

**Politechnika Świętokrzyska w Kielcach Wydział  
Elektrotechniki, Automatyki i Informatyki Katedra  
Informatyki, Elektroniki i Elektrotechniki**

*studia stacjonarne Informatyka rok 2*

**Programowanie Obiektowe (Java) projekt**

**Grupa** 2ID12B

**Zespół** Bartłomiej Grzesik Paweł Gawżyński

**Data**

**Temat projektu** Tic-Tac-Toe Android Game



Politechnika Świętokrzyska  
Kielce University of Technology

# 1. Opis projektu

Celem naszego projektu było stworzenie gry Tic-Tac-Toe w formie aplikacji mobilnej na urządzenia z systemem Android. Gracz ma do wyboru dwa tryby gry. Pierwszy tryb: możliwość gry z drugim graczem na jednym urządzeniu. Drugi tryb: możliwość gry z komputerem AI, gdzie został użyty wydajny algorytm sztucznej inteligencji. Do stworzenia naszego projektu użyliśmy oprogramowania Android Studio. Dzięki odpowiednim bibliotekom Androida stworzyliśmy interfejsy graficzne składające się z ładowania aplikacji, menu oraz interfejsów rozgrywki. Dodatkowo dużym atutem jest również to że aplikacja została zbudowana i załączona w formacie .apk i można ją normalnie zainstalować/używać na smartfonach z systemem Android. Plik instalacyjny znajduje się w folderze projektu.

## ***Technologie:***

- Java / Android Studio
- XML
- JUnit 5
- Dokumentacja

## ***Biblioteki: android , androidx***

- androidx.appcompat.app.AppCompatActivity
- android.graphics.Color
- android.os.Bundle
- android.view.View
- android.view.Window
- android.widget.Button
- android.widget.TextView
- android.widget.Toast
- android.content.Intent
- android.os.Handler

### ***Funkcjonalność projektu:***

- gra pomiędzy dwoma graczami na jednym urządzeniu
- gra z komputerem AI (Wydajny algorytm sztucznej inteligencji, przeciwnik nie popełnia błędów i wykonuje ruchy niczym najlepszy silnik szachowy)
- szybki system rozgrywek bez wychodzenia do menu, automatycznie rozpoczyna kolejną rundę
- możliwy restart gry
- zliczanie zdobytych punktów obu graczy
- szybkie działanie aplikacji
- responsywność oraz działanie na każdym urządzeniu mobilnym z systemem Android
- proste animacje przycisków, aplikacja intuicyjna w użyciu
- gra została stworzona w języku angielskim (dla każdego użytkownika niezależnie od kraju pochodzenia)

## **2. Uruchamianie projektu**

### **I. Sposób**

- a.** Pobieramy plik projektu z platformy GitLab, a następnie wypakowujemy
- b.** Instalujemy program Android Studio  
([https://developer.android.com/studio?gclid=CjwKCAjw7vuUBhBUEiwAEdu2pIUvIt-MbxH6P71GWO90Uh\\_8IQ1r6F9YNYi9ilfEsepr7piRD68rBoCVqIQAvD\\_BwE&qclsrc=a.w.ds](https://developer.android.com/studio?gclid=CjwKCAjw7vuUBhBUEiwAEdu2pIUvIt-MbxH6P71GWO90Uh_8IQ1r6F9YNYi9ilfEsepr7piRD68rBoCVqIQAvD_BwE&qclsrc=a.w.ds))
- c.** Uruchamiamy program Android Studio
- d.** Na pasku górnym wybieramy opcję File->Open
- e.** Wskazujemy folder (wypakowanego folderu)
- f.** Jeżeli mamy pobrany emulator to klikamy na przycisk RUN, jeżeli nie to pobieramy emulator do Android Studio
- g.** Projekt został uruchomiony

## II. Sposób (preferowany)

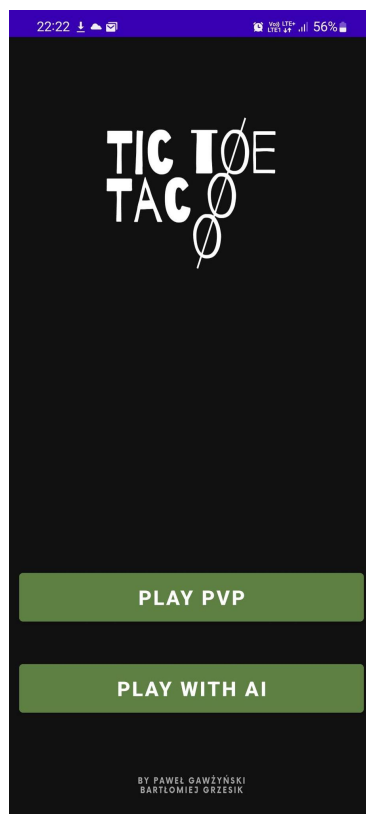
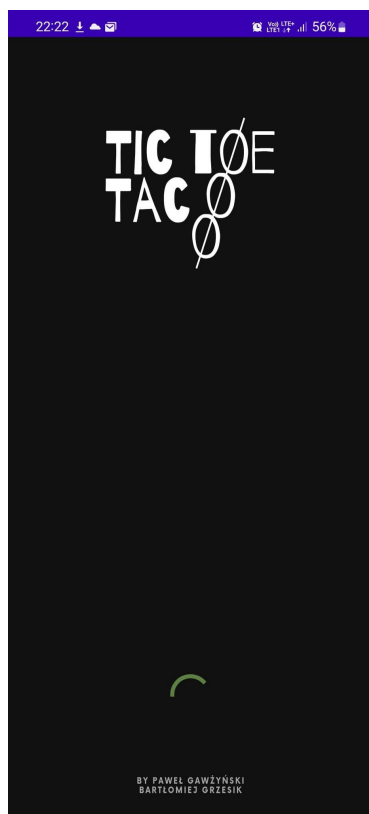
Projekt można również pobrać i zainstalować na swojego smartfona. Pobieramy TICTACTOE.apk z zip'a i przesyłamy na swoje urządzenie mobilne. Wchodzimy (na smartfonie) w Moje Pliki -> Downloads. Klikamy w plik, zatwierdzamy wszystko i instalujemy. Uruchamiamy aplikację i cieszymy się rozgrywką.

## 3. Obsługa projektu

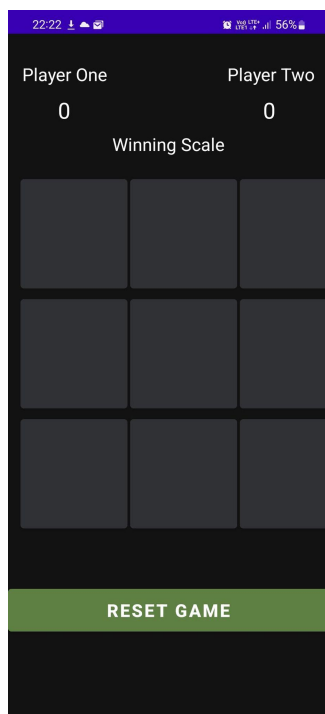
**Zasady Gry:** Graczy w tej grze jest dwóch, jeden ma kółka, drugi krzyżyki. Grają oni na zmianę ustawiając raz kółko, raz krzyżyk na wybranym polu. Jeśli jednemu z graczy uda się ułożyć trzy swoje znaki w jednym rzędzie (pionowym, poziomym lub po skosie) wówczas wygrywa.

*Czekamy aż gra się załaduje*

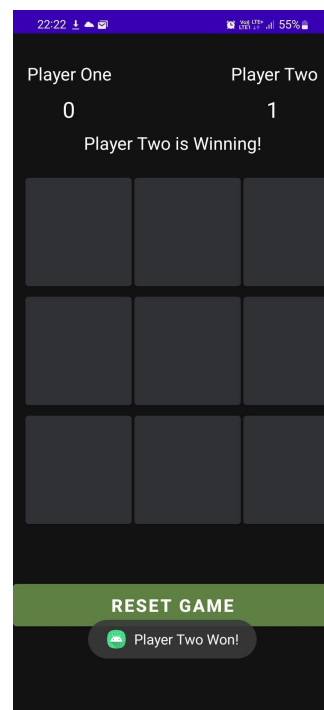
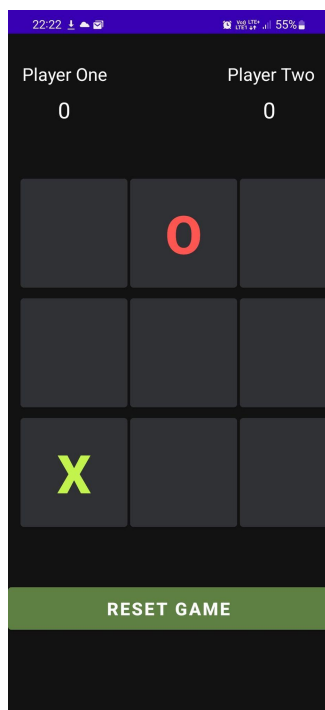
*W menu wybieramy tryb*



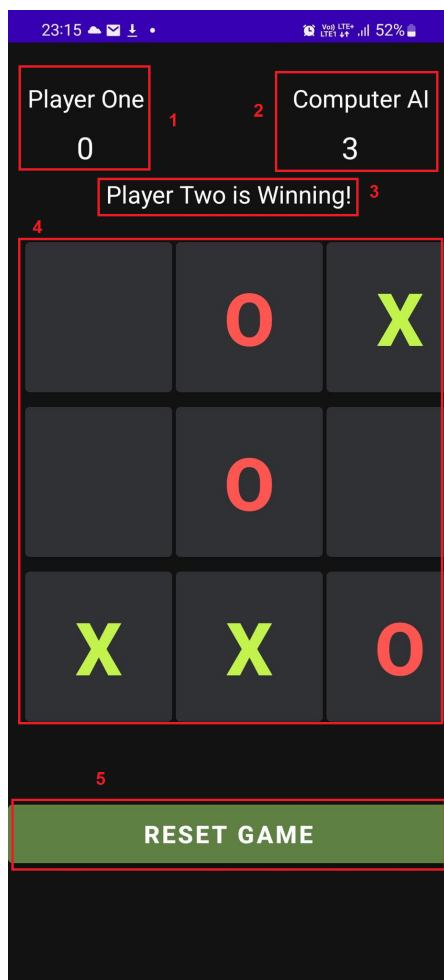
Po wybraniu trybu PvP  
gramy naprzemiennie  
(najpierw X potem O)



Analogicznie do zasad gry,  
przydziela punkt graczowi który  
wygrał rundę i resetuje plansze



*Grać można do woli, a rozgrywka jest bardzo szybka i przyjemna. Po kliknięciu przycisku RESET GAME, rozgrywka jest resetowana łącznie ze zdobytymi punktami. Podobnie jest w trybie PLAY WITH AI jednakże tam gramy przeciwko komputerowi. Algorytm AI przeciwnika jest bardzo wydajny i szybki. Interfejs gry jest bardzo intuicyjny, aby wyjść do menu wystarczy użyć przycisku lub gestu systemowego powrotu.*



### **Przewodnik po interfejsie gry:**

- 1 - Liczba punktów zdobytych przez Gracza Pierwszego
- 2 - Liczba punktów zdobytych przez Komputer AI lub w przypadku trybu PVP liczba punktów zdobytych przez Gracza Drugiego
- 3 - Pole informacji ( W tym miejscu wyświetlane są informacje na temat rozgrywki np. który użytkownik wygrał rundę)
- 4 - Pole rozgrywki składa się z 9 przycisków. Naprzemiennie stawiamy X i O wystarczy kliknąć
- 5 - Przycisk resetujący grę ( resetuje wszystko: punkty oraz pole rozgrywki)

## 4. Struktura Projektu

**PAKIET** `com.example.tic_tac_toe_app_java_project_ps` -  
Główny pakiet aplikacji

- **KLASA MainActivity** - Klasa Bazowa całej gry, opiera się na niej cały mechanizm rozgrywki dla trybu gry z komputerem, ma możliwości postawienia ruchu i reagowania na kliknięcie, sprawdza wygraną oraz liczy punkty zdobyte przez gracza i komputer. W klasie implementowana jest cała rozgrywka dla PLAY WITH AI (łączy wszystko) Klasa dziedziczy klasę AppCompatActivity oraz używa interfejsu z biblioteki Androida.

Pola:

**playerStatus** text informujący gracza dotyczący rozgrywki

**playerOneScore** text informujący gracza 1 o ilości punktów

**playerTwoScore** text informujący gracza 2 o ilości punktów  
(w tym przypadku jest to text ilości punktów zdobytych przez AI)

**buttons** przyciski rozgrywki

**playerOneScoreCount** zebrane punkty gracza 1

**platerTwoScoreCount** zebrane punkty gracza 2 ( w tym  
przypadku ilość punktów zdobytych przez AI)

**resetGame** przycisk resetowania gra

**roundCount** licznik rund

**activePlayer** aktywny gracz

**gameState** tablica z informacją czy w danym polu ma być O czy X  
początkowo jest 2 czyli pustka

**winningPositions** macierz z możliwymi opcjami wygranych

Metody:

**onCreate** - Tworzy planszę rozgrywki wykorzystuje elementu z biblioteki android wyświetla informacje takie jak nazwa gracza ilość punktów 9 przycisków oraz przycisk restartu.

**onClick** - Metoda rozgrywki wpisuje X i O w zależności od activePlayer za każdym razem sprawdza czy doszło do wygranej

jeżeli ktoś wygrał przypisuje danemu graczowi punkt sprawia że przycisk Reset Game jest użyteczny i resetuje rozgrywkę.

**checkWinner** - Metoda sprawdza na podstawie winning position czy doszło do wygranej. Zwraca true lub false w zależności od tego czy obecna kombinacja zgadza się z kombinacją z winningPosition.

**updatePlayerScore** - Zmienia wyświetlaną ilość punktów zmienia z wartości całkowitej na ciąg znaków i ustawia text

**playAgain** - Metoda umożliwia zagranie ponownie w grę resetuje punkty oraz czyści plansze

- **KLASA MainActPVP**- Klasa Bazowa całej gry, opiera się na niej cały mechanizm rozgrywki dla trybu gry pomiędzy graczami na jednym urządzeniu mobilnym, ma możliwości postawienia ruchu i reagowania na kliknięcie, sprawdza wygraną oraz liczy punkty zdobyte przez graczy. W klasie implementowana jest cała rozgrywka dla PLAY PVP(łączy wszystko) Klasa dziedziczy klasę AppCompatActivity oraz używa interfejsu z biblioteki Androida.

Pola:

**playerStatus** text informujący gracza dotyczący rozgrywki

**playerOneScore** text informujący gracza 1 o ilości punktów

**playerTwoScore** text informujący gracza 2 o ilości punktów

**buttons** przyciski rozgrywki

**playerOneScoreCount** zebrane punkty gracza 1

**playerTwoScoreCount** zebrane punkty gracza 2

**resetGame** przycisk resetowania gra

**roundCount** licznik rund

**activePlayer** aktywny gracz

**gameState** tablica z informacją czy w danym polu ma być O czy X początkowo jest 2 czyli pustka

**winningPositions** macierz z możliwymi opcjami wygranych

Metody:

**onCreate** - Tworzy planszę rozgrywki wykorzystuje elementu z biblioteki android wyświetla informacje takie jak nazwa gracza ilość punktów 9 przycisków oraz przycisk restartu.



**onClick** - Metoda rozgrywki wpisuje X i O w zależności od **activePlayer** za każdym razem sprawdza czy doszło do wygranej jeżeli ktoś wygrał przypisuje danemu graczowi punkt sprawia że przycisk Reset Game jest użyteczny i resetuje rozgrywkę.

**checkWinner** - Metoda sprawdza na podstawie winning position czy doszło do wygranej. Zwraca true lub false w zależności od tego czy obecna kombinacja zgadza się z kombinacją z winningPosition.

**updatePlayerScore** - Zmienia wyświetlaną ilość punktów zmienia z wartości całkowitej na ciąg znaków i ustawia text

**playAgain** - Metoda umożliwia zagranie ponownie w grę resetuje punkty oraz czyści plansze

- **KLASA Loader** - Klasa która implementuje Loading screen z logiem oraz twórcami

Metody:

**onCreate** - tworzy loading screen ustawia czas oraz tło

- **KLASA Navi** - Klasa zawierająca przyciski i wykonująca akcje nawigacji. Implementuje menu główne. Używa bibliotek allegro do wyświetlania przycisków oraz ich animacji

Pola:

**playPvP** przycisk służący do wyboru rozgrywki jeden na jeden

**playWAI** przycisk służący do wyboru rozgrywki z komputerem jako przeciwnikiem

Metody:

**onCreate** - implementuje menu główne tworzy warstwy do których są załączone przyciski. Ustawia listenera nasłuchujące klikanie przycisków.

## **PAKIET AI** - Pakiet przeciwnika AI

- **KLASA AIUtil** - Klasa implementująca przeciwnika (AI) do trybu PLAY WITH AI.
  - **KLASA Move** - Klasa przechowująca ruchy

Pola dla klasy Move:

**row** nr wiersza będący najlepszym wyborem

**col** nr kolumny będącej najlepszym wyborem

Pola dla klasy AIUtil:

**player** oznacza dla kogo szukamy najlepszego ruchu  
dla X czy O

**opponent** oznacza znak przeciwnika

**gameState** tablica reprezentująca aktualny stan gry

Metody dla klasy AiUtil:

**process** - Przetwarza oryginalna tablice na jej reprezentację a następnie wykonuje algorytm Obliczania najlepszego ruchu

**isMovesLeft** - sprawdza czy są dostępne ruchy,  
jeżeli istnieją ruchy zwraca prawdę, jeżeli nie ma żadnej  
możliwości ruchu zwraca fałsz

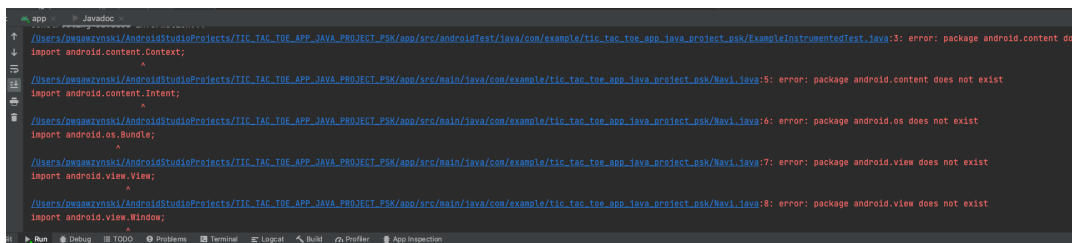
**evaluate** - Metoda odpowiadająca za obliczenie wag  
poszczególnych propozycji ruchów

**minmax** - Funkcja minmax znana z algorytmu minmax czyli  
algorytmu AI Computer\_Player. Zwraca najlepszy ruch

**findBestMove** - znajduje najlepszy ruch i zwraca go do głównego  
Drivera jako obiekt klasy move

## 5. Dokumentacja

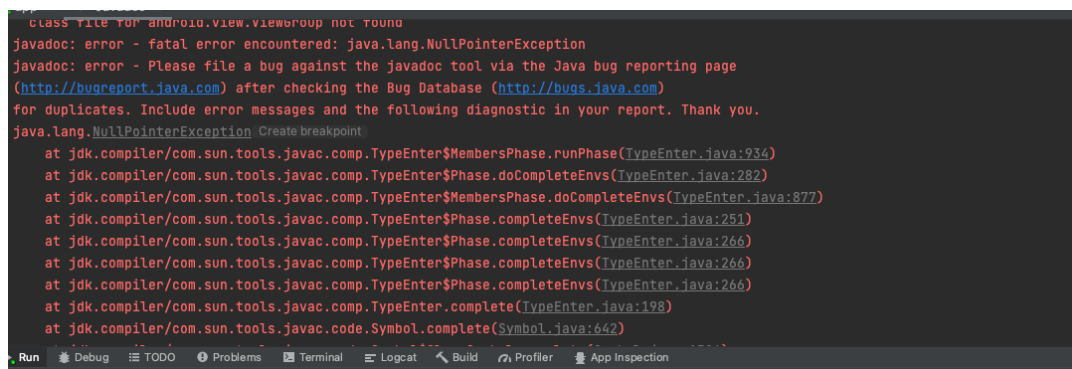
Niestety ale po 5 godzinnej próbie utworzenia dokumentacji javadoc w Android Studio wyskakiwały błędy, prawdopodobnie związane z niezgodnością wersji (screeny załączone poniżej). Ponowne reinstalacje zmiany wersji niestety nie przyniosły skutków. *Jednakże*, w ramach wypełnienia obowiązków utworzyliśmy dokumentację w programie Doxygen (opcja java). Dokumentacja wygląda i działa identycznie jak stworzona w javadoc. Całość została załączona w folderze projektu (zip'ie)



```

/Users/pawelzyski/AndroidStudioProjects/TIC_TAC_TOE_APP_JAVA_PROJECT_PSK/app/src/androidTest/java/com/example/tic_tac_toe_app_java_project_psk/ExampleInstrumentedTest.java:5: error: package android.content does not exist
import android.content.Context;
^
/Users/pawelzyski/AndroidStudioProjects/TIC_TAC_TOE_APP_JAVA_PROJECT_PSK/app/src/main/java/com/example/tic_tac_toe_app_java_project_psk/Nav1.java:5: error: package android.content does not exist
import android.content.Intent;
^
/Users/pawelzyski/AndroidStudioProjects/TIC_TAC_TOE_APP_JAVA_PROJECT_PSK/app/src/main/java/com/example/tic_tac_toe_app_java_project_psk/Nav1.java:6: error: package android.os does not exist
import android.os.Bundle;
^
/Users/pawelzyski/AndroidStudioProjects/TIC_TAC_TOE_APP_JAVA_PROJECT_PSK/app/src/main/java/com/example/tic_tac_toe_app_java_project_psk/Nav1.java:7: error: package android.view does not exist
import android.view.View;
^
/Users/pawelzyski/AndroidStudioProjects/TIC_TAC_TOE_APP_JAVA_PROJECT_PSK/app/src/main/java/com/example/tic_tac_toe_app_java_project_psk/Nav1.java:8: error: package android.view does not exist
import android.view.Window;
^

```



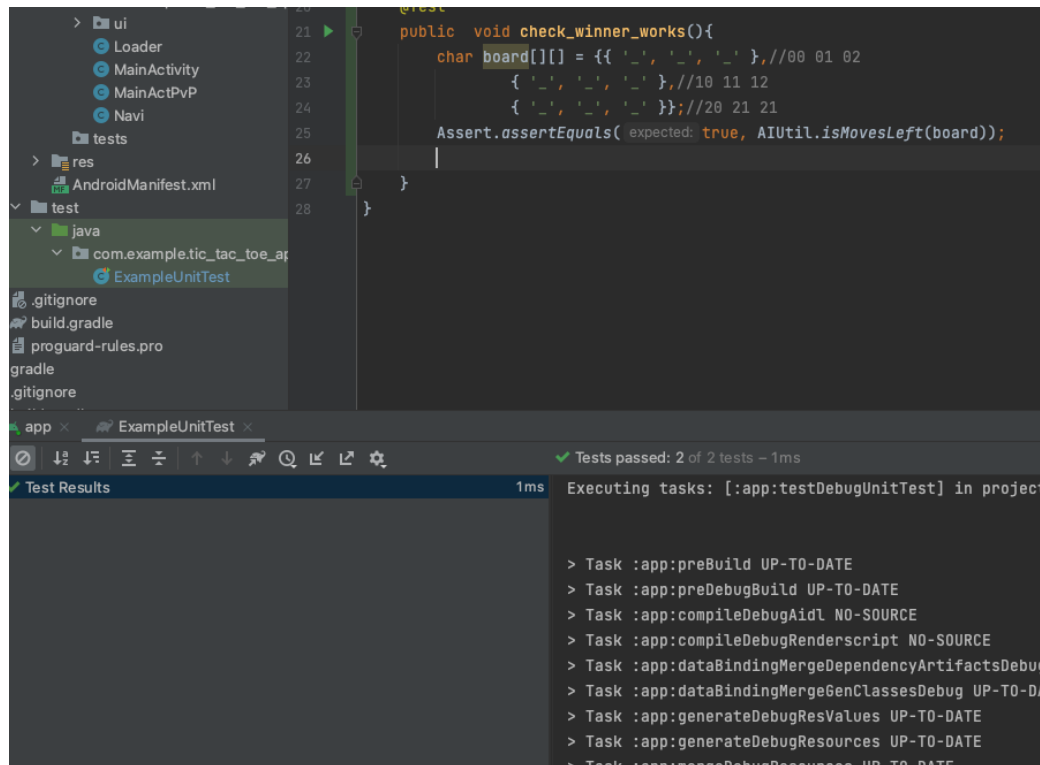
```

class file for 'android.view.ViewGroup' not found
javadoc: error - fatal error encountered: java.lang.NullPointerException
javadoc: error - Please file a bug against the javadoc tool via the Java bug reporting page
(http://bugreport.java.com) after checking the Bug Database (http://bugs.java.com)
for duplicates. Include error messages and the following diagnostic in your report. Thank you.
java.lang.NullPointerException Create breakpoint
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$MembersPhase.runPhase(TypeEnter.java:934)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$Phase.doCompleteEnvs(TypeEnter.java:282)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$MembersPhase.doCompleteEnvs(TypeEnter.java:877)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$Phase.completeEnvs(TypeEnter.java:251)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$Phase.completeEnvs(TypeEnter.java:266)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$Phase.completeEnvs(TypeEnter.java:266)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter$Phase.completeEnvs(TypeEnter.java:266)
    at jdk.compiler/com.sun.tools.javac.comp.TypeEnter.complete(TypeEnter.java:198)
    at jdk.compiler/com.sun.tools.javac.code.Symbol.complete(Symbol.java:642)

```

## 6. Testy Jednostkowe

Wykonano testy jednostkowe. Wykorzystanie framework'a zostało załączone w plikach projektu w test/java/com/example/tic\_tac\_toe\_app\_java\_project\_psk jako ExampleUnitTest.java



*Screen potwierdzający użycie frameworka*

## 7. Podział pracy oraz Podsumowanie

Każdy z członków projektu równomiernie uczestniczył w każdym etapie powstawania projektu. Pan Paweł włożył więcej pracy w tryb PLAY WITH AI, oraz naprawił napotkane błędy które uniemożliwiały nam dalszej pracy, a pan Bartek więcej pracy włożył w stworzenie systemu rozgrywki i tryb PLAY PVP. Każdy etap był konsultowany i opracowywany poprzez spotkania online na platformie Discord, jak i Messenger.

Podsumowując projekt był ciekawym doświadczeniem oraz uświadomił nam jak dużo pracy trzeba czasami włożyć w poszczególne etapy.

## 8. Link do GITLAB

[https://orkan.tu.kielce.pl/gitlab/PwGawzynski/TIC\\_TAC\\_TOE\\_APP\\_JAVA\\_PROJECT\\_PSK?fbclid=IwAR0qqFb7Q1sj1USzFDybmjazDIYsVUDKIMwjKD1jBfJrNxFf3E-wvjw1os0](https://orkan.tu.kielce.pl/gitlab/PwGawzynski/TIC_TAC_TOE_APP_JAVA_PROJECT_PSK?fbclid=IwAR0qqFb7Q1sj1USzFDybmjazDIYsVUDKIMwjKD1jBfJrNxFf3E-wvjw1os0)