

Hra šachy

Projekt ITU, 2019/2020, Z

Autor: Peter Horňák (xhorna14)

Další členové týmu: Matej Janček (xjance00), Martin Koči (xkocim05)

Datum odevzdání: 05-10-2019

1. Zadání projektu

Hledejte vhodné zaměření zadání ve svém okolí, včetně relevantních uživatelů. Ideálně hledejte mimo studenty ITU.

- Náš projekt sme zamerali na ľudí, ktorí sa chcú zlepšiť v hre šach, poprípade na hráčov ktorí sú úplní amatéri a poznajú iba pravidlá. Užívatelia, ktorých odozva bude pre nás smerodajná sú absolventi ITU, jeden z nich je amatér a druhý nevie hrať šach vôbec.

Co uživatel potřebuje? Jaký problém (pracovní úkon, proces apod.) má uživatel zefektivnit?

- Naša aplikácia má užívateľovi pomôcť a spríjemniť učenie sa hry šach.

Jak to dělá (řeší) uživatel dnes? V čem je současné řešení nedostatečné/pomalé/neefektivní?

- Vo veľa prípadoch musí užívateľ platiť predplatné, tak isto riešenia sú až moc komplikované, čo spôsobuje, že sa vytráca z tejto hry zábava a odrádza to užívateľa od tejto hry.

Dekomponujte úlohu na řadu úkonů/procesů. Z jakých kroků se procesy skládají?

- Zapnúť hru.
- Vybrať si farbu figúrok za ktoré bude užívateľ hrať.
- Hrať hru.
 - Počúvať rady od aplikácie.
 - Opakovať ťah.
 - Zvoliť si náročnosť súpera.
 - Vymeniť si strany.
- Zlepšovať sa.
- Baviť sa.

Co je cílem uživatele/procesu? Co uživatel vlastně opravdu potřebuje? Revidujte proces. Nelze kroky v procesu zredukovat nebo jinak efektivněji/srozumitelněji seskládat?

- Cieľom procesu je pobaviť užívateľa a zlepšiť jeho logické a taktické rozmýšľanie.
- Užívateľ potrebuje spríjemniť a zjednodušiť tento proces.

Existujúce riešenia.

- [lichess](#)
- [learningchess](#)

Dôvod prečo sme si vybrali zrovna šach, kde sa ľudia môžu zlepšovať je ten, že človek chce cítiť nejaký progres, to, že sa zlepšuje, ale zároveň sa chce pri tom baviť. Nechceme odradiť užívateľa od našej aplikácie tým, že budeme implementovať množstvo módov ale takým spôsobom, že pri normálnom hraní ho aplikácia upozorní na jeho chyby a poskytne mu možnosti ako by mohol ťah lepšie zahrať. Myslím si, že zaujímavou funkcionalitou pre tréning by bolo, keby si užívateľ mohol počas hry meniť strany za ktoré hrá. Tak isto chceme poskytnúť možnosť klasickej hry bez pomôcok.

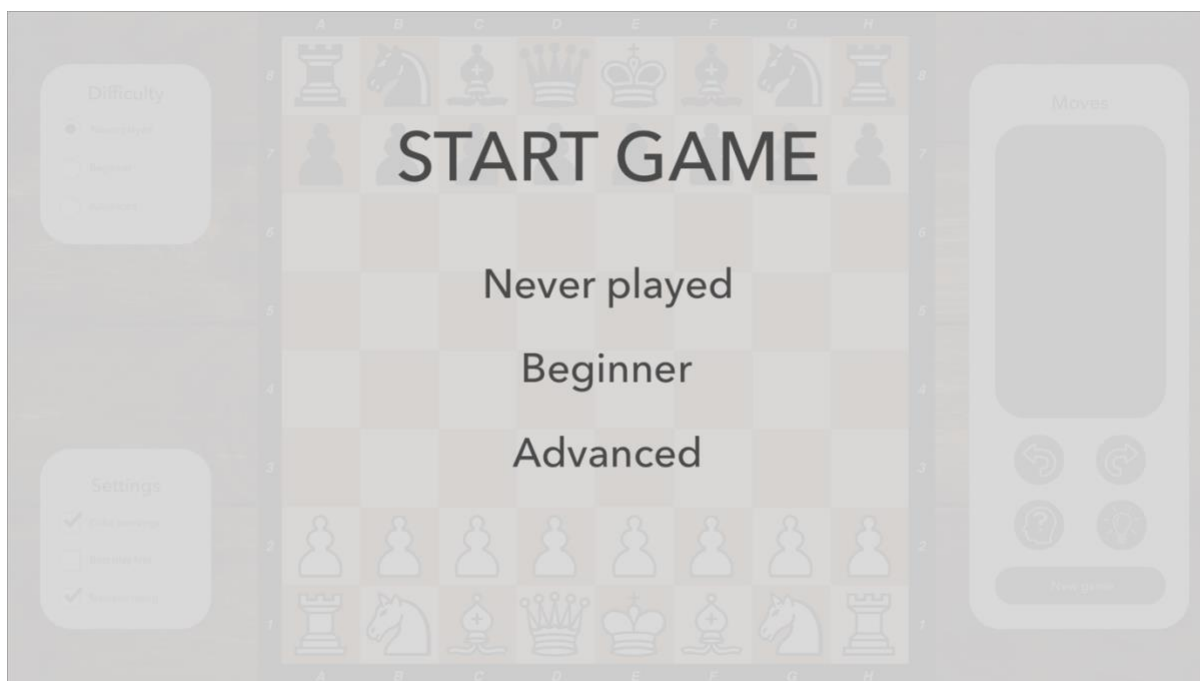
2. Návrh – GUI, backend

S jakými daty je třeba manipulovat a jak? Jaké informace/akce jsou potřeba od uživatele? Jak lze tyto akce uživateli zpřístupnit/nabídnout?

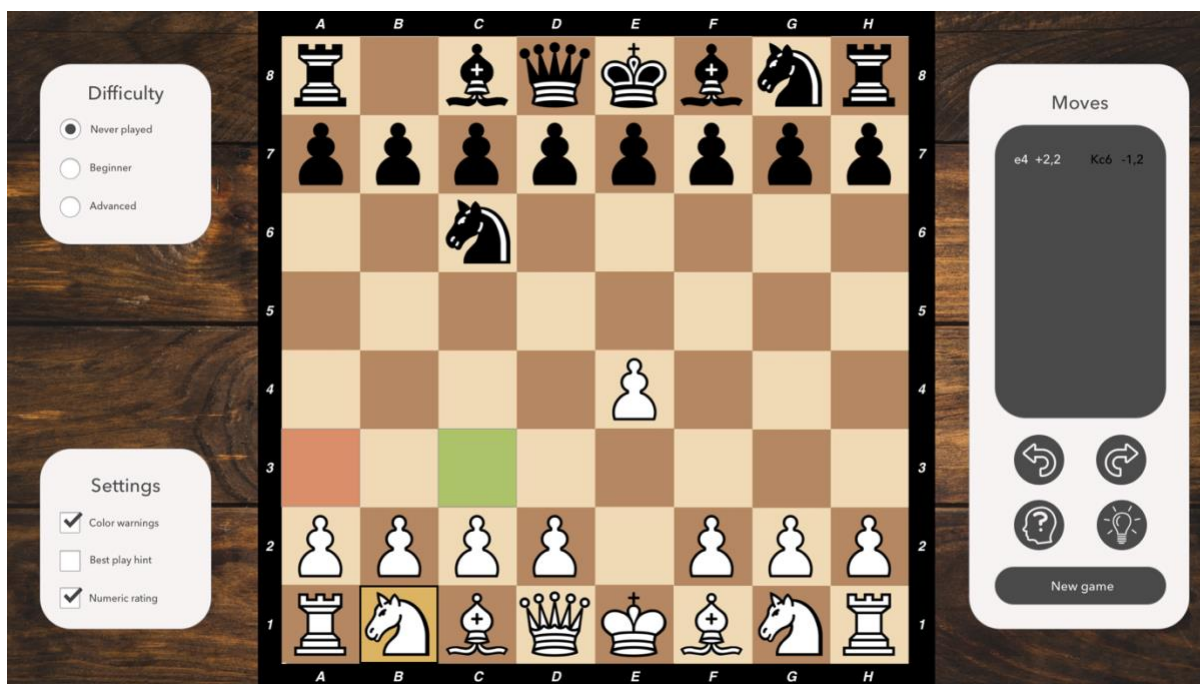
- Šach je v tomto prípade veľmi priamočiary, dáta na s ktorými je treba pracovať sú najmä nastavenia užívateľa (obtiažnosť, rady od aplikácie), jeho ťahy a popríklad kroky späť a reštartovanie hry. Naším cieľom je aby ovládanie figúriek bolo intuitívne, stlačiť na figurku, presunúť kurzor a potvrdiť kliknutím pozíciu kde premiestniť danú figúrku. Nastavenia bude možné nastaviť jak pred začiatkom hry, kde sa bude dať zvoliť s viacerých obtiažností, ale zároveň sa dajú upraviť počas hry.

Co je potřeba uživateli vizualizovat/zobrazit? Kolik dat a jakého typu je potřeba zobrazit? Jaké jsou různé možnosti zobrazení těchto dat?

- Potreba mu zobrazit aké sú možné jeho ťahy s figúrkou, ktorú zvolí. Tak isto užívateľovi treba vizualizovať ktoré kroky sú správne a ktoré naopak nesprávne. Jedna z možností, ktorá sa ponúka je po prvom kliknutí na figúrku, ktorou chce uskutočniť ťah, je farebne vyznačiť jednotlivé možnosti pohybu, zelená by znamenala, že ťah patrí k tým lepším, oranžová, že ťah nie je zlý ale profesionál by ho nespravil a červená, že ťah je chyba, ktorú vie súper ľahko potrestať. Tak isto užívateľ by ocenil keby jednotlivé možnosti boli číselne ohodnotené, kde väčšie číslo by znamenalo lepší ťah.



Menu hry uvidí užívateľ ihneď po spustení. Myslím si, že menu by malo byť jednoduché a hráč si vyberie iba svoju zručnosť v hre. Nechceme totižto hráča odradiť hneď zo začiatku zdĺhavým nastavovaním a možnosťami. V prípade, že si vyberie možnosť “Never played” sa mu spustí základný tutorial. Strana za ktorú bude užívateľ hrať sa vyberie náhodne. Treba predsa trénovať za obidve strany.



Počas hrania môže hráč vyberať obtiažnosť a AI, tak isto má možnosť si vypínať a zapínať niektoré pomôcky. Na pravej strane, sa ukazujú ťahy každej strany, pričom sa tak isto ukazuje aj hodnotenie ťahu relatívne k užívateľovi, kde ťah ktorý má hodnotu kladnú

naznačuje, že zlepšil jeho šancu k výhre a naopak, ťah so zápornou hodnotou zhoršil jeho šancu k výhre. Užívateľ má možnosť sa vrátiť po odohratom ťahu späť a tak isto aj túto akciu zvrátiť. Užívateľ má možnosť si spustiť opäť tutoriál alebo si nechať poradiť od aplikácie. Nakoniec môže vytvoriť novú hru, pričom v tomto prípade už dostane možnosť vybrať si strany. Pri ťahu ako je vidno na obrázku, mu aplikácia farebne označuje kvalitu ťahu, viz. vyššie popísaný princíp.

3. Architektúra aplikácie a návrh testovania

Výběr a zdůvodnění technologií, výběr a popis relevantních komponent, návrh architektury aplikace a specifických částí aplikace a GUI.

- Keďže sme si vybrali web ako našu platformu, ponúka sa nám viac spôsobov ako projekt implementovať. Prirodzene sa nám však ponúka možnosť tzv. Single-page application, ktorá dynamicky a interaktívne reaguje na akcie užívateľa bez potreby obnovovania stránky.
- Existuje viac technológií napríklad Ajax, Websockets alebo rôzne JavaScriptové frameworky ako React, AngularJS alebo Vue.js. Naším výberom bude posledný menovaný. Hlavným z dôvodov je, že má veľmi obsiahlu dokumentáciu, čo nám pomôže s vývojom, jeho jednoduchou integráciu a flexibilitou a hlavne tento framework bol navrhnutý pre architektúru MVVM (Model-View-ViewModel), ktorá sa využíva pre dynamické stránky.
- Ako je spomenuté vyššie, aplikácia bude pozostávať z 3 hlavných komponentov model, view a view-model. Model bude obsahovať jednotlivé grafické prvky, ktoré užívateľ uvidí, tieto prvky budú naviazané na model, čiže dáta, pomocou view-modelu, ktorý bude sprostredkovať komunikáciu medzi nimi. Hlavné prvky view bude šachovnica a jej jednotlivé figúrky. V modely bude implementovaná logika hry a AI, ktoré bude vyhodnocovať jednotlivé ťahy a ich hodnotu.
- Naša aplikácia nebude implementovať MVC ale MVVM, ktoré sa líšia minimálne.

Návrh a realizace experimentů, definice otázek/hypotéz, diskuze výsledků a vyvození závěrů.

- Testovaním chceme zistiť intuitívnosť GUI a náročnosť pre orientovanie sa v ňom. Tak isto sa zameriame na testovanie radenia užívateľom, chceme zistiť správny spôsob grafického poskytovania ťahov.
- Prvou fázou testovania bude prebiehať takým spôsobom, že sa stretneme s jednotlivými užívateľmi a predstavíme im našu aplikáciu. Ich úlohou bude spustiť aplikáciu a hrať hru. Popri tom však budú požiadaní aby nahlas rozprávali ich myšlienky. V tomto testovaní budeme pozorovať ako naše GUI je priateľské pre užívateľov a ako rýchlo si osvoja jeho používanie. Tak isto budeme môcť spozorovať, ktoré prvky GUI sú máťúce, poprípade ich užívateľ nevie nájsť alebo nevedia čo znamenajú.
- Cieľom druhej fázy testovania bude zistiť, ktorý spôsob radenia je u užívateľov najobľúbenejší. Každý testovaný, bude mať za úlohu odohrať 3 hry, pričom v každej z hier bude upravený spôsob doručovania pomôcok hráčovi. V každej hre budú ťahy ohodnotené farebne pomocou farieb červená, oranžová a zelená. V prvej hre bude

hodnota ťahu označená iba týmito tromi farbami, čiže to bude znamenať 3 možné hodnoty. V druhej hre budú jednotlivé farby ešte rozdelené podľa ich svetlosti, kde tmavší odtieň farby bude znamenať horší ťah v jednotlivkej skupine. V tretej hre budú farby naopak rozdelené podľa priesvitnosti, pričom čím priesvitnejšie budú tým menej významná bude jednotlivá farba. V príklade to znamená, priesvitnejšia zelená je horší ťah ako sýta zelená ale priesvitnejšia červená je lepší ťah ako sýta červená, pri oranžovej farbe to bude fungovať rovnako ako pri zelenej. Na konci testovania užívateľ určí poradie jednotlivých hier, ktorá sa mu hrala najlepšie až po najhoršiu.

- Z testovania sme zistili, že numerické hodnoty sú pre užívateľov príliš komplikované a vytráca sa intuitívnosť. Taktiež sme zistili, že názov “Never played” pri úvodnom výbere skúseností hráča nie je ideálny, pretože hráči, ktorý nepoznajú pravidlá si túto možnosť nevyberú aj keď bola myslená práve pre ľudí čo nepoznajú túto hru. Z druhého testu sme zistili, že ľudia preferovali jednoduchý spôsob značenia ťahov, kde ťahy boli označené iba 3 farbami, bez rôznych oddelovaní.
- Názov “Never played” bude potreba zmeniť tak aby hráči, ktorí vôbec nehrajú šach boli motivovaní na to kliknúť aj keď si to nemusia uvedomovať.
- V druhom prípade vyberieme ten spôsob, ktorý užívatelia vybrali.

4. Implementace a výsledky testování

- Testovanie bolo popísané a vykonané už v predchádzajúcej časti.
- Implementáciu sme si rozdelili do 3 častí, kde každú časť spravil jeden z tímu. Časti boli:
 - Hlavné menu (xkocim05)
 - Návod pre užívateľov (xjance00)
 - Šachovnica a hranie (xhorna14)
- Moja časť implementácie spočívala v spravení štýlu pre šachovnicu, tlačidiel na jej obsluhu a samotné hranie šachu.
- Celý problém som si rozložil do pod problémov a postupne som skladal jednotlivé časti. Keďže sme používali *Vue.js* tak som vyrobil komponentu *Board*, ktorý obsahoval pole komponentu *ChessBlock*. Základ bol vytvoriť samotnú šachovnicu pomocou HTML a CSS v komponente *Board*. To sa mi podarilo na základe tagov `<div>`, ktoré mi zaručili rozloženie jednotlivých políčok. Component *ChessBlock* nasledovne reaguje na kliknutie a volá tzv. mutácie. Mutácie sú metódy, ktoré naväzujú, naše views na view-model. Náš view-model je tzv. *Store*, čo je plugin do *Vue.js* ktorý zaručuje na základe zmeny v backende, reakciu všetkých potrebných komponent. Ako backend sme použili známu knižnicu [chess.js](#), ktorá sa používa v najznámejších šachových aplikáciach. Tak isto sme zakomponovali AI nazvané [p4wn](#), ktoré hrá proti užívateľovi a snaží sa mu poradiť najlepšie ťahy. Implementáciu sme sa snažili dodržať podľa nášho výskumu a testovania.