



AIRLOGO

v1.0.0

**DOKUMENTACJA
APLIKACJI**

ZESPÓŁ AIRTEAM
S. HINSINGER, S. KUCHARSKI,
M. PRUS, K. KOZŁOWSKI, K. MACHURA

SPIS TREŚCI

1. WSTĘP	3
2. INSTALACJA	4
3. INTERFEJS APLIKACJI	5
4. ARCHITEKTURA APLIKACJI	8

WSTĘP

Czym jest aplikacja AirLogo?

AirLogo to aplikacja, która pomaga w stawianiu pierwszych kroków w świecie programowania dla najmłodszych i nie tylko.

Skupia się ona na przesuwaniu po planszy żółwia za pomocą wywoływania rozmaitych komend, dzięki czemu można tworzyć np. figury.

Polecenia wydawane przez użytkownika są bazowane na tych, które były zaimplementowane w programie ATARI LOGO na konsolach ATARI w latach 80.

Rozbudowany system tworzenia procedur umożliwia jeszcze lepszą zabawę i możliwość rozwijania swoich umiejętności.

Swoją pracę można zapisać i udostępnić, za pomocą opcji zapisu planszy lub zrzutu ekranu, która widnieje na pasku przybornika.

INSTALACJA

WYMAGANIA SYSTEMOWE:

- Windows 10 (32 bit / 64 bit)*,
- 2GB+ RAM,
- ~180+ MB Przestrzeni Dyskowej

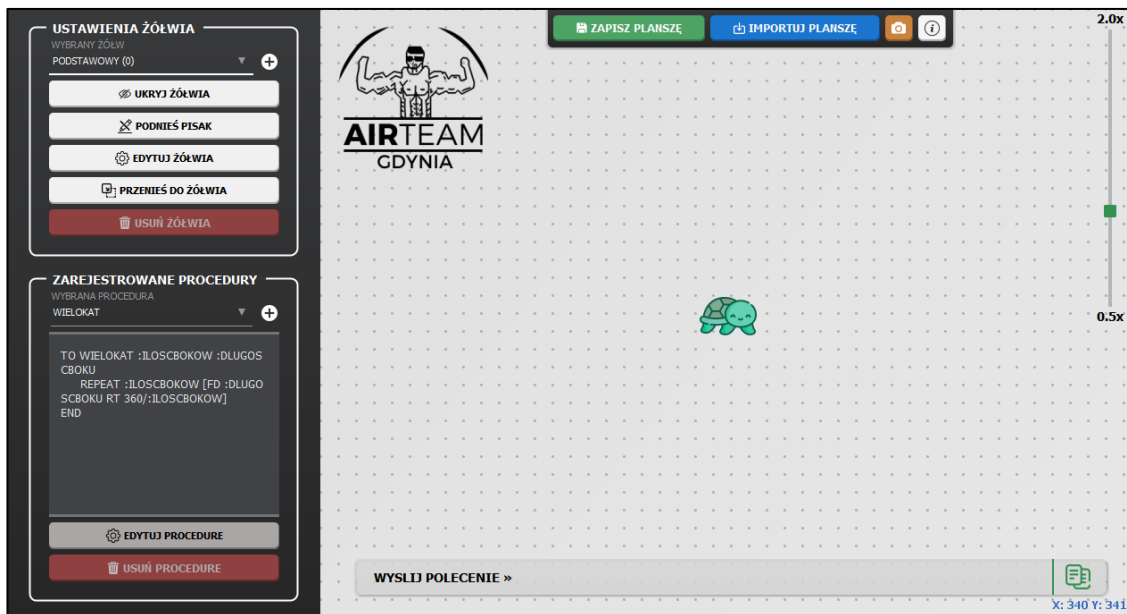
SPOSÓB INSTALACJI:

1. Należy pobrać instalator aplikacji wedle korzystanej wersji architektury systemu, na której będzie instalowana aplikacja. (32bit lub 64bit). ([LINK DO WYDANIA NA GITHUB.COM](#))
2. Po pobraniu, należy otworzyć instalator za pomocą dwukrotnego kliknięcia.
3. Gdy instalator się otworzy, należy postępować zgodnie ze wskazówkami jakie będą wyświetlane.
4. Po zainstalowaniu, jeżeli została wybrana opcja „Utwórz ikonę Pulpite”, to skrót aplikacji można znaleźć na pulpicie lub wyszukując go po nazwie w wyszukiwarce aplikacji.

* Aplikacja była testowana na wersji systemu Windows 10, lecz nie wykluczone jest wsparcie, także dla innych wersji systemów Windows.

INTERFEJS APLIKACJI

GŁÓWNY INTERFEJS:



Aplikacja jest podzielona na dwie sekcje. Ustawienia znajdują się po lewej stronie, umożliwiając one szybkie edytowanie właściwości żółwia i jego środowiska. Plansza żółwia po której się on porusza znajdują się po prawej stronie. Umożliwia ona przemieszczanie się także widoku użytkownika podczas przytrzymania lewego przycisku myszy, co pozwala na nieograniczoną zabawę użytkownika.

INTERFEJS APLIKACJI

PANEL USTAWIEŃ:



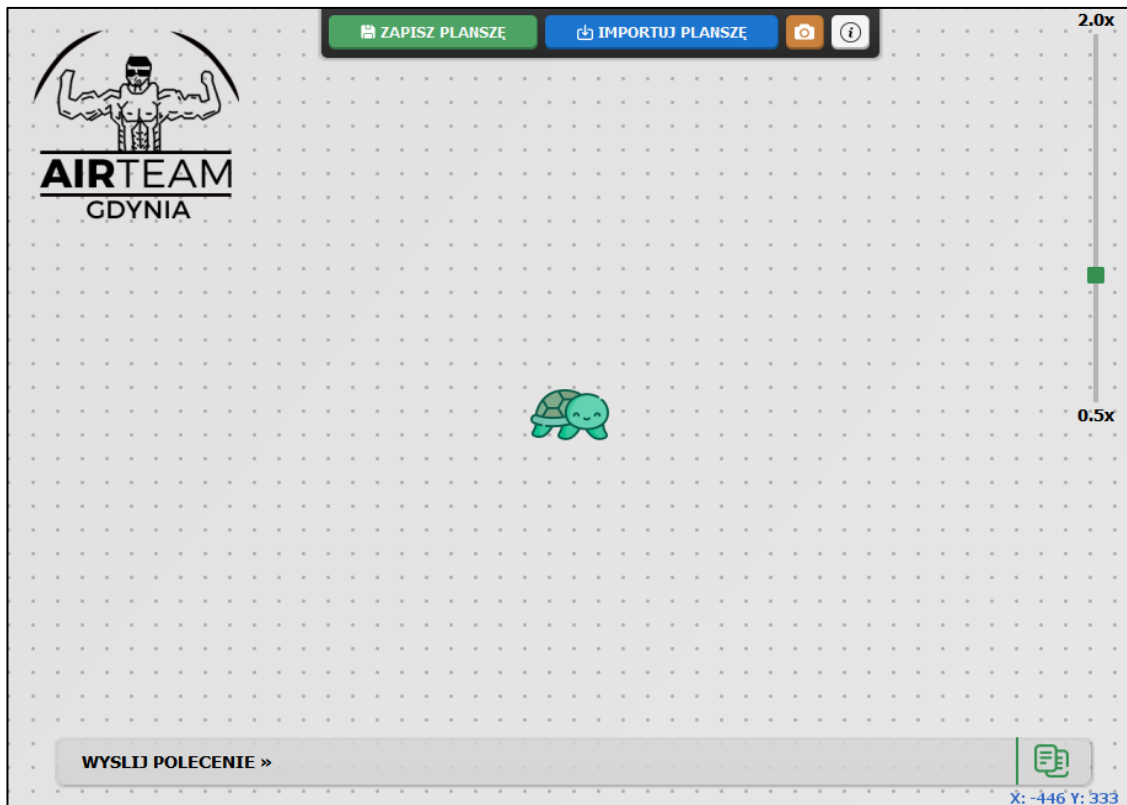
Panel ustawień jest podzielony na dwie sekcje.

Ustawienia żółwia pozwalają na zmianę parametrów żółwia, a także dodanie nowych żółwi za pomocą przycisku z plusem.

Zarejestrowane procedury umożliwiają przeglądanie i edytowanie utworzonych wcześniej procedur lub dodanie nowych za pomocą przycisku z plusem.

INTERFEJS APLIKACJI

PLANSZA ŻÓŁWIA:



Plansza żółwia jest podzielona na trzy elementy. Przybornik z opcją zapisu planszy lub jej importowaniem, a także z możliwością zapisu zrzutu ekranu znajduje się na górze. Na dole znajduje się wiersz w którym można wysłać polecenia. Po prawej stronie wiersza poleceń znajduje się przycisk umożliwiający pokazanie/ukrycie konsoli. W tle znajduje się strefa planszy żółwia. Po prawej stronie znajduje się także suwak do zmiany skali planszy.

ARCHITEKTURA APLIKACJI

OPIS TECHNICZNY APLIKACJI:

Aplikacja została napisana w języku Java. Interfejs został utworzony dzięki wbudowanej bibliotece „SWING”. Do jej wyeksportowania zastosowano oprogramowanie do budowania aplikacji „Apache Maven”.

ZASTOSOWANE BIBLIOTEKI:

- MIGLAYOUT (com.miglayout)
- JGOODIES (com.jgoodies)
- JSON (org.json)
- JUNIT (org.junit.jupiter)

DODATKOWE INFORMACJE:

Wykorzystane w aplikacji ikony zostały pobrane ze strony <https://www.flaticon.com>

Testy automatyczne znajdują się w katalogu plików źródłowych „SRC/TEST/JAVA” i są uruchamiane podczas eksportowania aplikacji za pomocą narzędzia Apache Maven.

ARCHITEKTURA APLIKACJI

OPIS DZIAŁANIA NIEKTÓRYCH KLAS:

Każde polecenie jest inną klasą zaczynającą się od prefixu „**CMD_**” i znajduje się w pakiecie „**airlogo.commands**”. Każda klasa komendy posiada statyczną metodę „**execute**”, która przyjmuje listę procedur i wykonuje konkretne działania związane ze środowiskiem żółwia.

Do zainicjowania komend służy metoda „**parse**” z klasy „**CommandManager**”, która przyjmuje listę procedur, która jest filtrowana w celu określenia jakim poleceniem jest 1-szy element, który następnie służy wywołaniu metody „**execute**” z klasy jemu odpowiadającej.

Do przetwarzania równań matematycznych służy metoda „**parseMath**” z klasy „**CommandManager**”, która przyjmuje równanie matematyczne w postaci tekstu, a także id żółwia, który zastępuje wyraz „**WHO**”. Do przetwarzania równania został wykorzystany silnik JavaScript, który za pomocą metody „**eval(równanie)**” zwraca obliczoną wartość wyrażenia.

ARCHITEKTURA APLIKACJI

Klasa „**Turtle**” służy tworzeniu obiektów żółwi. Posiada w sobie wszystkie jego parametry (listę przesunięć, współrzędne względem planszy, imię, obrót, itd.)

Za działanie procedur odpowiada klasa „**FunctionManager**”, która posiada w sobie statyczny parametr mapy zarejestrowanych funkcji. Metoda „**parse**” służy do odczytywania zapisanej funkcji i wykonywania jej poprzez odczytanie zapisanych komend i wywołanie ich przy pomocy metody „**parse**” z klasy „**CommandManager**”.

Za zapisywanie i importowanie planszy odpowiada klasa „**SaveManager**”. Informacje na temat planszy i środowiska żółwia są zapisywane w formacie **JSON**.

Klasa „**GraphicsUtility**” zawiera w sobie metody pomocnicze do edytowania obiektów graficznych.

Klasa „**NTSCUtility**” posiada listę wszystkich kolorów odpowiadających tym z formatu NTSC zapisanych w formacie HEX, które można odczytać korzystając z metody „**getAtariColorFromNumber**” podając liczbę od 0 do 127.