

Politechnika Poznańska
Wydział Elektryczny
Instytut Automatyki i Inżynierii Informatycznej



Piotr Maleszczuk
Łukasz Błaszczak
Przemysław Burdelak

Mobilna gra Lau Kata Kati

Poznań, 2017

Spis treści

1	Wstęp	3
1.1	Opis tematu	3
1.2	Założenia realizacyjne	3
2	Podział i harmonogram pracy	4
3	Funkcjonalności oferowane przez aplikację	5
3.1	Rozgrywka singleplayer	5
3.2	Rozgrywka multiplayer local	5
3.3	Rozgrywka multiplayer bluetooth	5
3.4	Funkcjonalność sklepu	5
3.5	Zmiana trudności AI	5
4	Wybór technologii informatycznych	6
4.1	System Latex	6
4.2	System Github	6
4.3	System Trello	6
4.4	Język programowania C#	6
4.5	Środowisko programistyczne Visual Studio	6
4.6	Środowisko Unity	6
5	Architektura	7
5.1	Model View Controller	7
5.2	Kontroler planszy	7
5.3	Kontroler tur	7
5.4	Kontroler logiki	7
5.5	Kontroler wizualizacji	7
5.6	Kontroler kliknięć	8
5.7	Kontroler Bluetooth	8
5.8	Kontroler aktora Bluetooth	8
5.9	Kontroler SI	8
5.10	Kontroler zapisu danych	8
5.11	Kontroler gromadzenia danych	8
5.12	Kontroler pauzy	8
5.13	Kontroler końca gry	9
5.14	Kontroler menu	9
6	Instrukcja użytkowania aplikacji	10
7	Podsumowanie	13
	Spis rysunków	13
	Spis tabel	13

1 Wstęp

1.1 Opis tematu

Realizacja gry Lau Kata Kati, polegająca na utworzeniu ogólnej architektury gry pozwalającej na przeprowadzenie rozgrywki zgodnie z zasadami, a następnie podłączenie maszyny grającej, która rywalizuje z użytkownikiem.

1.2 Założenia realizacyjne

Aplikacja przeznaczona na urządzenia mobilne z system Android w wersji minimum 4.0.3. Do implementacji gry został wykorzystany silnik Unity 5.5.2f1, który zapewnia szeroką gamę narzędzi potrzebnych do napisania kompletnej gry, tj. graficzny interfejs zarządzania sceną, pisanie skryptów w języku C# obsługujących poszczególne elementy gry, budowanie aplikacji na urządzenia Android. Do pisania skryptów wykorzystano środowisko Visual Studio.

Maszyna grająca swoje działanie opiera na strategii Minimax z Alfa-Beta cięciami. Algorytm pobiera aktualny stan planszy w postaci tablicy dwuwymiarowej, a następnie oblicza najlepszą ścieżkę do określonej głębokości drzewa za pomocą rekurencyjnej metody. Ścieżka wybierana jest na podstawie zysku punktów (punkty maszyny grającej - punkty gracza). Po wybraniu najlepszej drogi, pobierany jest z niej pierwszy ruch, a następnie wywołuje się metodę imitującą naciśnięcie odpowiedniego pionka i punktu przeznaczenia do którego ma zostać przeniesiony. Sposób, w którym imituje się naciśnięcie danego elementu został wybrany, ponieważ był to najszybszy i najprostszy sposób przemianowania dwuosobowej rozgrywki lokalnej na rozgrywkę ze sztuczną inteligencją.

Aplikacja posiada:

- Trzy tryby rozgrywki (singleplayer, multiplayer lokalny oraz przez połączenie Bluetooth),
- zmianę poziomu trudności maszyny grającej,
- zmianę skórek pionków,
- system reklam,
- system zbierania danych.

2 Podział i harmonogram pracy

W rozdziale przedstawiony został podział pracy między autorami projektu (Tabela 1). Harmonogram pracy widoczny jest w Tabeli 2.

<i>Autor</i>	<i>Podzadanie</i>
Piotr Maleszczuk	GUI, wizualizacja ruchów, obsługa różnych trybów gry, implementacja metody Minimax
Łukasz Błaszczak	Przygotowanie planszy, obsługa kliknięć ekranu, przeliczanie punktów, imitacja kliknięć
Przemysław Burdelak	Odnajdywanie możliwych ruchów pionka, budowanie dostępnych plansz, podłączenie maszyny grającej do rozgrywki

Tabela 1: Podział pracy

<i>Lp.</i>	<i>Opis zadań</i>
1.	Przygotowanie zdalnego repozytorium
2.	Przygotowanie środowiska programistycznego
3.	Przygotowanie środowiska Unity oraz założenie wstępnego projektu
4.	Rozpisanie projektu na Trello, podzielenie na poszczególne zadania wykonywane przez cały czas projektu.
5.	Utworzenie menu startowego gry (ekran główny, opcje, sklep)
6.	Utworzenie menu wyboru trybu (singleplayer, multiplayer, online multiplayer)
7.	Implementacja rozgrywki - logika gry
8.	Implementacja gry jednoosobowej - SI
9.	Komunikacja Bluetooth
10.	Implementacja gry wieloosobowej
11.	System reklam
12.	System zbierania danych
13.	Testy i ewentualne poprawki

Tabela 2: Harmonogram pracy

3 Funkcjonalności oferowane przez aplikację

3.1 Rozgrywka singleplayer

Rozgrywka realizowana z maszyną grającą opierającą się na strategii Minimax z Alfa-Beta cięciami. Tak jak w przypadku każdego trybu na początku losowany jest gracz rozpoczynający grę, następnie odbywa się całość rozgrywki z botem.

3.2 Rozgrywka multiplayer local

W tym przypadku mamy możliwość rozegrania gry dwuosobowej na jednym urządzeniu.

3.3 Rozgrywka multiplayer bluetooth

Możliwość rozegrania gry dwuosobowej pomiędzy dwoma urządzeniami na których wykonywane są ruchy poszczególnych graczy.

3.4 Funkcjonalność sklepu

Gra rozwinięta jest o możliwości zmiany widoku gry, konkretnie chodzi tutaj o wygląd pionków graczy. Wszystkie oferowane w aplikacji pionki możliwe są do obejrzenia w sklepie. Pionki można używać dopiero po wykupieniu ich za wirtualne monety nabywane bezpośrednio w rozgrywce bądź jako nagroda za obejrzenie reklamy wideo w menu (dostępne raz na 20 minut) za 10 wirtualnych monet, wszystkie pionki dostępne są w różnych cenach.

3.5 Zmiana trudności AI

Po wejściu w Opcje ("Options") możliwa jest również zmiana poziomu maszyny grającej, dostępnej w rozgrywce dla jednego gracza, odpowiadająca głębokości przeszukiwania drzewa a algorytmie minimax'a. Maksymalny możliwy poziom do wybrania to '9'.

4 Wybór technologii informatycznych

W rozdziale znajduje się opis wybranych technologii informatycznych. W kolejnych podsekcjach umieszczone zostały szczegółowe opisy technologii.

4.1 System Latex

Oprogramowanie przeznaczone do zautomatyzowanego składu tekstu, a także związany z nim język znaczników, służący do formatowania dokumentów tekstowych i tekstowo-graficznych.

4.2 System Github

Hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Wybrany ze względu na łatwość w użyciu.

4.3 System Trello

Narzędzie wykorzystane do zarządzania zadaniami podczas tworzenia projektu. Projekt w całości został rozpisany w systemie na zadania główne, składające się następnie z mniejszych szczegółowych podzadań. Pozwoliło to na przejrzysty i zaplanowany z góry sposób działania.

4.4 Język programowania C#

Język C# doskonale współpracuje ze środowiskiem Unity.

4.5 Środowisko programistyczne Visual Studio

Zintegrowane środowisko programistyczne firmy Microsoft. Jedno z najbardziej rozwiniętych tego typu narzędzie posiadających ze sobą wiele udogodnień. Pomimo swojej wielkości dobrze sprawdza się w każdego rodzaju projekcie. W porównaniu ze środowiskiem dostarczonym przez Unity (MonoDevelop) jest to narzędzie bardziej stabilne i wygodniejsze w użyciu.

4.6 Środowisko Unity

Zintegrowane środowisko firmy Unity Technologies używane do tworzenia gier wideo na PC, konsole, urządzenia mobilne.

Wyborem w tym przypadku kierowaliśmy się głównie na podstawie informacji z internetu oraz niewielką znajomością tego narzędzia przez jednego z członków naszego zespołu. Środowisko pomimo tego, że na pierwszy rzut oka może wydawać się bardzo skomplikowane, po krótkim zapoznaniu okazuje się bardzo przyjemne w użyciu, a do tego ogromna ilość materiałów pomocniczych znajdujących się w internecie.

5 Architektura

Rozdział opisuje architekturę, która została wykorzystana w projekcie oraz ogólny opis funkcjonalności zaimplementowanych w danym module.

5.1 Model View Controller

Wzorzec projektowy służący do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika. Zastosowanie go w projekcie umożliwia wygodny podział na kontrolery, które przejmują odpowiedzialność na zadaną funkcjonalnością wykonywaną na modelach. Widoki umożliwiają oddzielenie warstwy prezentacji od wszelkich operacji wewnętrznych.

5.2 Kontroler planszy

Kontroler odpowiedzialny za stworzenie oraz obsługę macierzy stanu planszy, na której opiera się cała rozgrywka. Macierz początkowa ma postać:

$$\begin{vmatrix} 1 & 1 & 1 & -1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 0 & 2 & 2 & 2 \\ 1 & 1 & 1 & -1 & 2 & 2 & 2 \end{vmatrix}$$

-1 - pole nieużywane

0 - pole puste

1 - pole z pionkiem gracza 1

2 - pole z pionkiem gracza 2

5.3 Kontroler tur

Główny kontroler odpowiedzialny za przebieg całej rozgrywki.

Odpowiada za zmianę tur graczy, przeliczanie punktów, zakończenie gry.

5.4 Kontroler logiki

Kontroler na podstawie pozycji pionka oraz macierzy planszy zwraca możliwe dla niego ruchy oraz flagę czy dostępne jest kolejne bicie w tej samej turze.

5.5 Kontroler wizualizacji

Jest to kontroler obsługujący wizualizację przebiegu rozgrywki. Przeszukuje macierz pozycji pionków - odpowiednik wirtualnej planszy. Gdy kontroler na danej pozycji w macierzy napotka wartość '1' lub '2' ustawia na odpowiadającym jej polu pionek. Jeżeli wartość ta wynosi '0' usuwa pionek z danego pola planszy.

Dodatkowo kontroler obsługuje funkcję wyświetlania podpowiedzi możliwych ruchów dla danego pionka po jego naciśnięciu.

5.6 Kontroler kliknięć

Pobiera punkt kliknięcia i szuka elementu tj. pionka, pustego pola, który w danym miejscu się znajduje, a następnie przesyła go do kontrolera plan-szy.

5.7 Kontroler Bluetooth

Uruchamia moduł bluetooth, zależnie od wybranego trybu jako klient lub serwer. Korzysta ze specjalnego pluginu, który mapuje działanie bluetooth na wbudowaną obsługę połączenia sieciowego w Unity.

5.8 Kontroler aktora Bluetooth

W tym kontrolerze, odbywa się główna komunikacja bluetooth pomiędzy klientem a serwerem. Komunikaty wysyłane są w prostej czterocyfrowej postaci:

s t x y

,gdzie:

s – status komunikatu 1-setup, 2-waiting, 3-transfer

t – tura w rozgrywce

x – pozycja x kliknięcia

y – pozycja y kliknięcia

5.9 Kontroler SI

Odpowiedzialny za działanie sztucznej inteligencji podczas rozgrywki jednoosobowej. Maszyna grająca korzysta ze strategii MiniMax z Alfa-Beta cięciami.

5.10 Kontroler zapisu danych

Odpowiada za przechowywanie danych użytkownika tj. stan konta, kupione skórki. Dodatkowo umożliwia komunikację pomiędzy scenami.

5.11 Kontroler gromadzenia danych

Kontroler zapisuje do pliku informacje na temat użytkownika: specyfikacja telefonu, lokalizacja, oglądanie reklam, długość korzystania z aplikacji.

5.12 Kontroler pauzy

Umożliwia wykonanie pauzy podczas rozgrywki. Umożliwia wyjście do głównego menu, zrestartowanie rozgrywki oraz wznowienie aktualnej.

5.13 Kontroler końca gry

Wywoływany po zakończeniu rozgrywki. Uruchamia reklamy, a następnie umożliwia przejście do menu głównego.

5.14 Kontroler menu

Odpowiada za działanie całego menu głównego: obsługa sklepu, zmiany ustawień aplikacji, uruchamianie poszczególnych trybów gry.

6 Instrukcja użytkowania aplikacji

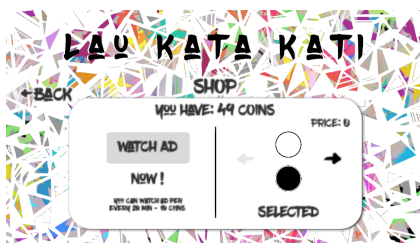
Aplikacja przeznaczona na urządzenia mobilne z system Android w wersji minimum 4.0.3.

1. Po uruchomieniu aplikacji wyświetli się menu główne(rys.1.).



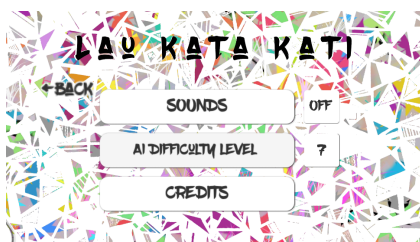
Rys. 1: Ekran główny

2. W menu „Shop” istnieje opcja kupna/zmiany skórki pionków za zebrane punkty oraz możliwość obejrzenia reklamy co 20 minut – reklama dodaje 10 punktów (rys.2.).



Rys. 2: Ekran sklepu

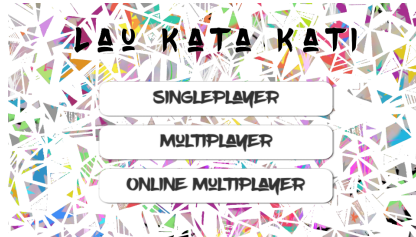
3. W menu „Options” można ustawić poziom trudności maszyny grającej od 1 do 9 (rys.3.).



Rys. 3: Ekran opcji

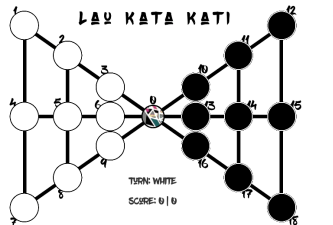
4. Po naciśnięciu „*Start game*” ukażą się trzy tryby rozgrywki (rys.4.):

- *Singleplayer* – rozgrywka przeciwko maszynie grającej,
- *Multiplayer* – lokalna rozgrywka dwuosobowa,
- *Online Multiplayer* – rozgrywka dwuosobowa przez Bluetooth.



Rys. 4: Ekran wyboru trybu gry

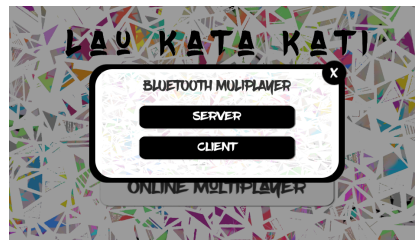
5. Po wybraniu trybu „*Singleplayer*” zostanie rozpoczęta rozgrywka przeciwko SI (rys.5.).



Rys. 5: Widok trybu singleplayer

6. Wybranie trybu „*Multiplayer*” rozpocznie lokalną rozgrywkę multiplayer.

7. Po wybraniu trybu „*Online Multiplayer*” jeden z graczy musi utworzyć serwer gry wybierając opcję „*Server*”. Po utworzeniu serwera gracz drugi łączy się przez Bluetooth wybierając opcję „*Client*”. Rogrywka rozpocznie się po wylosowaniu i przesłaniu tury do klienta (rys.6.).



Rys. 6: Opcje trybu Online multiplayer

7 Podsumowanie

Pomimo napotkanych błędów wszystkie założenia projektu zostały zrealizowane pomyślnie.

Projekt jest wrat kontynuacji. Możliwości jego rozwoju są wręcz nieograniczone. Główny rdzeń aplikacji jest przygotowany do kolejnych rozszerzeń, co zawdzięczamy architekturze MVC.

Przykładowe kierunki rozwoju:

- Rozgrywka sieciowa
- Parowanie urządzeń za pomocą NFC
- Ranking graczy
- Nowe skórki pionków
- Skórki plansz

Spis rysunków

1	Ekran główny	10
2	Ekran sklepu	10
3	Ekran opcji	10
4	Ekran wyboru trybu gry	11
5	Widok trybu singleplayer	11
6	Opcje trybu Online multiplayer	12

Spis tablic

1	Podział pracy	4
2	Harmonogram pracy	4