

## KTULU

# Projekt techniczny

## Autorzy

Marcin Chrzanowski

Klaudia Laks

Michał Naruniec

Karol Pieniący

## Platforma

Aplikacja powstanie przede wszystkim na platformę mobilną Android, ale użyte technologie mają pozwalać na łatwe wprowadzenie wersji na urządzenia z systemem iOS.

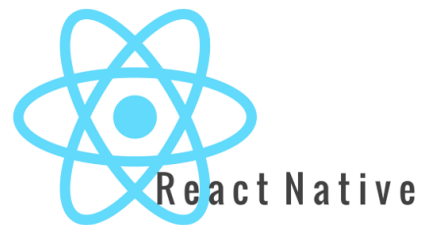
## Moduły

- ♦ Podręcznik
- ♦ Moduł przygotowania gry
- ♦ Moduł prowadzenia gry

## Nie planujemy korzystać z

- ♦ Bazy danych
- ♦ Usług sieciowych

Aplikacja  
będzie  
napisana we  
frameworku  
React Native.



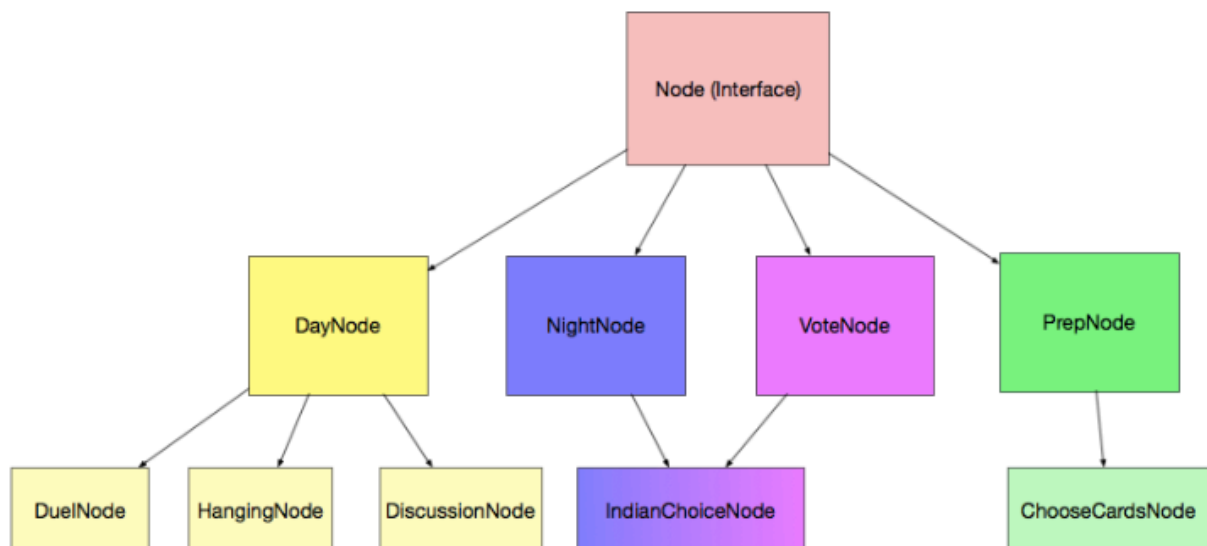
React Native enables you to build world-class application experiences on native platforms using a consistent developer experience based on JavaScript and React. The focus of React Native is on developer efficiency across all the platforms you care about - learn once, write anywhere. Facebook uses React Native in multiple production apps and will continue investing in React Native. React Native is the best thing ever. (No)



Aplikacja jest napisana w frameworku React Native z użyciem Reduxa do zarządzania stanami oraz języku JavaScript ES6.

Jednym z wymagań Reduxa jest niemutowalność stanu - zawsze zwracany jest nowy obiekt, oparty na poprzednim.

Historia stanów gry jest utrzymywana na stosie o ograniczonej pojemności. Elementy stosu to wierzchołki w „maszynie stanowej”.



**Current\_game\_state** zawiera informację:

- o tym kto wciąż żyje
- gdzie aktualnie znajduje się posązek
- który jest dzień
- itp.

Metoda **Game.action(Message m)** powinna być wykonywana podczas przejścia do kolejnego stanu. Parametr message zawiera informację o wykonanej przez gracza akcji (np. kliknięciu w konkretny guzik, wyborze z listy). W zależności od wartości message metoda action ustala nowy stan gry i wkłada go na stos.

Metoda render() jest wymuszona przez framework React Native i odświeża widok przy aktualizacji stanu. Stan zawiera informacje potrzebne frameworkowi do poprawnego wygenerowania odpowiedniego widoku.

W stanie znajduje się trzypoziomowa identyfikacja aktualnego etapu gry i odpowiadającego mu widoku:

- stage - setup, night, day, game\_over
- step - identyfikuje aktualny etap gry

- substep - charakteryzuje widok

Widoki wywoływane na podstawie substep są znormalizowane i wykorzystywane w wielu miejscach ze zmienionymi argumentami.

Interfejs graficzny jest oparty na systemie stylów imitującym CSS. Style poszczególnych elementów są wydzielone w oddzielnym pliku, gdzie możemy uniknąć redundancji.

