

# Závěrečný report

Tým: Daniel Pernička (xpernid00), Juraj Mesík (xmesikj00), Patrik Mokruša (xmokrup00)

## Názov a téma:

Cieľom nášho projektu bolo vytvoriť program „Piškvorky“. Ide o klasickú a jednoduchú hru ktorú budú používatelia môcť hrať kdekoľvek a kedykoľvek. Je dôležité aby bol program jednoduchý, a efektívny.

## Rozdelenie Práce:

Rozhodli sme sa vytvoriť tri rôzne varianty pre hru piškvorky. Naším cieľom je ukázať tri známe technológie JavaScriptu a možnosť v každej z nich spracovať zadanie.

## Zapracování připomínek z Kontrolní prezentace

xmokrup00:

Na kontrolní prezentaci jsme dostali připomínku, že naše řešení je moc jednoduché, a že by se hodilo udělat hrací plochu o větší velikosti, a nějak vyřešit navigaci na ní.

V mojím řešení jsem si udělal container (div), který responzivně mění svoji velikost, podle parametrů viewportu. V tomto containeru je tabulka s hrací plochou. Venkovní container má nastaveno v css, že když overflowne, tak začne scrollovat. Jelikož jsem svoji implementaci navrhoval pro mobilní zařízení, tak se pohybuje po hracím poli jenom posunutím prstem (popřípadě se dá pěkně pohybovat na touchpadu nebo zmáčknutím kolečka myši). V nastavení jsem přidal 2 slidery, které mění parametry hry. Dále jsem dolů přidal slider, který udává scale vnitřní herní plochy, což simuluje oddalování pro lepší orientaci na herní ploše. Scale se mění pomocí CSS transform.

Dále jsem přidal v rámci refaktORIZACE nastavení jsem přidal veškeré slidery a přepínače jako slider pro upravení velikosti hrací plochy, slider pro nastavení počtu políček na výhru a toggle pro historii a AI oponenta.

## Změny v implementaci

- změny týkající se připomínek z kontrolní prezentace (viz. výše)
- důležitým částem UI jsem dal absolutní polohu
- reorganizace funkčních tlačítek do nastavení

xmesikj00:

- Rozsah manipulácie s dátami:

Po kontrolnej prezentácii sme do BE implementácie pridali nové atribúty: **veľkosť hracej plochy, počet výherných políček, súradnice výherných políček a príznak**

**hry jedného hráča.** Manipuláciu s dátami sme upravili hlavne pre možnosť hry na hracej ploche väčšej, než 3x3. Užívateľ má po novom možnosť meniť veľkosť hracej plochy a počet políček potrebných na výhru podľa vlastnej preferencie.

## ■ Rozšírenie hernej funkcionality / spôsobu interakcie:

1. **Veľkosť hracej plochy a počet výherných políček** som pridal do okna s nastaveniami v hlavnom menu. Tam je užívateľ upozornený, že zmeny týchto nastavení spôsobia vymazanie aktuálne dosiahnutého herného progresu, nakoľko program pre správne vyrenderovanie novej veľkosti hracej plochy volá funkciu *gameReset()*.

Zmeny veľkosti a orientáciu po hracej ploche som v mojej verzii hry vyriešil zmenšením políček a nastavením veľkosti symbolu hráča ( X alebo O ) relatívne na veľkosť políčka ( v predošlej verzii bola veľkosť fixná ), pričom veľkosť hracej plochy ostáva rovnaká. Tento prístup z hľadiska hrateľnosti znemožňuje hru na väčšej ploche ( políčka by boli príliš malé ), a preto som obmedzil maximálnu veľkosť hracej plochy na 20x20, čo mi pripadá adekvátne. Hracia plocha má vždy rozmery štvorca - užívateľ teda upravuje jedno číslo.

Užívateľ má tiež možnosť voľby počtu výherných políček. Táto hodnota, ale spravidla závisí na veľkosti plochy - nemôže byť väčšia ako jej rozmer a zároveň musí byť aspoň 3. Ak užívateľ v nastaveniach poruší niektorú hranicu ( v ktoromkoľvek nastavení ), je na to upozornený a hodnota sa zmení na najbližšiu možnosť, napr. keď by zadal 5 pre počet výherných políček, pričom je rozmer plochy 4, je upozornený vyskakovacím oknom a hodnota je zmenená späť na 4. Predvolené nastavenia sú 3x3.

2. V pôvodnej verzii som **hru jedného hráča** implementoval zväčša vo FE, kedy sa volala metóda *makeMove()* ako pri ťahu hráča, tak pri umelom ťahu "počítača", kedy boli vo FE náhodne generované súradnice. Po kontrolnej prezentácii sme do BE pridali funkciu *makeCpuMove()*, ktorá je zodpovedná za náhodný ťah a výrazne zjednodušila môj FE kód. Táto nová funkcia môže byť volaná hneď z *makeMove()*, ale to by znamenalo, že keď užívateľ spraví ťah, ihneď spraví ťah aj počítač a užívateľ by bol znova na rade ( zaplnia sa hneď dve rôzne políčka ). V mojej implementácii som chcel docieľiť u užívateľa pocit, že reálne hrá proti niekomu tak, že na ťah súpera malú chvíľu čaká. Preto v mojej implementácii program nevolá funkciu *makeCpuMove()* vo funkcii *makeMove()*, ale pomocou konštrukcii, ktoré ponúka React, malú chvíľu čaká a následne priamo volá *makeCpuMove()*. Takto to viac vyzerá, že sa hráči striedajú. V opačnom prípade by to vyzeralo, že je stále na ťahu užívateľ.

xpernid00:

### **Spracovanie pripomienok z kontrolnej prezentácie:**

Na kontrolnej prezentácii sme dostali feedback, že by sa hodilo rozšíriť naše riešenie, konkrétne manipuláciu s dátami a pridať viac reaktívnych prvkov.

Pridal som viaceré nastavenia hry, ktoré manipulujú s dátami - možnosť hry proti počítaču a úpravu veľkosti hracej plochy a počtu symbolov pri sebe potrebných na výhru.

GUI používa interaktívne prvky ktoré okamžite informujú používateľa a momentálnom stave hry a zmenách ktoré vykonal. Ide o možnosť zmeniť, ktorý hráč je na ťahu a zmena farieb celej hry. Toto je implementované pomocou kartičiek z vizuálnim indikátorom s farbami tém.

Hracia plocha je implementovaná ako dynamický *grid-layout* natívny z CSS. Pri väčších hracích poliach sa dynamicky mení veľkosť políčok a hrúbka stien, čo zabezpečuje čitateľnosť pre hráča. Pridal som taktiež možnosť upraviť počet symbolov potrebných na výhru, ktorého maximum a minimum sa odvíja podľa veľkosti plochy. Hráč si môže toto nastavenie zmeniť, zmena je implementovaná pomocou tlačítok. Toto rozhodnutie som urobil aby bol používateľ navedený na menšie hracie plochy, pri väčších plochách prichádza ku stratám kvality hry. Alternatívne môže vložiť číslo manuálne.

Ďalej som implementoval možnosť nastavenia herného módu proti hráčovy alebo proti AI, toto zabezpečuje slider v nastaveniach.

Ďalšou zmenou manipulácie je implementácia, výherca predchádzajúceho kola začína nasledujúce kolo prvý. V móde proti AI ide hráč vždy prvý.

Hra je základne nastavená na pole veľkosti 3x3 hráča proti hráčovy, prvý začína 'X'.

## **Výsledná implementace**

Stručný popis source kódu z pohľadu komponent.

xmokrup00:

Komponenty v Angular:

1. square-table - plocha s tlačítky, ukazateľom aktuálneho hráča, score a nastavení:  
tento komponent slouží jako hlavní komponent pro celou aplikaci
  - a. square - každé tlačítko je namapované na seznam a pokud se v seznamu objeví hodnota, tak ji vykreslí
  - b. nastavení - má slidery, pro nastavení parametrů hry, toggle historie, toggle CPU oponenta, tlačítko pro reset score
2. historie-table - ukazuje historii odehraných tahů:  
je provolána funkce pro vytvoření historie po každém tahu
  - a. History element - obdobně jako u tlačítka, každý element je namapovaný na pole seznamu, a pokud se detekuje, že není prázdné, tak se vykreslí

xmesikj00:

Komponenty v React:

1. výber módu - na začiatku je užívateľ vyzvaný na výber módu buď jedného alebo dvoch hráčov, po ktorom je presmerovaný na hlavnú obrazovku
2. hlavná obrazovka - obsahuje ikonu zvoleného módu; hernú plochu, kde hráči robia ťahy; vysúvacie hlavné menu

3. hlavné menu - obsahuje možnosti, ktorými vie užívateľ:
  - a. reštartovať hru
  - b. skončiť hru a vrátiť sa k výberu módu
  - c. zmeniť nastavenia hry
  - d. zobrazíť nápovedu

xpernid00:

Komponenty vo Vue:

1. Hlavný komponent ktorý implementuje hru: *PiskvorkyGame*  
Inicializuje hru a integruje všetky komponenty
2. *gameBoard* zobrazuje hraciu plochu
3. 4 dialógové komponenty implementujú nastavenia hry
4. *scoreCard* implementuje ukazovateľ skóre a momentálneho hráča na ťahu
5. *themeStore* obsahuje *pinia store* implementáciu na témy farieb
6. Všetka logika sa nachádza v API súbore, komponenty obsahujú iba logiku potrebnú pre UI

## Testování s uživatelem

xmokrup00:

Uživatel: 22 let, silnoproud, muž, hraje hry na počítači

Průběh testu: Sešli jsme se osobně, kdy jsem mu půjčil můj YOGA laptop.

Jednou jsem ho nechal hrát na touchpadu, a podruhé jsem mu laptop dal jako tablet.

- Při scrollování přes historii vždycky hledal nastavení (původně bylo hned pod hrací plochou) => dal vždy důležitým prvkem UI absolutní hodnotu.
- Chvilí mu trvalo než pochopí jak se ovládá pohyb po hrací ploše na touchpadu, ale když jsem mu dal laptop v tablet formě, hnedka to pochopil.
- Divil se, že mu při otevření aplikace neotevřel prompt pro nastavení inicializačních parametrů hry
- Oddalování mu připadlo trochu "divné"
- Některé labely se mu špatně četly (měl jsem nastavenou moc velkou průhlednost)

Klíčové výsledky:

- hledá důležité tlačítka (reorganizoval jsem do nastavení a pozici ukotvil ke kraji obrazovky)
- Moc velké transparentnost některých elementů (snížil jsem průhlednost)
- úvodní prompt pro nastavení hry. Nechci přidávat další krok před hraním nejčastěji krátké hry (defaultní velikost je 4x4), pokud by někdo chtěl hrát delší hru na větší ploše, tak ho nezabije klik navíc do nastavení.

xpernid00:

### **Testovanie s používateľom:**

Testovací používatelia: Skupina študentov 4. a 8. ročného gymnázia, rôzne vekové kategórie, pohlavia a zručnosti pri počítači.

Priebeh testu: Každý používateľ dostal notebook z bežiacov aplikáciou, v prípade hry viacerých hráčov hrali študenti proti sebe.

- Prvotné ovládanie bolo bezproblémové, používatelia nevyjadrili nijaké sťažnosti ani problémy
- Pri zobrazení nastavení niektorí používatelia nezistili intuitívne ktoré nastavenie robí čo
- Hra proti AI bola viac menej u všetkých používateľoch ohlásená ako ľahká, keďže AI hraje náhodne, je potrebné AI zlepšiť a zvýšiť tak obtiažnosť hry
- Pri väčších plochách ohlásili používatelia zhoršenú čitateľnosť hry
- Pri výbere veľkosti plochy je neustále klikanie tlačítka + otravné, podobne pri zmene farebných tém
- Základné nastavenie hry bolo ocenené pre rýchli začiatok

### **Výsledky:**

Problém nadbytočného klikania som vyriešil implementáciou možnosti priamo vložiť číselnú hodnotu veľkosti plochy, pri zmene tém som odstránil potvrdzovacie tlačidlo, farba je zmenená okamžite.

Problém čitateľnosti väčšej plochy som čiastočne vyriešil zúžením stien a zmenšením štvorčekov.

Pre používateľa je najdôležitejšia prehľadnosť a efektivita, chce bez problémov vidieť hraciu plochu a ľahko s ňou manipulovať.

xmesikj00:

Užívateľ je študent na vysokej škole na fakulte technického zamerania v treťom ročníku bakalárskeho štúdia, muž, 21 rokov. Vo voľnom čase hráva moderné počítačové/konzolové hry.

Užívateľovi som spustil moju verziu hry a povedal som mu nech postupuje ako pri hraní nejakej novej hry a nech "rozmyšľa nahlas". Počas testovania sa preklíkal skoro všetkými možnosťami, vyskúšal hru pri rôznych nastaveniach, hru jedného aj dvoch hráčov (kedy hral za oboch), snažil sa nájsť nedostatky hlavne v nastaveniach.

### **○ Výsledok:**

- V nastaveniach bolo podľa užívateľa dobre viditeľné a zrozumiteľné upozornenie, ale zo svojich herných skúseností je zvyknutý na systém nastavení, kde po úprave hodnôt existuje tlačítko ' *Uložiť* ', ktoré aplikuje všetky zmeny a na následky ho upozorní až jeho stlačenie. Tento systém preferuje užívateľ viac. **Poučenie:** zmeny, ktoré užívateľ vykoná v nastaveniach neaplikovať na dáta v BE ihneď, ale niekam ich uložiť a vytvoriť tlačítko, ktoré aplikuje všetky zmeny naraz. Ak by užívateľ nejakú zmenu vykonal, tak by po stlačení bol upozornený na jej vplyv a bol by vyzvaný, či chce zmeny skutočne aplikovať.

- Pri hre na väčšom poli je hra pre jedného hráča príliš ľahká, keďže je umelý ťah generovaný vždy náhodne. To má častokrát za následok z hľadiska stratégie nezmyselné ťahy súpera a hra je potom menšia výzva.

**Poučenie:** vymyslieť lepší systém na vytváranie umelých ťahov pri hre jedného hráča, čo otvára aj možnosť zavedenia systému rôznych obtiažností.