**Instytut Informatyki  
Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Programowanie urządzeń mobilnych**

**Podręcznik użytkownika aplikacji:**

***Bouncy Bird***

**Wykonał:**

**Moskal Dominik**

**Zakielarz Michał**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2023**

Spis treści

[1. Opis programu 3](#_Toc157457701)

[2. Przegląd zastosowań aplikacji 3](#_Toc157457702)

[3. Wymagania sprzętowe 3](#_Toc157457703)

[4. Instalacja i konfiguracja 3](#_Toc157457704)

[5. Środowisko programu 3](#_Toc157457705)

[5.1. Dane wejściowe 4](#_Toc157457706)

[5.2. Komunikacja z użytkownikiem 4](#_Toc157457707)

[5.3. Wyniki dostarczane przez program 4](#_Toc157457708)

[5.4. Przykłady danych wejściowych i wyników programu 4](#_Toc157457709)

[6. Sytuacje niepoprawne 6](#_Toc157457710)

[6.1. Wykaz komunikatów diagnostycznych 6](#_Toc157457711)

[6.2. Możliwe błędy wykonania się programu 7](#_Toc157457712)

[7. Literatura 7](#_Toc157457713)

# Opis programu

Bouncy Bird to gra mobilna z gatunku side-scroller zaprogramowana na urządzenia   
z systemem Android. Aplikacja polega na wcieleniu się w postać ptaka i przemieszczanie się nim pomiędzy przeszkodami za pomocą dotyku ekranu smartphone. Gra pozwala na dostosowywanie pod użytkownika wielu komponentów programu takich jak: wygląd postaci, poziom trudności czy dźwięki i wibracje telefonu. Aplikacja zapewnia intuicyjne GUI co sprawia, że nowi użytkownicy nie mają problemu z odnalezieniem się w aplikacji i komfortowemu jej używania.

# Przegląd zastosowań aplikacji

Aplikacja zapewnia szereg zastosowań o charakterze rozrywkowym. Jej głównymi zastosowaniami są:

* Rozwijanie umiejętności koncentracji uwagi
* Trening refleksu
* Trening koordynacji wzrokowo-ruchowej
* Relaks i umilenie czasu użytkownikowi poprzez wciągający gameplay i intuicyjne GUI

# Wymagania sprzętowe

* System operacyjny Android 10 lub nowszy
* Wsparcie ekranu dotykowego
* Wbudowana opcja latarki
* 512 MB pamięci RAM-u lub więcej
* Obsługa odtwarzania dźwięku
* Obsługa wibracji
* Minimum 150 MB wolnej przestrzeni dyskowej

# Instalacja i konfiguracja

1. Wejść w link do repozytorium:

https://github.com/dominikmoskal/PUM\_PROjekt.git.

1. Pobrać plik .apk na komputer.
2. Przenieść plik .apk do dowolnego folderu na smartphonie.
3. Wejść w plik .apk z pozycji smartphona i postępować zgodnie z poleceniami instalatora.

# Środowisko programu

Gra Bouncy Bird jest napisana w języku C# przy użyciu silnika Unity, który jest popularnym wyborem dla twórców gier na różne platformy, w tym Android. Gra jest zoptymalizowana do działania na urządzeniach z systemem Android w wersji 10 lub nowszej, które mają ekran dotykowy. Sterowanie ptakiem w grze odbywa się poprzez interakcje użytkownika z ekranem dotykowym.

# 5.1. Dane wejściowe

Dane wejściowe do gry pochodzą z interakcji użytkownika z ekranem dotykowym. Głównym wejściem jest dotknięcie ekranu, które powoduje, że ptak w grze wykonuje skok. Ta interakcja jest niezbędna do nawigacji ptaka przez przeszkody w grze. Są to:

* Stan ekranu
* Pozycja gracza
* Pozycja przeszkody
* Stan gracza

# 5.2. Komunikacja z użytkownikiem

Komunikacja z użytkownikiem odbywa się głównie za pośrednictwem interfejsu graficznego gry. Na ekranie wyświetlane są różne dane liczbowe, takie jak aktualny wynik użytkownika. Po zakończeniu gry, na ekranie pojawia się komunikat informujący o przegranej, a użytkownik ma możliwość rozpoczęcia nowej gry.

# 5.3. Wyniki dostarczane przez program

Program dostarcza wyniki w postaci punktów zdobytych przez gracza. Punkty są przyznawane za każde udane przeloty ptaka przez przeszkody. Wynik jest wyświetlany na ekranie, co pozwala graczowi na bieżąco śledzić swoje postępy.

# 5.4. Przykłady danych wejściowych i wyników programu

Poniżej przedstawiono przykładowe wyniki działania programu po interakcji gracza z ekranem dotykowym:

* Pozycja gracza

Obraz zawierający zrzut ekranu, miejsce parkingowe/przestrzeń, projekt graficzny, Wszechświat

Opis wygenerowany automatycznieObraz zawierający miejsce parkingowe/przestrzeń, Przestrzeń kosmiczna, Wszechświat, Obiekt astronomiczny

Opis wygenerowany automatycznie

Rysunek 2

Rysunek 1

Rysunek 1 przedstawia obiekt przed interakcją gracza, natomiast rysunek 2 pokazuje zależność interakcji gracza z ekranem dotykowym a położeniem obiektu w scenie.

* Obraz zawierający miejsce parkingowe/przestrzeń, Przestrzeń kosmiczna, Wszechświat, Obiekt astronomiczny

  Opis wygenerowany automatycznieStan gracza

Rysunek 3

Obraz zawierający tekst, zrzut ekranu, projekt graficzny, Grafika

Opis wygenerowany automatycznie

Rysunek 4

Na rysunku 3 przedstawiono stan kontrolowanego obiektu na początku rozgrywki. Obiekt kontrolowany przez gracza jest aktywny i grywalny. Rysunek 4 przedstawia zmianę stanu kontrolowanego obiektu wywołaną przez zdarzenie kolizji z przeszkodą.

# Sytuacje niepoprawne

Gra jest zaprojektowana w taki sposób, że minimalizuje możliwość wystąpienia błędów, dlatego nie przewiduje sytuacji niepoprawnych. Oznacza to, że jest ona odporna na różne scenariusze, które mogłyby potencjalnie prowadzić do błędów lub awarii wywołanych   
z winy użytkownika.

# 6.1. Wykaz komunikatów diagnostycznych

Gra jest wolna od błędów, nie ma potrzeby wyświetlania komunikatów diagnostycznych. Wszystkie potencjalne problemy są rozwiązywane w taki sposób, aby nie wpływały na doświadczenia użytkownika.

# 6.2. Możliwe błędy wykonania się programu

Gra jest zaprojektowana tak, aby unikać błędów podczas wykonywania. Oznacza to, że wszystkie aspekty gry, od renderowania grafiki po przetwarzanie danych wejściowych, są zaprojektowane tak, aby minimalizować ryzyko wystąpienia błędów. Dzięki temu gra jest stabilna i niezawodna. Podczas testów nie wykryto żadnych potencjalnych błędów oraz problemów z działaniem aplikacji.

# 7. Literatura

* https://docs.unity.com
* https://stackoverflow.com