Uniwersytet Wrocławski

Wydział Fizyki i Astronomii

*Kamil Dłutowski*

**Interaktywny quiz jako forma unowocześnienia dydaktyki**

Praca inżynierska na kierunku

Informatyka Stosowana i Systemy Pomiarowe

Opiekun

dr hab. Zbigniew Koza

Wrocław 2020

Spis treści

Streszczenie 3

1.Wstęp 4

1.1 Cel i zakres pracy 4

2. Technologie i narzędzia 5

2.1 Unity 5

2.1.1 ScriptableObject 8

2.2 Photon 9

2.3 Unity Quick-sheet 9

2.4 Firebase 9

2.5 RestClient 9

2.6 Repozytorium projektu 9

3.Wygląd i funkcje aplikacji 10

3.1 Ekran początkowy 10

3.2 Tryby gry 10

3.2.1 Gra Prawda- Fałsz 11

3.2.2 Gra z czterema odpowiedziami 12

3.3 Gra sieciowa i jednoosobowa 13

3.3.1 Lobby 14

3.4 Ekran podsumowania 15

3.5 Lista pytań w arkuszu „Google” 15

3.6 Tworzenie nowych pytań 16

4.Wnioski 17

4.1 Problemy i trudności podczas pisania aplikacji 17

4.1.1 Technologie i problemy z oprogramowaniem 17

4.1.2 Problem z wyświetlaniem pytań z bazy danych 17

4.3 Rzeczy warte poprawienia 18

4.4 Podsumowanie 18

5.Bibliografia 19

# Streszczenie

Celem niniejszej pracy inżynierskiej jest stworzenie programu ułatwiającego sprawdzanie wiedzy studentów podczas wykładu.

Aplikacja umożliwia rozwiązywanie quizu zarówno w formie indywidualnej, jak i w trybie multiplayer. Quiz można uruchomić na ekranie komputera (Windows, MacOS oraz Linux) oraz na smartfonie z systemem operacyjnym IOS lub Android. Do dyspozycji graczy dostępne są dwie formy odpowiedzi na pytania. W pierwszej gracz stwierdza prawidłowość wyświetlanego twierdzenia, w drugiej zaś każde pytanie posiada 4 losowo ułożone odpowiedzi ,z których zawsze jedna odpowiedź jest prawdziwa.

Interface użytkownika został wykonany przy pomocy silnika graficznego Unity z asystą narzędzia Microsoft Visual studio, w którym zostały napisane wszystkie skrypty języka C# użyte w grze. W celu ustanowienia połączenia między graczami w trybie gry sieciowej posłużyło rozszerzenie do silnika Unity – „Photon”, które w wersji darmowej umożliwia obsługę maksymalnie do 20 jednoczesnych graczy. Do edycji pliku pytań i odpowiedzi został wykorzystany dodatek „Unity Quick-sheet” który umożliwia przesłanie danych z pliku umieszczonego w chmurze do pliku ScribitableObject w edytorze Unity. Dodatkowo dzięki programowi Xcode przetestowane było uruchamianie aplikacji na smartfonie z systemem IOS. Baza danych została stworzona przy wykorzystaniu platformy „Firebase” dostarczonej przez firmę Google.

# 1.Wstęp

W dobie Internetu wykładowcom może być coraz trudniej utrzymać uwagę studentów. Rozproszeni przez telefony komórkowe i komputery, studenci, a przynajmniej ich znaczna część, nie są w stanie koncentrować swojej uwagi na treści trwającego kilkadziesiąt minut wykładu i gdy stracą wątek, spędzają resztę czasu, przeglądając portale społecznościowe lub grając w gry. W ten sposób powstał pomysł stworzenia aplikacji, która nie tylko pozwoli prowadzącemu na sprawdzenie anonimowo poziomu wiedzy studentów, ale która również pokaże, w formie zabawy i rywalizacji, że przyswajanie nowej wiedzy wcale nie jest męczące i może być dobrą zabawą.

W niniejszej pracy przedstawię implementację interaktywnego quizu oraz omówię pokrótce technologie oraz funkcjonalności, jakie wykorzystałem przy tworzeniu programu.

## 1.1 Cel i zakres pracy

Celem niniejszej pracy inżynierskiej jest stworzenie aplikacji komputerowej oraz mobilnej umożliwiającej określenie poziomu znajomości danego tematu poprzez uczniów podczas prowadzonego wykładu. Zestawy pytań i odpowiedzi mogą zostać wgrane przez prowadzącego za pomocą arkusza kalkulacyjnego umieszczonego w chmurze. W grze występują dwa rodzaje pytań, na które gracz może udzielić odpowiedzi, są to pytania typu prawda-fałsz oraz pytania z 4 możliwymi odpowiedziami, z których zawsze jedna jest poprawna. Aplikacja obsługuje również dwa tryby gry: jednoosobowy oraz wieloosobowy. W trybie gry sieciowej każdy z uczestników może odpowiedzieć na zadane pytanie bezpośrednio z aplikacji na swoim telefonie a na koniec rozgrywki na na ekranie podsumowującym quiz zobaczyć wyniki swoje oraz pozostałych graczy. Gracz podłączając się do pokoju gry może wybrać, czy chce pozostać anonimowy lub podać nazwę, jaka będzie używana przez niego w grze. Tryb jednoosobowy posiada identyczny system gry jak tryb wieloosobowy.

# 

# 2. Technologie i narzędzia

## 2.1 Unity

Unity 3D[1] jest zintegrowane środowisko służące do tworzenia i edycji gier oraz programów komputerowych. Pozwala na tworzenie zarówno gier 2D, jak i 3D.

Środowisko to posiada własny silnik graficzny jak również silnik dźwiękowy oraz symulator fizyki.

Przy wykorzystaniu Unity możliwe jest stworzenie gier na urządzenia mobilne (Android/IOS), jak również aplikacji na tradycyjne komputery działające pod kontrolą systemu Windows, MacOS lub Linux.

Niewątpliwie jedną z największych zalet Unity3D jest to, że pomimo dużych możliwości i ogromnej liczby gotowych do wykorzystania funkcji, jest łatwy w obsłudze oraz posiada przyjemny dla użytkownika interfejs edytora.

Stworzenie prostej gry 3D nie wymaga kilkuletniego doświadczenia, wystarczy zrozumieć podstawy pracy z edytorem, znaczenie danych pól i na podstawie dostępnych instrukcji tworzonych przez samo Unity bądź innych poradników dostępnych w Internecie praktycznie każdy może zacząć swoją przygodę z tworzeniem gier.

Środowisko Unity nie wymaga również mocnych podzespołów komputerowych, ponieważ doskonale radzi sobie również na słabszym sprzęcie. W moim przypadku opisywana w tej pracy interaktywna gra została stworzona na komputerze o następującej specyfikacji:

* **MacBook Air** (13-inch, Early 2015)
* **Procesor** 1,6 GHz Dwurdzeniowy procesor Intel Core i5
* **Pamięć** 8 GB 1600 MHz DDR3
* **Grafika** Intel HD Graphics 6000 1536 MB

Jak można odczytać z przedstawionych wyżej parametrów komputer nie należy do najnowszych ani najszybszych laptopów dostępnych na rynku, lecz mimo to nie odczuwałem żądnych problemów podczas tworzenia gry w edytorze Unity oraz podczas pisania i testowania kodu w Microsoft Visual Studio[[1]](#footnote-1).

Łatwość i przystępność Unity nie znaczą, że jest to rozwiązanie tylko dla początkujących programistów. Środowisko to jest bardzo popularne i z powodzeniem wykorzystywane również przez duże studia zajmujące się masową produkcją gier. Między innymi Polskie studio CD Project Red przy wykorzystaniu wyżej wspomnianego środowiska stworzyło bardzo popularną grę "Gwint: Wiedźmińska gra karciana" [2]

Rysunek 1: Wygląd edytora Unity

Choć przy pierwszym uruchomieniu interfejs programu wydaje się dość skomplikowany i nieintuicyjny, to po kilku chwilach używania wszystko staje się w nim jasne, a sposób wykonania wszystkich niezbędnych akcji staje się prosty i dobrze przemyślany.

W samym centrum programu znajduje się kilka (dowolnie modyfikowanych) zakładek. Domyślnie są to:

1. *Scene*: inaczej edytor. To właśnie w tym miejscu tworzymy projekt, zmieniamy ustawienie obiektów oraz dodajemy nowe elementy, np. pola do wyświetlania tekstu, platformy w grze lub przycisk startu programu.
2. *Game*: w tym widoku możemy łatwo podejrzeć, jak nasz projekt będzie wyglądał po skompilowaniu i oddaniu do użycia. Innymi słowy jest to podgląd tego, jak robione przez nas zmiany wpływają na ogólny układ elementów gry. Co ważne w tym widoku nie są wyświetlane animacje, więc widzimy jedynie statyczny obraz obecnie edytowanej sceny gry.
3. *Assets* *store*: jest to biblioteka zawierająca zarówno darmowe, jak i płatne rozszerzenia i dodatki umożliwiające łatwe zwiększanie możliwości naszego projektu oraz znacząco zmieniające jego wygląd. Twórcy mogą znaleźć również inspirację wśród już gotowych przykładowych projektów i szkoleń mających na celu wprowadzenie nowych użytkowników w proces tworzenia na silniku Unty. Z *asset store* został pobrany między innymi „Photon”, dodatek umożliwiający utworzenie połączenia sieciowego między graczami.

Inne możliwe zakładki to m.in.: *project*, *inspector*, *animator*, *files*, *profiler* oraz wiele innych.

Wprawdzie silnik Unity jest bardzo rozbudowany pod względem możliwości tworzenia, edycji i modyfikacji elementów naszego projektu, lecz nie wszystko jest możliwe do zaimplementowania przy wykorzystaniu tylko i wyłącznie tego silnika. Do zaprojektowania i wykonania większości funkcji w grze należy dopisać własne metody w postaci skryptów w języku C#. Skrypty są tworzone w programie Visual Studio, z którym to Unity jest bardzo dobrze zintegrowany. Dodatkowo Unity3D, współpracując z firmą Microsoft, dostarcza dodatek dla Visual Studio pozwalający na debugowanie kodu w trakcie gry, a także połączenie z serwerem kontroli wersji.

Edytor Unty posiada własny tryb umożliwiający rozgrywkę sieciową, lecz ze względu na brakujące funkcje potrzebne do wykonania projektu zdecydowałem się na wykorzystanie osobnego dodatku („Photon”) umożliwiającego podłączenie wielu graczy do Quizu.

Tworząc aplikacje w Unity, można wybrać jedną z 3 metod połączeń graczy w trybie multiplayer[3]. Są to:

Server autorytatywny - wszystkie podłączone komputery graczy przesyłają informacje o tym, że gracz przesuwa się naprzód i to serwer oblicza kto, gdzie ma się przesunąć i zawraca informację do graczy jaki jest aktualny stan świata gry. Charakteryzuje się pewnymi opóźnieniami i nie jest najlepsza do gier zręcznościowych.

Serwer nieautorytatywny - cały świat gry trzymamy na naszym komputerze a do serwera przesyłamy informację o tym, że się przemieściliśmy. Serwer rozsyła pozostałym komputerom informację jaka jest nasza nowa pozycja, po czym na komputerach pozostałych graczy nasza pozycja jest aktualizowana. Informacje przesyłane są z niewielkim opóźnieniem

Połączenie Peer to peer - każdy z graczy łączy się bezpośrednio z komputerem innego gracza.

Taki rodzaj gry multiplayer nie wymaga zastosowania serwera - najczęściej połączenie tego rodzaju nawiązywanie jest za pomocą technologii bluetooth. Właśsnie ten rodzaj połączenia został wykorzystany w moim projekcie.

### 2.1.1 ScriptableObject

ScriptableObjects[4] to pojemniki na dane wykorzystywane w Silniku Unity. Niewątpliwie ich największą zaletą jest możliwość zapisania plików jako tzw. *assets* wewnątrz projektu. Dzięki temu nie muszą być one podpięte do obiektu gry. Dodatkowo mieszczą one bardzo duże ilości danych bez względu na ich typ. Ze względu na swoje właściwości ScriptableObjects nie mogą być edytowane z poziomu aplikacji. Każdorazowa zmiana danych wewnątrz pliku zmusza do ponownej kompilacji aplikacji w celu zastosowania poczynionych zmian. Niewątpliwie największe zastosowanie znajdują w procesie prototypowania, gdyż dzięki nim można bardzo łatwo stworzyć statyczną bazę danych aplikacji bez podłączania klasycznej bazy do aplikacji.

## 2.2 Photon

Podczas tworzenia opisywanej to aplikacji wykorzystałem jeden z dodatków dostępnych w *Unity Asstes Store* o nazwie „Photon[5]”. Jest to działający w czasie rzeczywistym *framework* umożliwiający połączenie do 20 użytkowników jednocześnie (w wersji darmowej). Przy wykorzystaniu opisywanego narzędzia możliwe jest stworzenie programu typu multiplayer na kilka popularnych platform jednocześnie, są to m.in.: Unity, Windows, MacOS, Android, czy Play Station.

## 2.3 Unity Quick-sheet

*Unity Quick-sheet*[6] jest to zewnętrzna biblioteka umożliwiająca dołączenie pliku arkusza kalkulacyjnego umieszczonego w chmurze Google Drive do projektu Unity. W szczególności umożliwia ona przypisanie danych pobranych z arkusza kalkulacyjnego do zawartego w projekcie pliku typu *ScriptableObjects*. Dużą zaletą tego rozszerzenia jest łatwość jego użycia we własnym projekcie. Po pobraniu paczki otrzymujemy w niej wszystkie potrzebne skrypty, więc jedyne co trzeba zrobić, to stworzyć plik arkusza na chmurze „Google Drive” i odpowiednio podłączyć Quick-sheet do projektu. Autor dodatku załącza obszerną instrukcję, jak należy takie połączenie stworzyć.

## 2.4 Firebase

*Firebase*[7] to oprogramowanie udostępniane przez firmę Google, które w swojej podstawowej, bezpłatnej wersji oferuje kilkanaście funkcjonalności pomagających w zarządzaniu aplikacją. Oprócz implementacji bazy danych narzędzie to pozwala również na sprawdzanie statystyk naszej aplikacji oraz jej moneteyzajcę. Niewątpliwie największą zaletą Firebase jest jej uniwersalność - istnieje możliwość zaimplementowania tej platformy na urządzenia z systemem Android lub IOS, w programach webowych i w programach korzystających z silnika Unity. W przypadku opisywanej tu aplikacji główną zaletą okazała się baza danych czasu rzeczywistego, która pozwala przechowywać i synchronizować dane w mikrosekundach.

## 2.5 RestClient

Zewnętrzna biblioteka *RestClient* [8] umożliwia rozszerzenie zakresu działania silnika graficznego Unity o obsługę REST API. Rozszerzenie to umożliwia korzystanie z podstawowych metod HTTP takich jak: GET, POST, PUT, DELETE, HEAD. Dzięki temu obsługa i przekazywanie danych bezpośrednio do bazy danych ulega znacznemu uproszczeniu.

## 2.6 Repozytorium projektu

Całość napisanego kodu wraz z opisami znajduje się na serwisie GitHub. Poniżej link do repozytorium.

<https://github.com/kamild160/Quiz.git>

# 3.Wygląd i funkcje aplikacji

## 3.1 Ekran początkowy

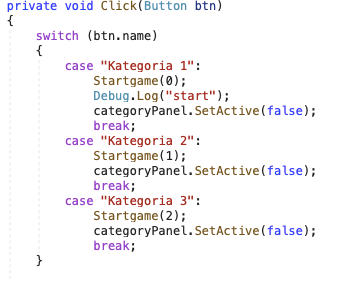
Wygląd aplikacji jest praktycznie identyczny zarówno w wersji komputerowej jak i w wersji mobilnej. Jedynym wyróżnikiem, jak zastosowałem w wersji na telefony, jest zmiana skalowania wyświetlanych elementów. Po otwarciu aplikacji gracz widzi ekran powitalny z numerem wersji programu. Jest to ważne, ponieważ gra w trybie sieciowym umożliwia podłączenie wyłącznie użytkowników korzystających z tej samej wersji aplikacji. Oprócz tego znajdują się tu jeszcze tytuł quizu oraz przycisk startu.

3.2 Tryby gry

Do wyboru graczy są dwa tryby gry. Tryb prawda-fałsz oraz tryb w którym każde pytanie posiada cztery warianty odpowiedzi z których zawsze jedna odpowiedź jest poprawna. Przed rozpoczęciem gry dla graczy wyświetlony zostaje ekran wyboru jednej z możliwych kategorii pytań.

### 3.2.1 Gra Prawda- Fałsz

Uruchamiając ten tryb, gracz zostanie powitany ekranem wyboru kategorii pytań. Po wybraniu jednej z trzech możliwych opcji ekran wyboru znika a użytkownik może rozpocząć odpowiadać na pytania. Funkcja wyboru kategorii opiera się na załadowaniu wybranego pliku z pytaniami po wciśnięciu przycisku przypisanego do danej grupy pytań.



Rysunek 2: Funkcja sprawdzająca wybór kategorii i ukrywająca ekran wyboru kategorii

Wybierając opcję „kategoria 2”, edytor wybiera plik z pytaniami przypisany do przycisku o tej właśnie nazwie i wyświetla je na ekranie gry. Ekran gry składa się z: pola z losowo wybranym pytaniem z listy, dwóch przycisków reprezentujące możliwe odpowiedzi - Prawda lub Fałsz, oraz przycisku powrotu na ekran wyboru trybu rozgrywki. Odpowiadając na zadane pytanie gracz klika odpowiednią według niego odpowiedź, jeżeli wybrana odpowiedź jest poprawna kliknięty przycisk zmieni kolor na zielony a suma punktów gracza zwiększy się o 1. W przypadku, gdy wybrana odpowiedź jest niepoprawna, zmieni ona kolor na czerwony a suma punktów gracza nie zmieni się. Po udzieleniu odpowiedzi na wszystkie możliwe w danej kategorii pytania gracz zostanie przeniesiony na ekran podsumowana.

### 3.2.2 Gra z czterema odpowiedziami

Podobnie jak w pierwszym trybie rozgrywki, również tutaj gracz rozpoczyna od wyboru kategorii pytań. Zasada wybierania kategorii i wyświetlania pytań opiera się na tych samych rozwiązaniach programistycznych jak w przypadku trybu prawda-fałsz. Po wybraniu kategorii ekran rozgrywki różni się jednak od opisywanego wcześniej. Posiadamy tu blok z pytaniem oraz 4 zawsze wyświetlane w losowej kolejności odpowiedzi, z których zawsze jedna jest poprawna. Gracz po kliknięciu na wybraną przez siebie odpowiedz widzi od razu, czy jego wybór jest poprawny, czy nie, ponieważ kolor wybranej odpowiedzi zmienia się w zależności od poprawnoci odpowiedzi. Jeżeli został kliknięty poprawny przycisk z odpowiedzą, zmieni on kolor na zielony a suma punktów gracza zwiększy się o 1, w przeciwnym wypadku przycisk

zmieni kolor na czerwony a suma punktów gracza nie zmieni się. Dodatkowo w każdym momencie gracz może wcisnąć przycisk powrotu do ekranu wyboru tryby w celu zmiany rodzaju gry lub kategorii.

## 3.3 Gra sieciowa i jednoosobowa

Jedną z kluczowych funkcji opisywanego tutaj programu jest możliwość przeprowadzenia rozgrywki sieciowej, w której może uczestniczyć jednocześnie do 20 graczy. W każdym przypadku gracz lub gracze mają do wyboru te same rodzaje starć do rozegrania: tryb prawda - fałsz lub tryb z czterema odpowiedziami z którego zawsze jedna odpowiedź jest prawdziwa. Przejście pomiędzy rozgrywką sieciową a jednoosobową wykonywane jest za pomocą przycisków na ekranie wyboru trybu rozgrywki.

W przypadku wyboru gry jednoosobowej (kliknięcie obrazka po lewej stronie ekranu) gracz zostanie przeniesiony na widok wyboru gry. Kliknięcie drugiego obrazka przeniesie gracza na stronę „lobby”.

### 3.3.1 Lobby

W lobby gracz jest w stanie definiować, jaka nazwa zostanie mu przypisana w trakcie rozgrywki. Istnieją dwie opcje deklaracji nazwy użytkownika

1. Gracz sam definiuje swoją nazwę

2. Gracz wykorzystuje przycisk „Graj anonimowo”, który przypisuje mu nazwę składającą się z 8 losowo wybranych liter.

Kolejnym ekranem, jaki zobaczą gracze ,jest ekran dołączania do rozgrywki - na środku ekranu wyświetlony zostaje napis „Czekam” który będzie wyświetlał się tak długo aż wszyscy zadeklarują swoją nazwę użytkownika i nacisną przycisk dołącz.

Osoba, która pierwsza dołączyła do pokoju gry, jest jednocześnie osobą która tworzy dany pokój i mistrzem gry. Mistrz gry decyduje, który rodzaj gry zostanie wybrany oraz podejmuje decyzję o zakończeniu trwającej rozgrywki.

## 3.4 Ekran podsumowania

Ekran podsumowania zawiera informacje o liczbie punktów zdobytej przez gracza oraz listę dziesięciu najlepszych wyników zebranych przez graczy. Z tego ekranu użytkownik może bezpośrednio powrócić do ekranu wybory gry.

## 3.5 Lista pytań w arkuszu „Google”

Lista pytań i odpowiedzi przechowywana jest na chmurze „Google Drive” w postaci dwóch arkuszy kalkulacyjnych. Pierwszy zawiera zbiór pytań do trybu gry prawda-fałsz i składa się z kolumny z treścią pytania oraz odpowiedzi czy zadane pytanie jest prawdziwe, czy nie. Jeżeli pytanie jest prawdziwe jest to wartość 1, jeżeli zaś nieprawdziwe to jest to wartość 0. Kolejne strony arkusza odpowiadają kolejnym kategoriom pytań w quizie.

W przypadku drugiego pliku podobnie jak opisywany wyżej posiada kolumnę z pytaniami dodatkowe cztery kolumny na możliwe odpowiedzi oraz kolumnę zawierającą poprawną odpowiedzą na pytanie. Również tutaj kolejne strony arkusza odpowiadają kolejnym kategoriom pytań. Dzięki temu rozwiązaniu możliwe jest łatwe modyfikowanie istniejących już pytań bądź dodanie nowych. Zmiany w pliku może dokonać każda osoba posiadającą lik. W celu dodania pytania należy dodać jego treść w kolumnie „Questions” w kolejnych kolumnach podajemy możliwe odpowiedzi oraz deklarujemy która odpowiedź jest poprawna. Ostatnia kolumna pozwala zadeklarować czy dane pytanie jest prawdziwe lub fałszywe. Arkusz kalkulacyjny dostępny jest pod linkiem

<https://docs.google.com/spreadsheets/d/1ufzrbmeoQQU6kXqX34gcPv9k1MbRGodE8jLOwfVPNgc/edit?usp=sharing>

## 3.6 Tworzenie nowych pytań

Program posiada również dodatkowy ekran umożliwiający dodawanie nowych pytań do bazy danych. Znajduje się tu pole tekstowe do wpisania treści pytania, przycisk określający czy danie pytanie jest prawdziwe lub fałszywe oraz pola do wpisania 4 możliwych wariantów odpowiedzi wraz z określeniem która z podanych odpowiedzi jest poprawna.

Dane przekazywane są za pomocą REST API bezpośrednio do bazy danych Firebase. Niestety ze względu na problemy natury technicznej jakie napotkałem, aplikacja nie posiada możliwości użycia tych pytań w grze. Więcej na ten temat w rozdziale 4.2.

# 

# 4.Wnioski

## 4.1 Problemy i trudności podczas pisania aplikacji

### 4.1.1 Technologie i problemy z oprogramowaniem

Podczas pracy nad opisywanym projektem natrafiłem na kilka problemów z którymi musiałem się zmierzyć w celu jego ukończenia. Pierwszym z problemów było wybranie odpowiedniej metody połączenia gry w trybie multiplayer. Mimo że silnik graficzny Unity

posiada własne rozwiązania potrzebne w celu utworzenia połączenia między graczami, z zaczerpniętych opinii dowiedziałem się, że oferowane funkcję nie spełniają całkowicie oczekiwań potrzebnych do wykonania quizu. Ostatecznie wybór padł na dodatek „Photon”, lecz przed wykorzystaniem tego rozszerzenia chciałem umieścić grę na serwerze oraz podłączyć do całości bazy danych w celu usprawnienia i poprawy funkcjonalności całej aplikacji. Niestety ze względu na problemy natury technicznej jakie napotkałem podczas pisania i czas potrzeby na nich rozwiązanie, musiałem zrezygnować z tego rozwiązania na rzecz połączenia bezpośredniego między graczami oraz użycia obiektów ScriptableObject.

Jednym z problemów technicznych, o których wspomniałem wcześniej, były problemy z oprogramowaniem Unity. Po wykonaniu aktualizacji wersji oprogramowania niezbędnej w celu wykonania podłączenia danych do arkusza kalkulacyjnego edytor przestał odczytywać podpięte pliki skryptowe. Problem został naprawiony po wykonaniu kolejnej aktualizacji do najnowszej możliwej wersji silnika.

### 4.1.2 Problem z wyświetlaniem pytań z bazy danych

Jednym z największych problemów jakie napotkałem podczas pisania wspomnianej tu aplikacji jest przejście z użytych początkowo *ScriptableObject* na połączenie czysto bazodanowe, w którym treści pytań przesyłane są bezpośrednio z serwera Firebase. Ze względu na początkowe przygotowanie i utworzenie mechaniki gry na wspomnianych wcześniej *ScriptableObject* przejście na bazy danych okazało się znacznie trudniejsze niż wskazywały na to początkowe przygotowania. Głównym problemem był mechanizm sprawdzania, czy na wszystkie pytania w turze udzielono odpowiedzi i czy były to odpowiedzi poprawne. Wybrany przeze mnie sposób polega na stworzeniu dodatkowej zmiennej, do której przypisuję dane znajdujące się w pliku *ScriptableObject*, a następnie po udzieleniu odpowiedzi dane pytanie jest usuwane z listy aż do jej całkowitego opróżnienia. Podczas próby dostosowania tego elementu kodu do obsługi baz danych doświadczyłem wielu problemów zarówno z przypisywaniem danych do nowej zmiennej, jak i dużych problemów z dostosowywaniem przygotowanej wcześniej mechaniki sprawdzania poprawności odpowiedzi i zliczania punktów. Dodatkowym problemem okazała się obsługa edytora Unity podczas próby przejścia na połączenie bazodanowe. Większość utworzonych połączeń i tekstów do wyświetlenia przestała wyświetlać poprawne teksty bądź całkiem przestała wyświetlać przypisane do wyświetlenia dane. Ponadto różnice w funkcjonowaniu wersji quizu w trybie czterech odpowiedzi oraz w trybie prawda-fałsz sprawiły że postanowiłem zostawić tę funkcję jako niedokończoną. Pytania do quizu wciąż można dodawać na dwa sposoby: bezpośrednio w pliku *ScriptableObject* lub wgrać je za pomocą arkusza „Google”, jednak oba te rozwiązania wymagają ponownej kompilacji aplikacji po zmianie danych.

## 4.3 Rzeczy warte poprawienia

Niewątpliwe pierwszą rzeczą do poprawienia jest dostosowanie aplikacji do pobierania danych z bazy danych. Dzięki temu dodawanie i odczytywanie pytań stanie się znacznie łatwiejsze, ale też, da to większą możliwość zapisu postępów graczy jak chodź by możliwość sprawdzania poczynionych postępów poprzez porównywanie wyników ze wcześniej już uzupełnionych quizów. Jako drugą rzecz do zmiany wymieniłbym szatę graficzną aplikacji. Obecnie jest ona bardzo prosta i pozbawiona różnego rodzaj animacji, które wzbogaciłyby odczucia graczy o lepsze doznania audio-wizualne.

## 4.4 Podsumowanie

Tworząc aplikację, wykorzystałem silnik graficzny Unity wraz z edytorem kodu Microsoft Visual Studio. Połączenie tych dwóch aplikacji sprawia, że tworzenie projektów jest stosunkowo proste ze względu na bardzo dobrą współpracę tych dwóch programów.

Należy również wspomnieć o wygodzie korzystania z edytora Unity, który z początku może przytłaczać ilością funkcji, lecz z czasem staje nie bardzo intuicyjny i przyjemny w obsłudze.

Stworzona aplikacja pozwoli na znaczne usprawnienie podczas sprawdzania wiedzy studentów na wykładzie. Dzięki wykorzystaniu oprogramowania Unity quiz można bez większych problemów uruchomić na najpopularniejszych systemach operacyjnych czy mobilnych, co niewątpliwie ogranicza wykluczenie technologiczne niektórych osób.

Opcją wyróżniającą ten projekt jest niewątpliwie możliwość gry w trybie sieciowym w którym to każdy z uczestników niejako konkuruje z pozostałymi w celu osiągnięcia najwyższego wyniku. Dodatkowo opcja gry anonimowej sprawia, że studenci nie będą odczuwali takiego quizu jako kolokwium ale bardziej jak ciekawą grę, która pomoże sprawdzić ich umiejętności i wiedzę z danej dziedziny nauki.

# 5.Bibliografia

* 1. <https://docs.unity3d.com/Manual/index.html>
  2. https://pl.wikipedia.org/wiki/Gwint:\_Wiedźmińska\_gra\_karciana
  3. Arkadiusz Brzegowy - Tworzenie gier multiplayer dla programistów Unity. Poziom pierwszy.
  4. <https://docs.unity3d.com/Manual/class-ScriptableObject.html>
  5. <https://doc.photonengine.com/en-us/realtime/current/getting-started/realtime-intro>
  6. <https://kimsama.gitbooks.io/unity-quicksheet/content>
  7. <https://itiq.pl/marketing/czym-firebase-marketing-aplikacji-mobilnych-unity/>
  8. https://github.com/proyecto26/RestClient

1. Edytor kodu od firmy Microsoft umożliwiający pisanie m.in. w języku C# [↑](#footnote-ref-1)