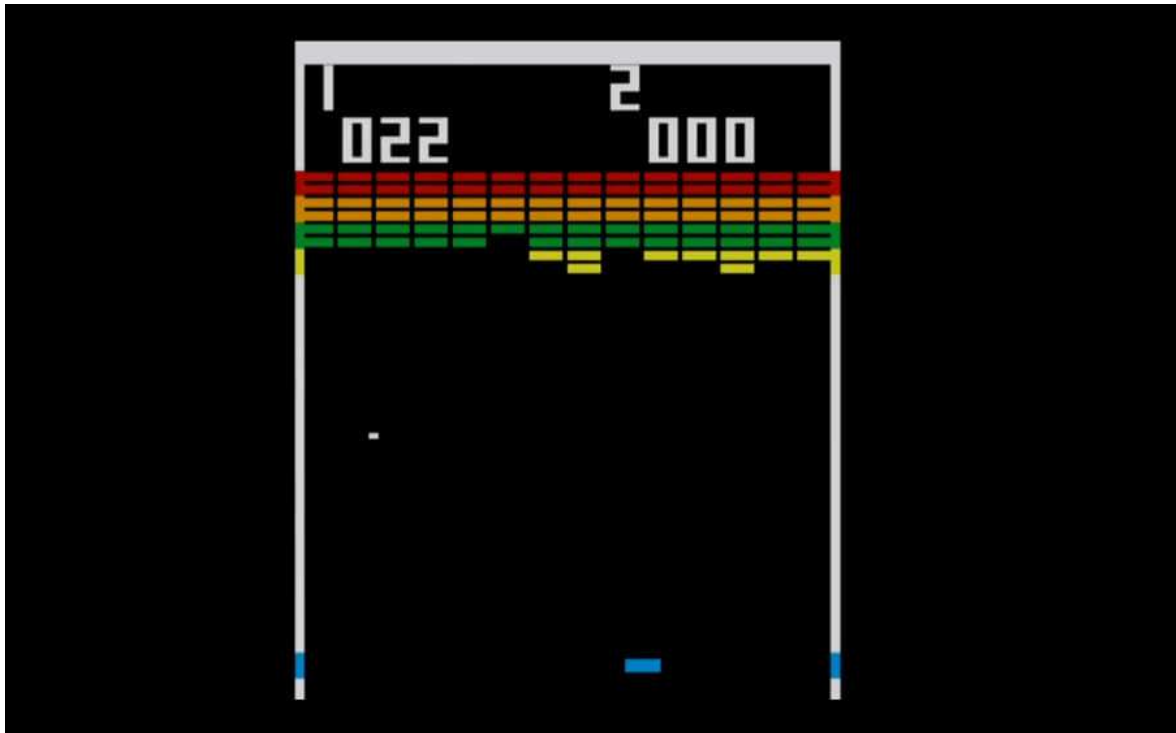
2.3. Gra arcade

Na początku trzeba się zastanowić, czym są gry Arcade. Jest to bardzo stary gatunek pochodzący z lat 70 i 80. W tamtym okresie były to automaty, do których wrzucało się monety i można było grać przez określony czas zależnie od typu maszyny. Grafika była bardzo prosta tak jak i efekty dźwiękowe, więc twórcy musieli to nadrobić poprzez ciekawą i wciągającą rozgrywkę lub historię, która była tam opowiedziana [11]. Mimo prostoty tych gier, zwolennicy takiego typu rozgrywki do dziś szukają miejsc, gdzie można znaleźć takie automaty, lub korzystają z alternatywy w postaci nowych gier tego typu na swoje telefony oraz komputery stacjonarne.

2.4. Gra typu Block Breaker

Gra typu Block Breaker należy do gatunku arcade. Pierwsza tego typu gra pojawiła się w 1976 roku i nosiła tytuł Breakout (Rys. 5). Jej głównym celem było rozbijanie cegieł za pomocą piłki, która była odbijana paletką sterowaną przez gracza.

Gra miała głównie symbolizować ucieczkę z więzienia, a gracz miał się wcielić w więźnia, który niszczy mury. Była ona oryginalnie zaprojektowana przez Steve Wozniaka i Steve Jobsa na jeden z automatów [12].



Rys. 5 Gra Breakout [12]

2.5. Akcelerometr w grach

Akcelerometr to przyrząd służący do mierzenia przyspieszenia. Działa on na zasadzie bezwładnego ciężarka, który reaguje przy zmianach prędkości urządzenia [13]. Tego rodzaju komponent jest instalowany w prawie wszystkich nowoczesnych telefonach czy tabletach. Jest wykorzystywany na przykład do zmiany orientacji ekranu na pionowy lub poziomy.

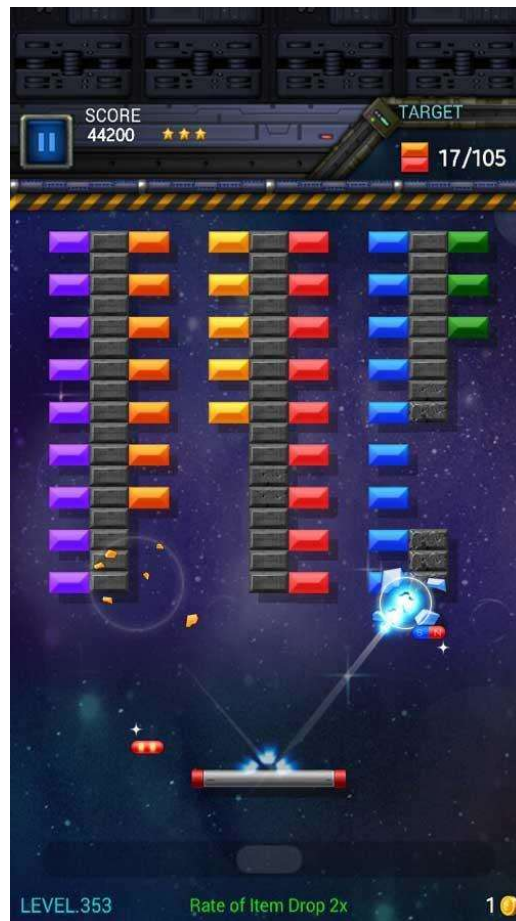
W grach na urządzenia mobilne, często wykorzystuje się akcelerometr aby bardziej zaangażować gracza w rozgrywkę oraz wprowadzić elementy zręcznościowe. Przykładem gdzie wykorzystuje się takie rozwiązanie jest Accelerometer Ball. Została wydana na

Google Play przez Mateusza Jaworskiego. Rozgrywka polega na nawigowaniu piłką po labiryncie za pomocą ruchów swoim urządzeniem. Tego typu rozwiązania zazwyczaj nie są implementowane, lecz zapewniają doskonałą rozrywkę.

2.6. Podobne tytuły na rynku

2.6.1. Brick Breaker Star

Brick Breaker (Rys. 6) Star jest grą typu Block Breaker o prostym sterowaniu i mechanice gry. Wszystkie elementy poziomu są widoczne od początku rozgrywki. Zadaniem gracza jest pozbyć się wszystkich możliwych bloków w celu ukończenia



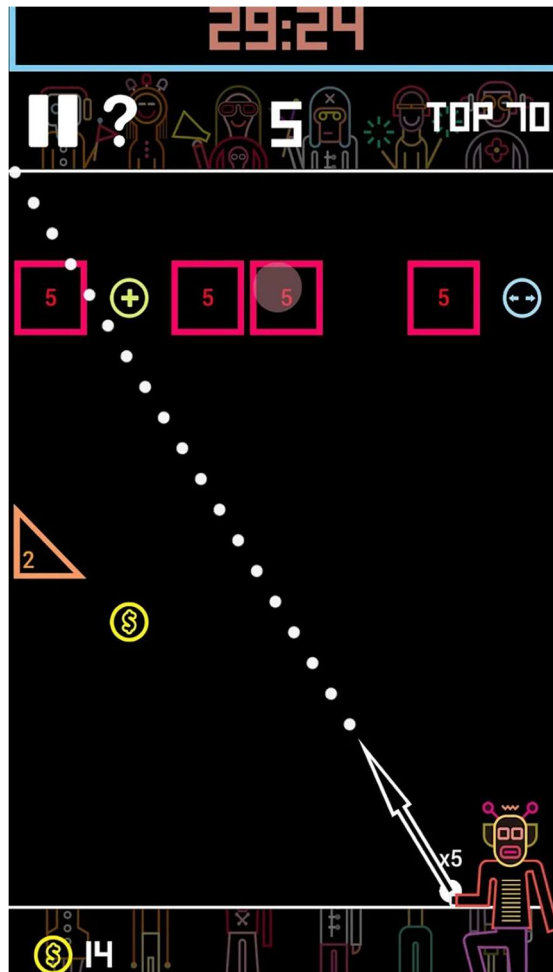
Rys. 6 Zrzut z gry Brick Breaker Star

Źródło:

<https://play.google.com/store/apps/details?id=com.lt.latte.brick&hl=en&gl=US>

poziomu. Bloki usuwane są, gdy zostaną trafione piłką. Gracz steruje panelem, przesuując palcem po ekranie, jego celem jest uniemożliwienie spadnięcia piłki poza obszar gry. Podczas zbijania bloków użytkownik jest w stanie zdobyć różne ulepszenia ułatwiające rozgrywkę oraz monety, które później można wykorzystać na przedłużenie trwania danego ulepszenia. Gracz poza ulepszeniami w trakcie poziomu, może też trafić na utrudnienia takie jak zmniejszona powierzchnia platformy. Ciekawą funkcją jest również znaczne spowolnienie czasu, gdy gracz podniesie palec z ekranu.

Brick Breaker Star jest grą bardzo rozbudowaną i posiada 500 poziomów, których trudność wzrasta wraz z postępem gracza. Pojawiają się nowe typy bloków, których głównym zadaniem jest utrudnić rozgrywkę. Występuje również system mikropłatności z którego pomocą można wykupić dodatkowe ulepszenia, lecz nie jest on konieczny w celu zwykłej rozgrywki i gra może równie dobrze działać bez dostępu do internetu.



Rys. 7 Zrzut ekranu z gry BBTAN
Źródło:
<https://play.google.com/store/apps/details?id=com.crater.bbtan&hl=en&gl=US>

2.6.2. BBTAN

BBTAN (Rys. 7) jest grą typu Block Breaker działającą w trybie nieskończonym. Znaczy to że gra nie ma poziomów, a trwa dopóki gracz nie przegra rozgrywki. Przebieg gry polega na tym, że gracz wypuszcza piłkę i próbuje zniszczyć jak największą ilość bloków. Gdy piłka skończy się odbijać wraca do pola początkowego i dokładana jest nowa pula bloków, a licznik punktów gracza zwiększa się o jeden wraz z każdą dodaną pulą. Każda fala bloków generowana jest losowo, a ilość potrzebnych uderzeń w każdej nowej puli jest o jeden większa od poprzedniej. Podczas gdy piłka odbija się od bloków i ścian może ona zdobyć różne ulepszenia takie jak: dodatkowa piłka, strzał laserowy w pionie lub poziomie oraz monety. Sterowanie w grze polega na przytrzymaniu palca na ekranie i wystrzeleniu piłki w wybranym kierunku, kiedy puścimy go z ekranu. Rozgrywka kończy się w momencie, gdy graczowi nie uda się zniszczyć bloków zanim dostaną się one na dół ekranu. Celem gracza jest osiągnięcie jak najwyższego wyniku. Z pomocą monet zdobywanych gracz może zakupić nowe piłki o różnych rozmiarach i kolorach. Wykup nowej piłki kosztuje sto monet, a piłka jest wybrana losowo.

BBTAN nie wymaga połączenia z siecią ani nie posiada systemu mikro transakcji.

Rozdział 3

Wymagania i narzędzia

Do stworzenia gry komputerowej nie potrzeba wiele programów lub innych narzędzi. Jeszcze kilkadziesiąt lat temu, większość tego typu programów było napisane przy użyciu jedynie edytora tekstu. Jednak w obliczu coraz to nowszych technologii powstało dużo programów umożliwiających tworzenie bardziej skomplikowanych gier w znacznie krótszym czasie.

3.1. Przegląd narzędzi

3.1.1. Unity

Unity jest oprogramowaniem umożliwiającym tworzenie gier oraz innego rodzaju aplikacji interaktywnych w technologii 2D lub 3D. Jego kod źródłowy został napisany głównie w języku C++. Programy takie jak Unity są nazywane silnikami ponieważ dostarczają główny kod gry wraz ze środowiskiem programistycznym oraz łączą ze sobą wszystkie komponenty i pozwalają na wzajemne oddziaływanie między nimi.

Początkowo skrypty do obsługi elementów można było pisać w UnityScript, Boo jak również w C#. Wraz z pojawianiem się nowszych wersji przestano wspierać najpierw język Boo a później język UnityScript, jednakże dalej można je skompilować, lub korzystać z nich w jakiś inny sposób. Najlepszym jednak rozwiązaniem pozostaje pisanie w C#, gdyż jest nadal wspierany.

Unity jest darmowym oprogramowaniem dla wszystkich twórców których roczne dochody nie przekraczają stu tysięcy dolarów, czyli jest idealnym rozwiązaniem dla entuzjastów jak i małych firm. Dużym atutem jest również bardzo dobrze wyposażony Asset Store w którym można kupić potrzebne elementy bez konieczności ich tworzenia.

3.1.2. Visual Studio

Visual Studio to zintegrowane środowisko programistyczne stosowane do tworzenia oraz rozwoju oprogramowania takiego jak programy komputerowe, strony internetowe, czy aplikacje mobilne. Zostało stworzone przez firmę Microsoft co daje dostęp do wielu platform tej firmy.

To narzędzie zawiera w sobie edytor tekstu ze wsparciem inteligentnego uzupełniania, co bardzo ułatwia i przyspiesza pracę. Znajduje się też zintegrowany debugger umożliwiający szybkie wykrywanie błędów w pisany programie.

Środowisko Visual Studio wspiera całą gamę języków programowania takich jak C#, Java Script, jak również HTML i CSS. Sumarycznie można korzystać z 36 różnych języków.

3.1.3. GIMP

Program GIMP jest darmowym programem, którego kod jest udostępniony publicznie dla wszystkich użytkowników. Stosuje się go do kreowania grafik rastrowych lub edycji już istniejących. Bez narzędzi umożliwiających tworzenia wizualnych assetów, gry były by mało atrakcyjne dla potencjalnych konsumentów.

3.1.4. Stable Diffusion

Wraz ze wzrostem popularności stosowania sztucznej inteligencji w wielu dziedzinach nastąpił jej gwałtowny rozwój. Stable Diffusion to model stworzony metodą głębokiego uczenia. Jest to bardzo nowy program ponieważ wypuszczony w 2022 roku. Pozwala na generowanie szczegółowych grafik poprzez wprowadzenie opisu oczekiwanego rezultatu. Jest również opcja wspomagania algorytmu poprzez dodanie bardzo surowej grafiki tego czego oczekujemy, przez co dostaniemy bardziej dokładny wynik.

Stable Diffusion jest napisane w języku Python i jest darmowym odpowiednikiem płatnych programów takich jak Dall-e. Pozwala na tworzenie różnych rozmiarów obrazków, chociaż najlepiej spisuje się na domyślnych ustawieniach 512 na 512 pixeli. Do działania wykorzystuje kartę graficzną, lecz zalecane jest używanie takiej, która posiada minimum 10GB video RAM.

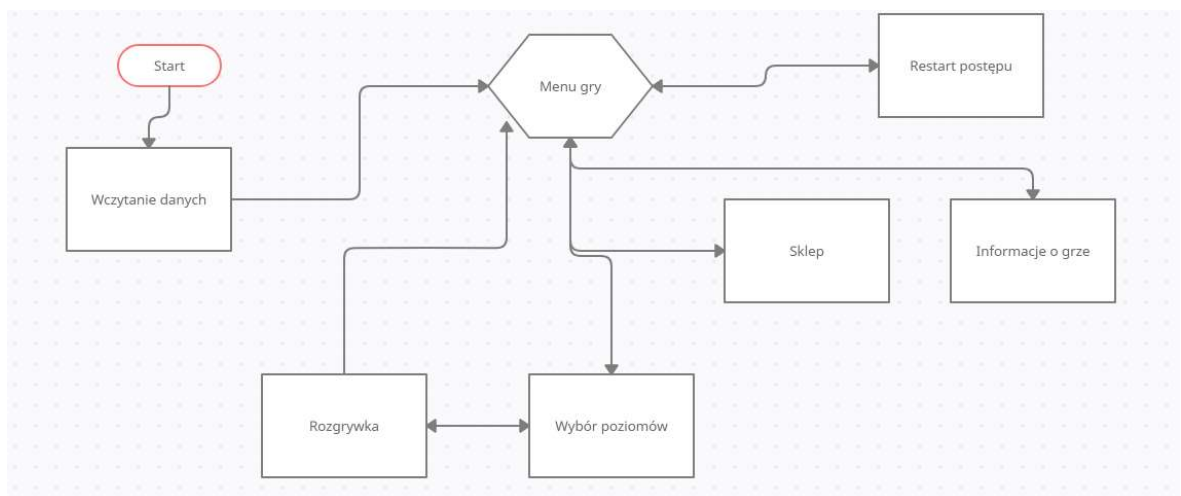
3.1.5. Audacity

Niemal każda gra komputerowa zawiera w sobie jakieś efekty dźwiękowe. W tym projekcie do tworzenia i obróbki dźwięku został użyty program Audacity, ponieważ posiada bardzo dużo funkcjonalności, przy czym jest oprogramowaniem typu open-source i jest darmowy. Pozwala na nagrywanie i modyfikowanie dźwięku oraz zapisania go w różnych dostępnych formatach.

3.2. Wymagania w stosunku do gry

3.2.1. Wymagania funkcjonalne

Większość gier posiada bardzo podobne schematy dotyczące przechodzenia pomiędzy poszczególnymi scenami. Tak jak na zamieszczonym grafie (Rys. 8), w centrum znajduje się menu użytkownika. Zadaniem głównej strony jest kierowanie gracza do wszystkich funkcjonalności które oferuje dana gra.



Rys. 8 Przebieg interakcji

W przypadku tego projektu gra powinna umożliwiać poruszanie się po poszczególnych funkcjonalnościach dzięki głównemu menu gry. Znajdować się tam mają przejścia do sklepu, informacji o zasadach gry, zresetowania postępu oraz do wyboru poziomów.

Reset postępu ma za zadanie wyczyścić wszystkie zapisane informacje na temat osiągnięć które uzyskał dotychczas gracz.

W sklepie znajdować się będzie kilka przedmiotów kosmetycznych do zakupu za zebrane w grze złoto.

W informacji o grze znajdować się będą informacje dotyczące sposobu wygrania gry oraz zachowania poszczególnych bloków

Wybór poziomów ma umożliwić przejście do wybranej przez użytkownika planszy, oraz zawierać mechanizm odblokowywania poziomu po jego przejściu. Powinny być 2 rodzaje poziomów, jeden generowany losowo i kilkanaście, wcześniej ułożonych i powtarzalnych.

W rozgrywce użytkownik powinien mieć możliwość sterowania paletką za pomocą obracania telefonu na boki. Po wygraniu lub przegraniu poziomu użytkownik może przejść do wyboru poziomów, lub do menu głównego.

W aplikacji muszą znajdować się efekty dźwiękowe oraz muzyka w grze.

3.2.2. Wymagania niefunkcjonalne

Użytkownicy aplikacji mają mieć możliwość włączenia gry na dowolnym urządzeniu z systemem Android 7.0 lub nowszym. Program nie będzie wymagał połączenia z Internetem do poprawnego funkcjonowania oraz nie będzie ograniczał żadnych funkcjonalności.

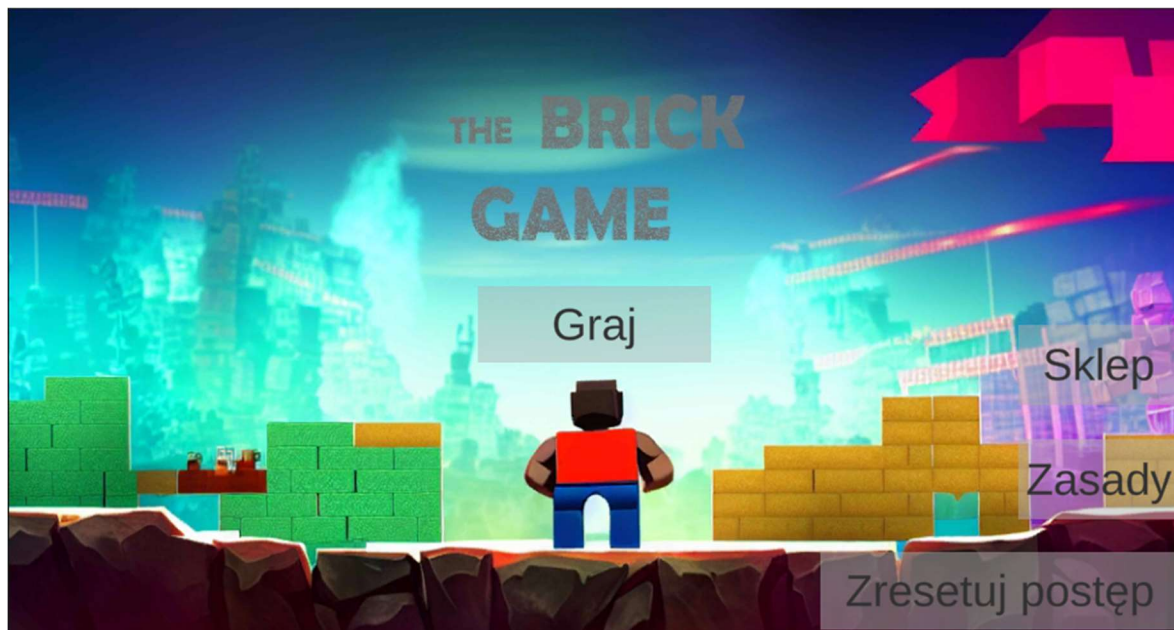
Gra ma zawierać grafikę generowaną przez algorytmy sztucznej inteligencji by poprawić aspekty wizualne.

Rozdział 4

Specyfikacja zewnętrzna

Gra typu Block Breaker jest jedną z gier Arcade. Jej głównym celem jest zniszczenie głównego bloku zwanego sercem. Blok można zniszczyć poprzez trafienie w niego piłką wystrzeloną z paletki. Paletka, zwana również platformą, jest sterowana przez użytkownika za pomocą przechylania telefonu w prawo lub lewo.

4.1. Menu główne



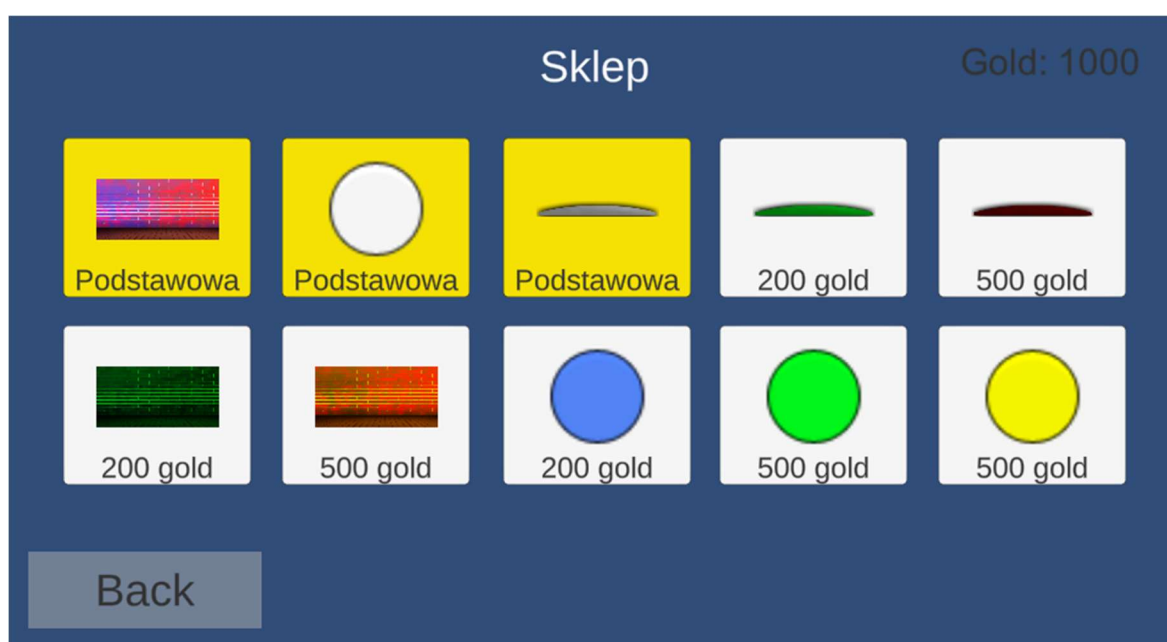
Rys. 9 Widok menu głównego

Po włączeniu gry, pierwszym ekranem ukazującym się użytkownikowi jest menu główne (Rys. 9). Zostaje również odtworzona muzyka tła która towarzyszy użytkownikowi na każdym kolejnym ekranie. W centrum znajduje się przycisk „Graj” przekierowujący użytkownika do wyboru poziomów.

Z prawej strony znajdują się kolenje trzy przyciski. Przycisk „Sklep” przekierowuje do sklepu z przedmiotami kosmetycznymi, natomiast przycisk „Zasady” otwiera widok zawierający opis panujących w grze zasad.

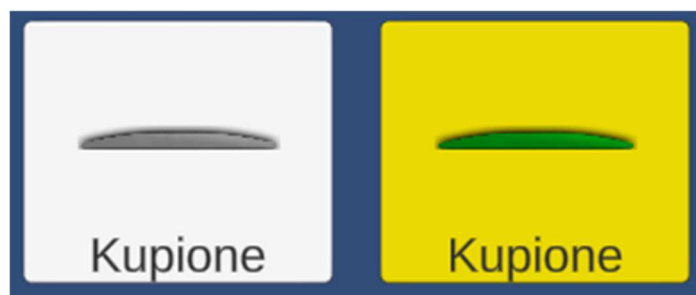
Jeżeli użytkownik wciśnie przycisk „Zresetuj postęp”, wtedy wszystkie zakupione przedmioty zostaną usunięte jak również postęp w przechodzeniu poziomów oraz zdobyte złoto.

4.2. Sklep



Rys. 10 Panel sklepu

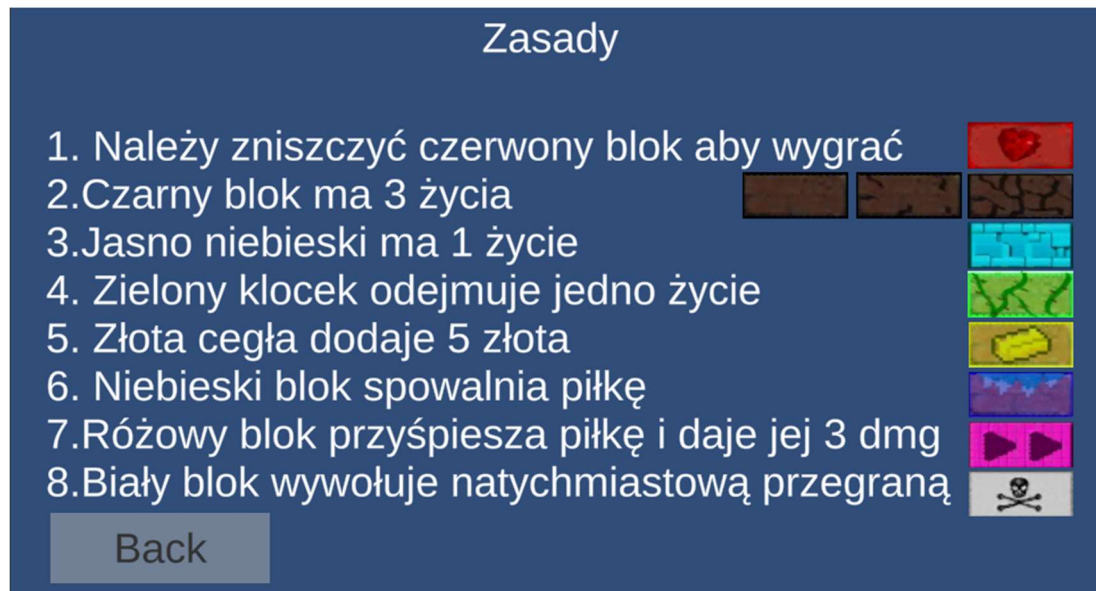
W panelu sklepu znajduje się 10 dostępnych przedmiotów (Rys. 10). Pierwsze trzy są domyślnymi przedmiotami i są dostępne na samym początku gry. Użytkownik może kupić dowolny przedmiot, jeżeli posiada wystarczającą ilość złota. Jeżeli użytkownik posiada



Rys. 11 Widok kupionego oraz wybranego elementu

kupiony przedmiot, to w ekranie sklepu może go ponownie kliknąć aby ustawić go jako wybrany co spowoduje że będzie on używany podczas rozgrywki (Rys. 11).

4.3. Zasady

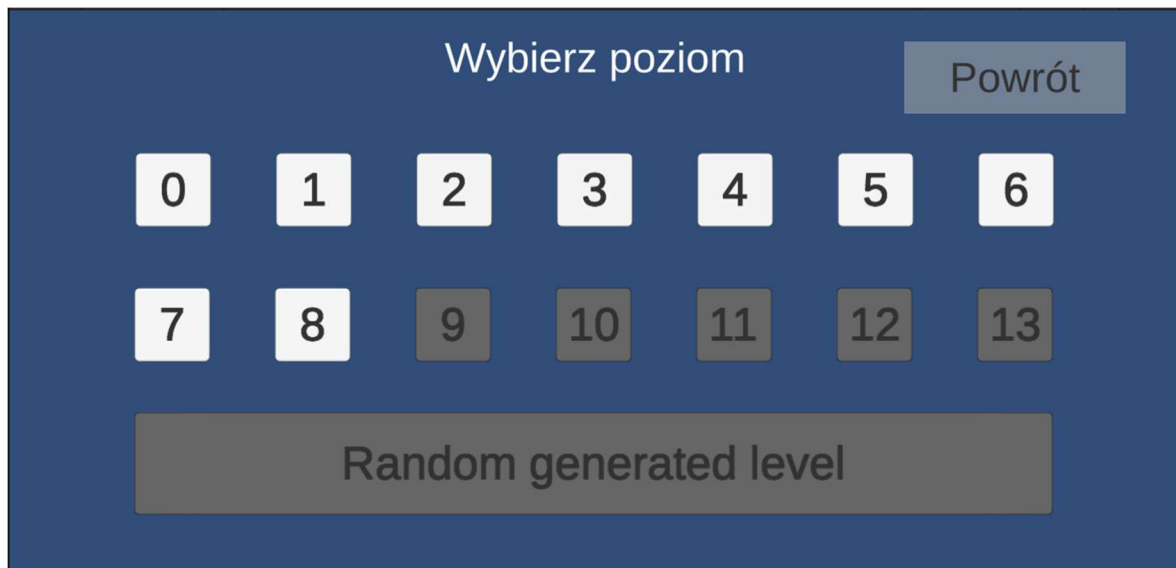


Rys. 12 Wygląd ekranu "Zasady"

Okno z zasadami przedstawia wszystkie rodzaje bloków dostępnych w grze (Rys. 12). Z prawej strony znajduje się grafika przedstawiająca blok, a po lewej opis działania bloku.

4.4. Wybór poziomów

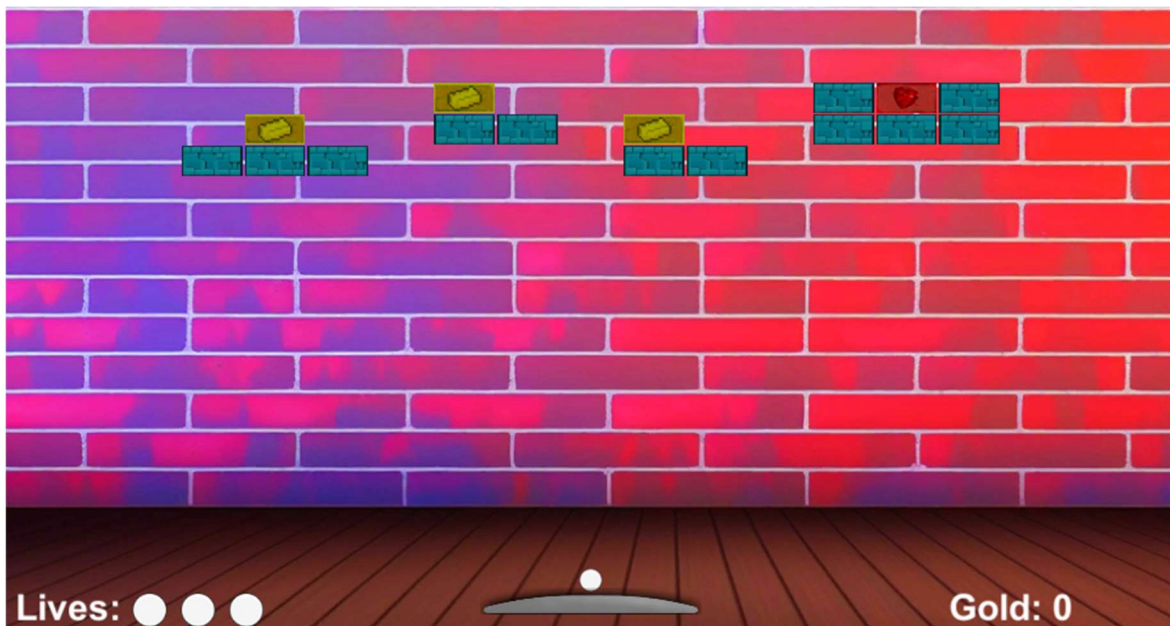
Wybór poziomów przez użytkownika jest realizowany w oknie „Wybór poziomów” (Rys. 13). Możliwe jest wybranie dowolnej odblokowanej planszy, czyli tej która jest podświetlona na białą. Aby odblokować dany poziom, użytkownik musi wcześniej ukończyć poprzedni. Istnieje również wybór losowo generowanego poziomu, który zostaje udostępniony po skończeniu wszystkich pozostałych plansz. Poziom zerowy jest poziomem testowym pokazującym działanie wszystkich dostępnych bloków. Z każdym kolejnym poziomem zostaje zwiększona trudność jak i zaczynają pojawiać się nowe bloki utrudniające rozgrywkę.



Rys. 13 Okno przedstawiające wybór poziomów

4.5. Okno gry

Jeżeli gracz wybrał już poziom to zostanie przekierowany do okna gry (Rys. 14). Zadaniem gracza jest nawigowanie szarą platformą za pomocą przechylania swojego telefonu na boki. Piłka jest przyczepiona do paletki do momentu kliknięcia na ekran. Podczas kolizji piłki z jakimkolwiek elementem otoczenia słychać dźwięk uderzenia. W lewym



Rys. 14 Okno przedstawiające pierwszy poziom

dolnym rogu znajduje się licznik żyć. Każdorazowo gdy gracz straci życie, poprzez upuszczenie piłki na podłogę, zostaje odjęta jedna piłka. Gdy straci wszystkie trzy życia, wtedy pojawia się ekran porażki. Jeżeli użytkownikowi uda się trafić w czerwoną cegielkę zwaną „sercem”, to oznacza przejście poziomu i pojawienie się ekranu wygranej (Rys. 15). Z obu ekranów można przejść do menu głównego oraz wyboru poziomów. Ekran przegranej (Rys. 16) umożliwia również zaczęcie bieżącego poziomu od początku.



Rys. 15 Wygląd ekranu wygranej



Rys. 16 Widok ekranu porażki

W grze pojawia się osiem różnych rodzajów bloków. Najważniejszym z nich jest blok serca, który daje graczowi zwycięstwo. Następnym i najbardziej podstawowym blokiem jest cegła w kolorze błękitnym. Ma ona tylko jeden punkt wytrzymałości, co oznacza że należy ją dotknąć piłką tylko jeden raz aby została zniszczona i nie posiada żadnych innych właściwości. Blok złoty po zniszczeniu daje graczowi pięć złota, które może wykorzystać do zakupu dodatków kosmetycznych w sklepie. W grze istnieje również blok posiadający więcej punktów wytrzymałości, a jest to blok czarny. Gracz musi trafić w niego trzy razy aby go zniszczyć, a jego wytrzymałość jest obrazowana poprzez pojawiające się na nim pęknięcia. W grze występują również dwa bloki funkcyjne. Różowy blok przyspiesza piłkę ale w zamian daje jej trzy obrażenia co umożliwia zniszczenie czarnego bloku przy jednym uderzeniu. Niebieski blok odpowiedzialny jest za spowolnienie piłki co zapewnia lepsze panowanie nad nią i ułatwia grę. Efekty jednak nie trwają w nieskończoność i kończą się po dziesięciu sekundach. Najbardziej niebezpieczne bloki są w kolorze zielonym oraz białym. Zielona cegła po trafieniu odbiera graczowi jedno życie, natomiast biały blok z czaszką powoduje natychmiastową przegraną.

4.6. Wymagania dotyczące telefonu oraz instalacja

Grę można zainstalować na każdym urządzeniu mobilnym z systemem operacyjnym android w wersji 7.0 lub nowszej. Urządzenie dodatkowo musi być wyposażone w akcelerometr oraz żyroskop aby można było sterować paletką.

Aby zainstalować aplikację należy pobrać plik z rozszerzeniem apk, a następnie zainstalować aplikację na swoim urządzeniu.

Rozdział 5

Specyfikacja wewnętrzna

Cały projekt został wykonany z wykorzystaniem narzędzia Unity. Wszystkie skrypty są napisane w języku c#. Do stworzenia elementów graficznych wykorzystano program Gimp oraz generowanie obrazów za pomocą programu Stable Diffusion. Dźwięki do gry przygotowano za pomocą narzędzia Audacity. Główna piosenka, która jest grana w tle to darmowy utwór autora Nojisuma.

5.1. Przechowywanie zmiennych

Przenoszenie zmiennych pomiędzy ekranami w Unity może być wykonane na wiele różnych sposobów. W projekcie została utworzona klasa MenuVars, która przechowuje wszystkie zmienne do których będzie potrzebny dostęp w innych oknach. Zawiera on również funkcję DontDestroyOnLoad, która blokuje usuwanie obiektu gry podczas przeładowywania okien, co umożliwia przechowywanie danych tam gdzie jest to potrzebne. Obiekt gry w środowisku Unity jest podstawowym elementem do którego przyczepiane są komponenty nadające mu pożądane właściwości i zachowania. Komponentem nazywamy na przykład skrypty opisujące zachowanie obiektu, wbudowane pola kolidujące, czy elementy opisujące własności fizyczne obiektu.

Obiekt do którego przypięta jest klasa przechowująca zmienne, ma również inne zastosowanie. Jako iż przechodzi on pomiędzy oknami, to umożliwia dodanie komponentu odtwarzającego dźwięk, przez co muzyka z tła jest odtwarzana niezależnie od okna w którym znajduje się obecnie gracz.

5.2. Klasy interfejsu

5.2.1. Klasa przejść

Główną klasą odpowiedzialną za zarządzanie przejściami między oknami gry jest TransScript. Znajdują się w niej metody LoadLevel, LoadTransition, ReLoadLevel i

PlayLevel odpowiedzialne za przełączanie pomiędzy oknami, otwieranie okien z danym poziomem oraz aktywacji animacji przejścia.

Metoda LoadLevel przyjmuje jako parametr numer okna które powinno zostać wyświetlona i po wywołaniu tej metody następuje przejście do zadanej sceny. W Unity sceny, to osobne okna pomiędzy którymi można się poruszać. W przypadku tej metody nie jest wyświetlana animacja przejścia.

Metoda LoadTransition działa podobnie jak LoadLevel ale dodatkowo wykonuje animację przejścia, w tym przypadku jest to przyciemnienie ekranu, a następnie przełącza okno po skończeniu animacji.

Metoda ReLoadLevel służy do ponownego wczytania sceny z grą z obecnie rozgrywanym poziomem. Gra zostaje wtedy rozpoczęta na nowo, ze świeżą planszą oraz wszystkimi życiami.

Metoda PlayLevel służy załadowania poziomu, który został podany w argumencie metody.

5.2.2. Klasa odblokowywania poziomów

Zachowanie okna Menu do wybierania poziomów jest opisane za pomocą klasy o nazwie LevelUnlocks. Służy ona do aktywacji przycisków w wyborze poziomów, bazując na najwyższym osiągniętym przez gracza poziomie.

Klasa posiada zmienną tablicową przechowującą wszystkie obiekty przycisków. Podczas załadowania okna Menu do wyboru poziomów, zostaje wywołana funkcja Start, która w pętli sprawdza jakie przyciski powinny być aktywne, bazując na zmiennej LevelReached, która przechowuje najwyższy poziom do którego dotarł gracz.

5.2.3. Klasa sklepu

Do obsługi sklepu użytkownika wykorzystywana jest klasa o nazwie TradeHandling. Jest on odpowiedzialny za poprawne działanie sklepu oraz zapisywanie zakupionych oraz wybranych elementów w pamięci programu aby nie zostały usunięte podczas ponownego uruchomienia aplikacji.

Klasa sklepu posiada trzy metody. Pierwsza z nich (Start) jest wywoływana z każdym otwarciem sklepu. Odpowiada ona za pobranie informacji o złocie oraz wywołaniu kolejnej metody o nazwie Reset.

Metoda Reset powoduje przeładowanie informacji o sklepie i ustawienie przycisków do ich prawidłowego stanu, czyli ustawia przyciski w stan do kupienia, kupiony lub wybrany w zależności od zapisanych w programie zmiennych.

Ostatnia metoda `BuyItem` jest wywoływana przez wciśnięcie przycisku odpowiedzialnego za zakup przedmiotu. Metoda ta sprawdza stan przycisku który został naciśnięty i w zależności od tego wykonywane są odpowiednie procedury. Jeżeli przedmiot który użytkownik chce kupić jest zapisany jako nie kupiony, to sprawdzany jest stan złota i zostaje on oznaczony jako zakupiony oraz złoto zostaje odjęte lub nic się nie dzieje, jeśli nie było wystarczająco złota. Gdy zostanie wciśnięty zakupiony przedmiot a nie był oznaczony jako wybrany, to zostaje zmieniony kolor przycisku na żółty oraz ustawiona zostaje zmienna, że ten przycisk jest aktywny.

5.3. Klasy odpowiedzialne za rozgrywkę

5.3.1. Klasa cegły

Cegła jest obiektem gry 2D, do którego przypięte są komponenty kolizji prostokątnej oraz klasa cegły `Brick_vars`.

Klasa cegły zawiera dwie zmienne. Pierwsza zmienna `hitpoints`, jest odpowiedzialna za określenie punktów życia bloku, czyli potrzebnej ilości uderzeń przez piłkę aby cegła została zniszczona. Druga zmienna `type`, jest typem numerycznym i zapisany jest w niej typ bloku. Typ bloku jest potrzebny w klasie piłki która obsługuje kolizję z obiektami.

5.3.2. Klasa piłki

Piłka to obiekt gry 2D. Do tego obiektu podpięty jest wbudowany komponent fizyki 2D oraz pole kolizji w kształcie okręgu. Zachowanie piłki opisuje klasa `Script_For_Ball` odpowiedzialną za poruszanie się piłki oraz za obsługiwanie kolizji piłki z otoczeniem.

Początkowo pozycja piłki jest zależna od pozycji paletki aż do momentu kliknięcia w ekran i wywołania metody pozwalającej na odczepienie piłki oraz nadania jej prędkości w przeciwnym kierunku do paletki.

Klasa piłki o nazwie `Script_For_Ball` wykorzystuje wbudowaną w Unity metodę wykrywającą kolizję z otoczeniem. Jeżeli zostanie wykryte zderzenie piłki z jakimkolwiek obiektem, wyemitowany będzie dźwięk uderzenia. W zależności od obiektu z którym piłka weszła w kolizję, zostaną uruchomione różne działania. Jeżeli wykryte zostało zderzenie ze ścianami bocznymi, sufitem lub paletką to nic się nie stanie poza odbiciem piłki od powierzchni. Po wykryciu zderzenia z blokiem, zostanie pobrany rodzaj bloku i dla każdego rodzaju zostanie wywołana inna metoda obsługująca zderzenie z blokiem. Jeśli piłka dotknie dolnego panelu to zostaje odjęte jedno życie. Po utracie wszystkich żyć następuje wywołanie funkcji `GameOver` odpowiedzialnej za obsłużenie przegrania.

Metoda `GameOver` zatrzymuje grę oraz odkrywa ekran z wiadomością o przegranej oraz przyciskami do powtórzenia poziomu lub wyjścia do menu głównego.

W klasie piłki występuje osiem metod odpowiedzialnych za wykonanie akcji dla ośmiu bloków. Metoda `hitDefaultBlock` jest wywoływana w wyniku uderzenia w podstawowy blok przez piłkę. Na początku odejmowana jest liczba obrażeń piłki od wytrzymałości bloku, jeżeli został on zniszczony, zostaje wygenerowany emiter cząsteczek Unity, który symuluje rozpad bloku na kawałki, oraz obiekt bloku zostaje usunięty. Dla każdego kolejnego bloku metody zawierają dodatkowe instrukcje takie jak dodanie złota w przypadku złotego bloku (`hitYellowBlock`), czy odjęcie życia w przypadku bloku białego (`hitWhiteBlock`) i zielonego (`hitGreenBlock`). Funkcja bloku w kolorze fukcji (`hitMagentaBlock`) oraz niebieskiego (`hitBlueBlock`) przyspieszają lub spowalniają piłkę, przy czym wywołują dodatkową funkcję z opóźnieniem dziesięciosekundowym (`delay_blue` dla spowolnienia i `speed_magenta` dla przyspieszenia), która przywraca początkowy stan przed modyfikatorem. Uderzenie w blok czerwony spowoduje wywołanie metody `Victory`.

Metoda `Victory` zatrzymuje grę oraz wyświetla ekran zwycięstwa z przyciskami do wyboru poziomów oraz powrotu do menu głównego. W tej funkcji sprawdzane jest również najwyższy dotąd poziom, do którego doszedł gracz. W przypadku kiedy obecny poziom jest najwyższy, to postęp zostaje zapisany w plikach gry aby nie został stracony przy ponownym uruchomieniu.

5.3.3. Klasa paletki

Zachowanie obiektu paletki opisuje klasa `Paddle`. Jest ona odpowiedzialna za poruszanie się paletki. Występuje w niej jedna metoda, która jest wywoływana w każdej klatce działania gry. Podczas jej wywołania jest pobierana wartość akcelerometru a następnie mnożona przez zmienną prędkości oraz globalną zmienną `deltaTime` która jest wbudowana w środowisku Unity, a jej zadaniem jest zapewnienie prawidłowej prędkości paletki niezależnie od ilości klatek na sekundę, ponieważ funkcja jest wykonywana w każdej klatce, więc bez tej zmiennej, prędkość różniła by się w zależności od ilości wywołań funkcji na sekundę.

5.3.4. Klasa usuwania postępu

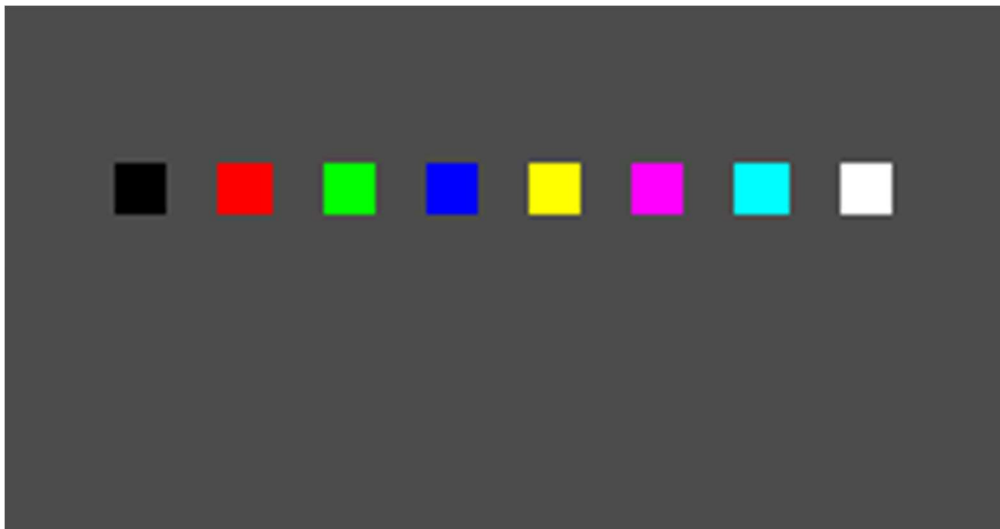
Do resetowania postępu w grze służy klasa `ResetProgress`. Zawiera ona jedną metodę o nazwie `HardReset`, która jest wywoływana po przez naciśnięcie przycisku w menu głównym o nazwie `Zresetuj Postęp`. Ta funkcja usuwa wszystkie zapisane wcześniej dane, a następnie przypisuje domyślne wartości do parametrów obiektów gry.

5.3.5. Klasa generowania poziomów

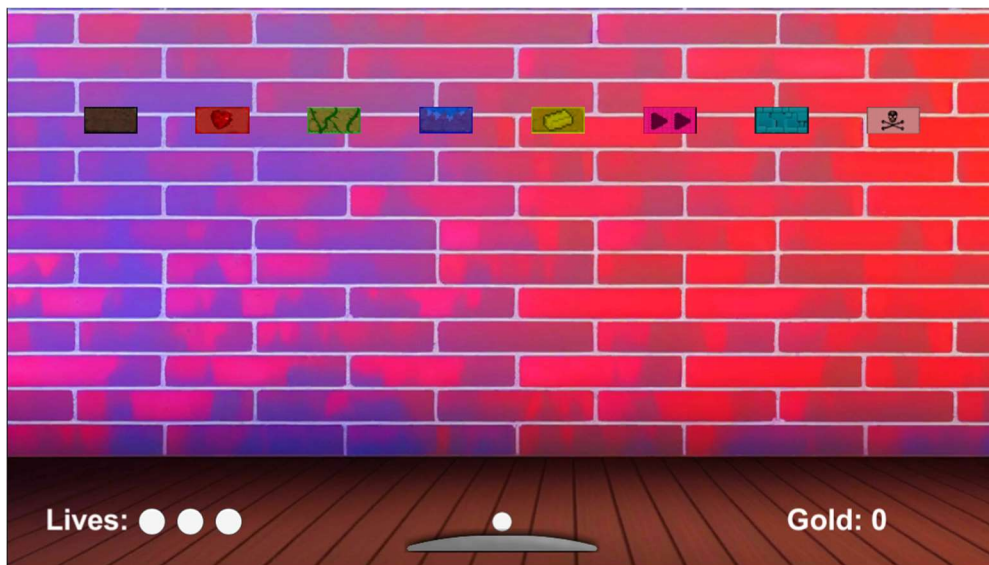
Za wyświetlanie poziomów odpowiedzialna jest klasa LevelGen. Służy do generowania paletki, piłki oraz bloków na scenie gry. Dostępne są dwa typy generowania poziomów.

Przy inicjowaniu klasy LevelGen wywoływana jest funkcja Spawner, która odpowiada za wygenerowanie paletki, piłki oraz wybór tła. Pobiera te dane z plików gry, na podstawie wyborów użytkownika, które dokonał on w oknie sklepu. Następnie sprawdzany jest rodzaj poziomu i w zależności od tego wybierany jest typ generowania bloków.

Pierwszy typ to generowanie na podstawie bitmapy. Każdy z początkowych czternastu poziomów posiada unikalną bitmapę. Rysunek 17 przedstawia bitmapę poziomu zerowego (Rys. 18). Każdy kolor pixela odpowiada innemu blokowi w grze, natomiast szary kolor jest



Rys. 17 Bitmapa służąca do generowania poziomu zerowego



Rys. 18 Wygląd poziomu zerowego

kanalem alfa równym zero, co oznacza pełną przeźroczystość. Tam gdzie nie występuje żaden kolor, jest pusta przestrzeń w grze. Takie rozwiązanie pozwala na bardzo szybkie tworzenie poziomów z wykorzystaniem dowolnego edytora graficznego.

Drugim typem generowania poziomu jest generowanie losowe. Wykorzystuje on wbudowaną w Unity funkcję Random z parametrami pozwalającymi generować losowe liczby z zakresu od 0 do 100. Każdy blok zajmuje pewien zakres liczbowy przedstawiony poniżej:

- Brak bloku – 0-69
- Cyjan – 70-79
- Czarny – 80- 84
- Zielony – 85-86
- Niebieski – 87-89
- Fuksja – 90-92
- Żółty – 93-97
- Białły – 98-100

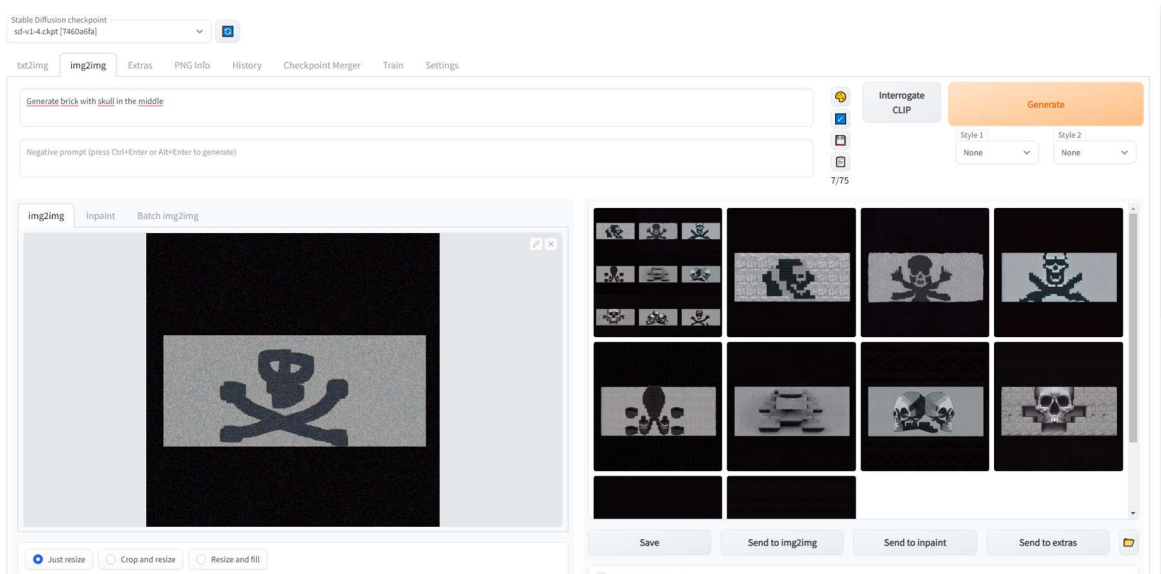
Dla każdego pola generowany jest losowy numer i na jego podstawie generowany odpowiedni blok. Po wygenerowaniu wszystkich bloków, jest generowany blok serca w pustym miejscu na planszy. Czerwony blok jest generowany osobno aby mieć pewność iż każdy poziom posiada warunek konieczny do jego ukończenia.

5.4. Tworzenie grafiki i efektów dźwiękowych

5.4.1. Generowanie elementów graficznych

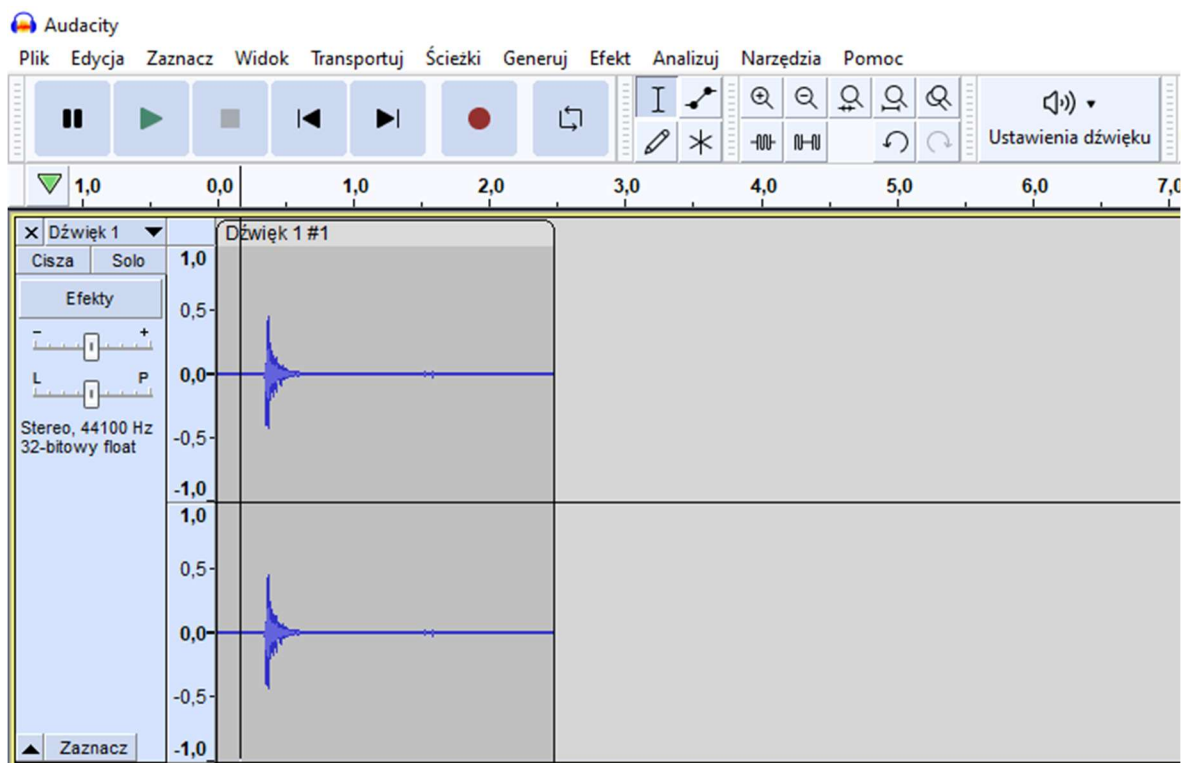
Do generowania większości elementów graficznych w aplikacji został wykorzystany darmowy program wykorzystujący model sztucznej inteligencji o nazwie Stable Diffusion. Głównie używany został tryb generowania za pomocą przykładowego obrazka wraz z tekstem opisującym co miało się znajdować na docelowej grafice (Rys.19). Obraz wejściowy został przygotowany tak aby jego rozmiar wynosił 512x512 pixeli, gdyż w tym rozmiarze był plik generowany przez program. Dla lepszego działania na wejściowy element został nałożony również szum aby znacznie poprawić rezultaty gdyż program do generacji obrazu bazuje w pewnym stopniu na usuwaniu szumów.

Po wygenerowaniu pierwszej partii obrazów należało wybrać najlepsze elementy z każdego z nich a następnie w programie graficznym, w tym przypadku Gimp, złączyć je w jeden obraz i następnie wgrać ponownie do programu generacji obrazów. Tą procedurę trzeba było powtarzać do otrzymania satysfakcjonującego efektu.



Rys. 19 Wygląd narzędzia Stable Diffusion podczas generowania szarego bloku

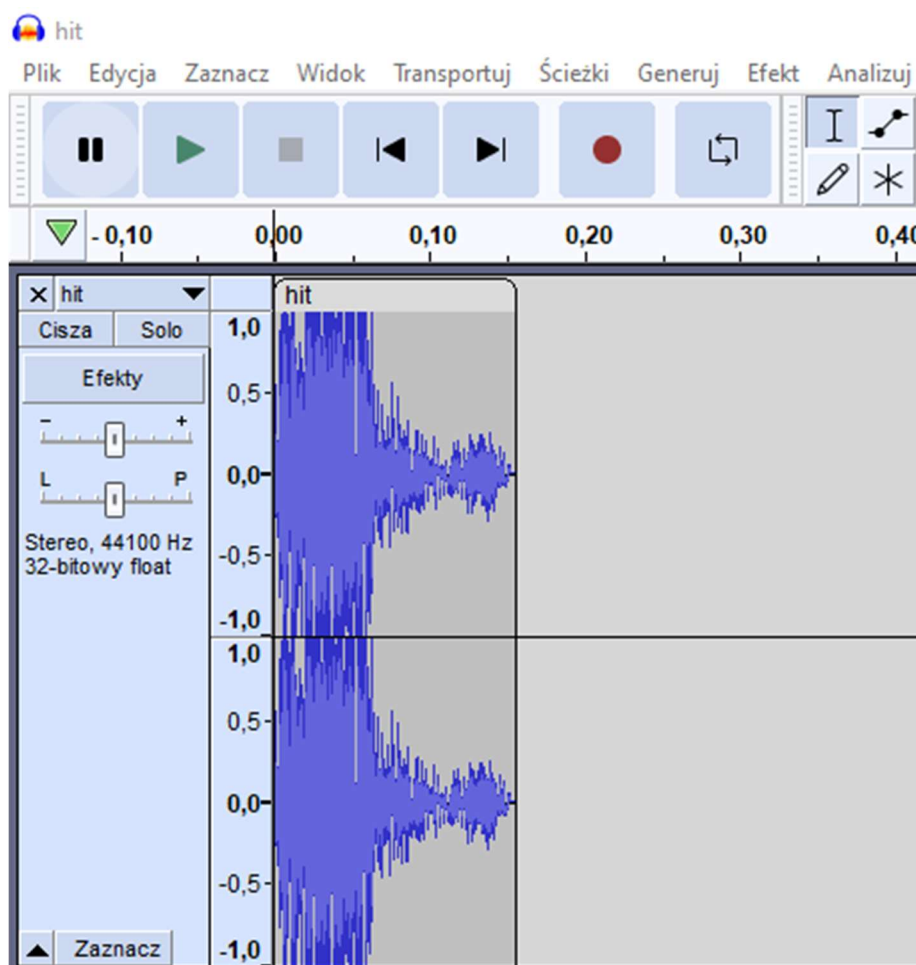
5.4.2. Tworzenie efektów dźwiękowych



Rys. 20 Nagrany dźwięk bez obróbki w Audacity

Do stworzenia efektów dźwiękowych posłużył darmowy program Audacity. Za jego pomocą został nagrany dźwięk uderzenia w biurko (Rys. 20), który bezpośrednio nie nadawał się do wykorzystania w grze.

Z wcześniej nagrałego dźwięku należało wyciąć elementy ciszy, gdyż powodowała by ona niepotrzebne opóźnienie dźwięku podczas odtwarzania. Następnie została podniesiona głośność dźwięku, gdyż był on zbyt cichy. Plik dźwiękowy nie był jeszcze jednak gotowy do wykorzystania w grze, ponieważ nie przypominał zbyt dźwięku który wydawała by piłka uderzająca w blok. Został podniesiony bas utworu oraz obniżony sopran, co spowodowało efekt „głuchego uderzenia” bardziej przypominającego uderzenie deską o ścianę. Na sam koniec wykorzystano narzędzie do usuwania szumów aby dźwięk był bardziej „czysty”. Wynikowy plik dźwiękowy (Rys.21) był gotowy do wykorzystania w aplikacji.



Rys. 21 Dźwięk uderzenia po przekształceniach

Rozdział 6

Testy i dalszy rozwój

Niniejszy rozdział zawiera opis testowania aplikacji na urządzeniach mobilnych. Znajduje się tutaj również plan na dalszy rozwój oprogramowania, oraz pomysły na wprowadzenie go na rynek.

6.1. Testowanie

Gra została przetestowana na telefonie Galaxy s20 marki Samsung z systemem operacyjnym Android w wersji 13. Przeprowadzono testy manualne polegające na korzystaniu z oprogramowania przez kilka godzin i przejściu każdego z poziomów kilka razy. Do testów poproszono również kilka osób z zewnątrz aby zagrały w grę i próbowały znaleźć niepożądane zachowania aplikacji.

Początkowo aplikacja wykorzystywała żyroskop do sterowania, ale w fazie testów, okazało się że sterowanie działa tylko, jeśli gracz jest skierowany w kierunku północnym. Zostało to naprawione, poprzez użycie akcelerometru, który działa niezależnie od kierunku w którym obrócony jest gracz.

Podczas testowania zostało znalezionych kilka błędów z rozmiarem interfejsu graficznego oraz z wyświetlaniem złota oraz zakupionych i aktywnych przedmiotów, oraz problem z resetowaniem postępu gracza.

Wszystkie znalezione błędy zostały naprawione w końcowej wersji aplikacji. Testy były prowadzone do momentu w którym nie można było znaleźć żadnych innych niepoprawnych zachowań.

6.2. Dalszy rozwój

Gra w obecnym stanie pozostawia wiele możliwych ścieżek rozwoju, w którym można dodawać nowe funkcjonalności.

Podstawowym elementem, jeżeli gra miała by trafić do dystrybucji, jest implementacja jakiegoś modelu biznesowego, pozwalającego na monetyzację aplikacji. Najprostszym i bardzo popularnym rozwiązaniem jest implementacja reklam, oraz możliwość zakupu wersji która jest ich pozbawiona. Można również wprowadzić elementy w sklepie za walutę premium, która była by możliwa do zdobycia tylko za prawdziwe pieniądze.

Kolejnym elementem rozwoju jest dodanie większej ilości zawartości grywalnej, czyli zapewnienie graczom więcej możliwości spędzenia czasu w grze. Jednym z takich rozwiązań jest dodanie znacznie większej ilości poziomów. Można również zaimplementować poziomy rotacyjne, czyli takie które zmieniają się co jakiś okres czasu i z których można dostać jakieś nagrody w postaci ekskluzywnych przedmiotów.

Aplikacja pozwala również na zaimplementowanie elementów rozgrywki sieciowej. Gracze mogli by tworzyć poziomy a następnie wyzywać innych graczy aby zmierzyli się z nimi. Można również zaimplementować ranking definiujący, który gracz pokonał najwięcej poziomów, lub pokonał dany poziom najszybciej.

Możliwe jest również dodanie systemu osiągnięć, zintegrowanego z Google Play, aby użytkownicy mieli możliwość śledzenia ilości zniszczonych bloków lub innych ciekawych wyzwań.

Rozdział 7

Podsumowanie

Cel pracy dyplomowej został osiągnięty. Stworzono w pełni funkcjonalną, grę mobilną w środowisku Unity. Gra zawiera wszystkie cechy uwzględnione w wymaganiach funkcjonalnych jak i нефункциональных, przez co można uznać ją za zakończony projekt z możliwością dalszego rozwoju opisanym w rozdziale szóstym.

Tworzenie gier komputerowych jak i nauka związana z nowymi technologiami gier komputerowych jest bardzo przyszłościowym zajęciem. Jak pokazuje historia to rynek ten rozwija się bardzo szybko, co powoduje znaczny wzrost popytu na tego rodzaju oprogramowanie na rynku.

Bibliografia

- [1] C. Crawford, „The Art of Computer Game Design”, Wyd. McGraw-Hill/Osborne Media, 1984
- [2] „Pierwszy komputer na świecie - kiedy powstał i gdzie? - NANO”, 9 wrzesień 2022. <https://nano.komputronik.pl/n/pierwszy-komputer-na-swiecie-kiedy-powstal/> (dostęp 9 listopad 2022).
- [3] „A history of the computer game”. <https://www.jesperjuul.net/thesis/2-historyofthecomputergame.html> (dostęp 9 listopad 2022).
- [4] M. J. P. Wolf, *The Video Game Explosion: A History from PONG to Playstation and Beyond*. ABC-CLIO, 2008.
- [5] „CGW numer 1”, *Computer Gaming World*, październik 1982. <https://www.cgwmuseum.org/galleries/index.php?year=1981&pub=2&id=1> (dostęp 14 listopad 2022).
- [6] D. Bricklin, „IBM PC Announcement 1981”, 1999. <http://www.bricklin.com/ibmpcannouncement1981.htm> (dostęp 14 listopad 2022).
- [7] *Byte Magazine Volume 06 Number 12 - Computer Games*. 1981. Dostęp: 14 listopad 2022. [Online]. Dostępne na: <http://archive.org/details/byte-magazine-1981-12>
- [8] „CGW numer 61”, *Computer Gaming World*, lipiec 1989. <https://www.cgwmuseum.org/galleries/index.php?year=1989&pub=2&id=61> (dostęp 14 listopad 2022).
- [9] „The History of Development and Boom of Mobile Game in the Gaming industry”, *Buy app reviews from real users*. <https://proreviewsapp.com/the-history-of-development-and-boom-of-mobile-game> (dostęp 9 listopad 2022).
- [10] „Mobile_Games.pdf”. Dostęp: 16 listopad 2022. [Online]. Dostępne na: https://homepages.tuni.fi/frans.mayra/Mobile_Games.pdf
- [11] „What arcade games looked like before video games, 1968 - Rare Historical Photos”, <https://rarehistoricalphotos.com/>, 20 sierpień 2019. <https://rarehistoricalphotos.com/arcade-games-history-1968/> (dostęp 20 listopad 2022).
- [12] S. Ozay, „A Brief History of Brick Breaker Video Games”, *Hero Concept*, 9 wrzesień 2018. <https://www.heroconcept.com/a-brief-history-of-brick-breaker-video-games/> (dostęp 20 listopad 2022).
- [13] „Co to jest akcelerometr i jak działa”, 30 listopad 2019. <https://www.benchmark.pl/aktualnosci/co-to-jest-akcelerometr-i-jak-dziala.html> (dostęp 9 styczeń 2023).

Spis rysunków

Rys. 1 Pierwsza gra komputerowa Spacewar! [3].....	9
Rys. 2 Gra "Pong" [3].....	10
Rys. 3 Okładka pierwszego wydania czasopisma "Computer Gaming World" [5].....	10
Rys. 4 Gra Tetris na telefonie Hagenuk MT-200.....	11
Rys. 5 Gra Breakout [12]	13
Rys. 6 Zrzut z gry Brick Breaker Star.....	14
Rys. 7 Zrzut ekranu z gry BBTAN	15
Rys. 8 Przebieg interakcji.....	19
Rys. 9 Widok menu głównego	21
Rys. 10 Panel sklepu	22
Rys. 11 Widok kupionego oraz wybranego elementu.....	22
Rys. 12 Wygląd ekranu "Zasady"	23
Rys. 13 Okno przedstawiające wybór poziomów	24
Rys. 14 Okno przedstawiające pierwszy poziom.....	24
Rys. 15 Wygląd ekranu wygranej	25
Rys. 16 Widok ekranu porażki	25
Rys. 17 Wygląd poziomu zerowego	31
Rys. 18 Bitmapa służąca do generowania poziomu zerowego	31
Rys. 19 Wygląd narzędzia Stable Diffusion podczas generowania szarego bloku.....	33
Rys. 20 Nagrany dźwięk bez obróbki w Audacity.....	33
Rys. 21 Dźwięk uderzenia po przekształceniach	34

Dodatki

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie, do pracy dołączono dodatkowe pliki zawierające:

- projekt źródłowy dla środowiska Unity,
- dokumentacja kodu,
- zbudowany plik wynikowy na system Android
- krótki film demonstracyjny