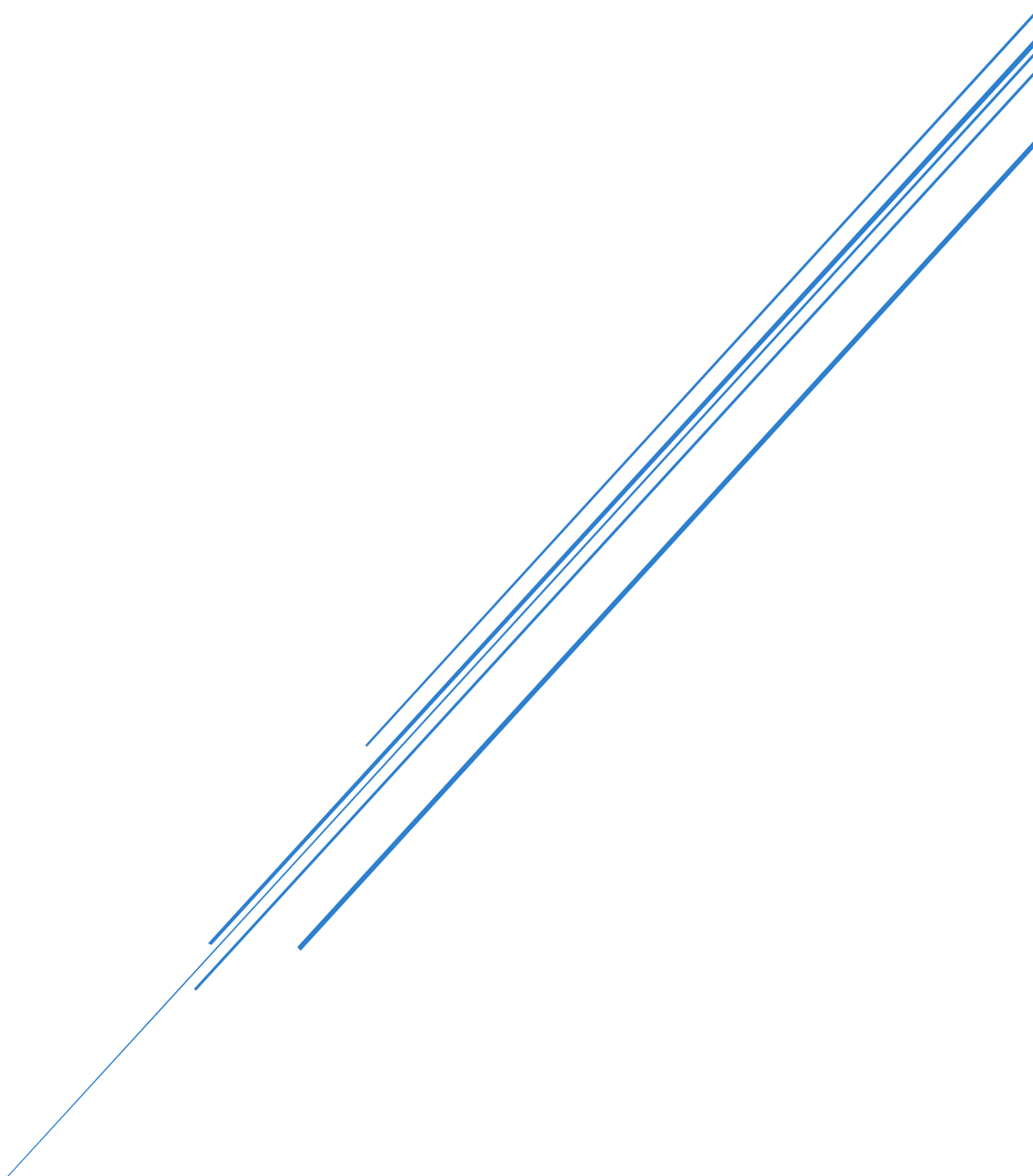


DOKUMENTACE ZÁPOČTOVÉHO PROGRAMU C#

POKROČILÝ + ZÁKLADNÍ C#



MATEMATICKO-FYZIKÁLNÍ FAKULTA UNIVERZITY KARLOVY
Jáchym Mraček

Obsah

Popis programu a tématu.....	
Popis jednotlivých her a popis řešení	
Dáma	
Piškvorky	
Prší.....	
Řešení síťování a více vláknové programování.....	
Síťování	
Více vláknové programování	
Vývojové prostředí a vývojové prostředí	
Vývojové prostředí	
Operační systém.....	
Obrazovky	
Zdroje.....	

Popis programu a tématu

Vytvořený zápočtový program je multiplayer platforma 2D deskových her s dvěma hráči. Konkrétně se jedná o dámu, piškvorky a prší.

Popis her

Dáma

Hra probíhá podle klasických pravidel dámy, kde si uživatel může nastavit, zda chce zvolit čas nebo ne.

Hráč vyhrává, pokud sebere všechny soupeřovi figurky nebo soupeřovi vyprší čas.

Pohyb figurek

Figurky se mohou pohybovat, pouze pokud příslušný hráč je na tahu, tedy tlačítko určující, zda je uživatel na tahu musí svítit světle zeleně.

Pokud je uživatel na tahu a chce pohybovat figurkou, pak musí příslušné tlačítko stisknout. V případě stisknutí se uživateli zobrazí zelená tlačítka, na které může figurka vstoupit. Následujícím stisknutím se figurka pohne a odešle na server, pohyb příslušné figurky. Potom může hrát soupeř.

Figurka, které může vyhodit protihráčovu figurku má přednost před figurkou, která nemůže vyhodit.

Příslušná hra obsahuje dva typy figurek – pěšec a dáma. Obe figurky se pohybují diagonálně, kde pěšec pouze o jedno políčko ku předu a dáma se může pohybovat vpřed i do zpět o několik políček, kde nesmí vystoupit z desky nebo ignorovat svojí nebo protihráčovu figurku.

Piškvorky

Klasická hra, kde se hráči střídají a první z hráčů, který zvládne položit 5 svých kamenů do sloupce, řádku nebo diagonály vyhrává.

Program umožňuje hráči, který nastavuje hru zvolit, zda chce klasickou hru, kde se střídají hráči po jednom nebo turnajovou hru, kde první hráč rozehraje dva červené a jeden modrý symbol. Následně protihráč si vybírá barvu, se kterou chce hrát.

Prší

Hraje se podle standartních prší pravidel, kde v naší hře svršek je označen 12Z, král 13 a eso 14X.

První hráč, který odehraje všechny karty vyhrává.

Hráč hru zahajuje stisknutím tlačítka táhnout nebo tlačítka karty.

Soubory s kody

Práce je rozdělená do několik souborů:

- DamaHelpClasses.cs
- EndGameClasses.cs
- Form1.cs
- Form1.Designer.cs
- GameClasses.cs
- PageHelpClasses.cs
- Pages.cs
- PiskvorkyhelpClasses.cs
- prsiClasses.cs
- Signal.cs
- UserInformation.cs

Pages.cs

Soubor, který obsahuje potomkovou třídu pages, která obsahuje **List<Button> buttons**, **List<Label> labels**, které slouží k načtení tlačítek a textů na příslušné stránce, kde page se načte pomocí metody **SeePage(int pageIndex)**, kde argument je index v listu List<Page> pages.

Soubor obsahuje třídu Page0, což je stránka, kde obsahuje pouze textové pole pro zadání jména, které lze zadat pouze v případě odlišného zvolení než někdo jiný připojený na server.

Následně Page1 je stránka obsahující dvě tlačítka VSTUP A ODEJÍT, kde stisknutím VSTUP se zobrazí stránka Page2 s listem založených her a tlačítka ZPĚT na stránku Page1.

V případě stisknutí herního tlačítka ZALOŽIT se zobrazí Page3 s tlačítky DÁMA,PIŠKVORKY,PRŠÍ.V případě stisknutí se zobrazí nastavení hry.

Pokud Uživatel stiskne Dáma, pak se zobrazí Page4 s možností volby hry – s časem nebo bez času, pokud uživatel si vybere příslušné pravidla dostane se na hrací stránku Page11 s hrací deskou (List tlačítek boardFigure), tlačítkem určující tah hráče **Button turnButton** a s tlačítky s textem zbývajícího času. **Button myTimeButton**, **Button OpponentTimeButton**.

V případě stisknutí PIŠKVORKY se dostaneme na Page5, která obsahuje tlačítka CLASSICAL,TURNÁJOVÁ HRA a ZPĚT. V případě stisknutí CLASSICAL

nebo TURNAJÉOVÁ HRA se uživatel dostane na stránku hřiště Page10, které obsahuje **Board board** hřiště a **Button turnButton** tlačítko určující tah hráče.

V případě stisknutí Prší se uživateli zobrazí stránka Page6, která obsahuje dvě tlačítka HRA S HRÁČEM a ZPĚT, kde stisknutím HRA S HRÁČEM se uživatel dostane na stránku Page7, která obsahuje položky **PrsiTable prsiTable** a **PrsiPlayer prsiPlayer**, kde se uživatel už může hru prší hrát.

Pokud hráč zahraje kartu svršek, pak se zobrazí Page8 s možností výběru barvy, tato třída obsahuje **public Blue blue;public Black black;public Red red;public Yellow yellow;** což jsou barvy jednotlivých karet.

Pokud libovolný hráč vyhraje, pak se dostane na stránku Page9 s tlačítky KONEC nebo OPAKOVAT, kde stisknutím KONEC se oba hráči dostanou na stránku Page1 a pokud oba uživatel stisknout tlačítko OPAKOVAT, pak si zahrajou stejnou hru se stejnými pravidly znova.

prsiClasses.cs

Soubor obsahuje třídu ColourCard která ve hře zajišťuje správnou interpretaci barev jednotlivých karet. Třída **obsahuje public Button button { get; set; }** a **bool choose = false;** jestli karta byla hraná. Třída obsahuje metodu **Symbol_Click(object sender, EventArgs e)**, tedy metodu která se zavolá po stisknutí tlačítka.

Dále soubor obsahuje třídy jednotlivých barev Blue,Red,Yellow a Black, kde tyto třídy dědí od ColourCard třídy.

Hra prší obsahuje tlačítko reprezentující tlačítko, které je v kodu popsánou třídu DeckButton dědící od UsesOfProject. Třída implementuje metodu **DeckButton_Click**, která se následně zavolá **na private void PickCard()** kterým si hráč líže kartu a nechává hrát protihráče.

Karta na stole je reprezentovaná třídou class CardTable, která implementuje **Button button, bool valid** – zda karta ještě platí například eso a **int takeCards** počet karet, kterých si protihráč musí lízat (mění se v případě zahrání esa nebo sedmičky).

Dále hra obsahuje prsiTable, která obsahuje CardTable a Button turnButton, DeckButton a list jednotlivých karet - **List<Card> cardsInDeck**

Dále máme tříduPrsiPLayer, která dědí od UsesOfProject, kde třída obsahuje **List<Card> cardsInHand**. Tedy list karet, které má hráč v ruce. Třída

obsahuje metody **void AddToHand a MatchCards()** – pro porovnání karet, zda stisknutelná karta je hratelná.

class GreenButtonSetPlayable zajišťuje, změnu tlačítka v případě odehraní karty.

Třída Card je implementovaná s feldem Button, jelikož se jedná o kartu s metodami **bool Playable(Button sender)** – jestli je karta hratelná **bool EsoPlayed()** - jestli se jedná o eso **bool SevenPlayed()** – jestli se nejedná o sedmičku, dále metodu **Delete(Button sender)** , která odebírá kartu se hry. Další důležitou metodou je asynchronní **async Task PlayCard(Button sender)**, která posílá signal, že byla odehraná karta na server, který následně pošle signal protihráči, že karta byla odehrána.

Dále kod obsahuje třídu Seven, která dědí od SpecialCard, jelikož se jedná o speciální kartu, kde odehráním protihráč buď bere dvě karty nebi hraje sedmičku a přičítá dvě karty, které si protihráč musí vybrat. Třídy esoCard a svrsekCard také dědí od SpecialCard.

Ostatní karty jsou popsány třídou NormalCard, která dědí od Card

Závěrečná třída, kterou soubor obsahuje je MovePrsi dědicí od UsesOfProject, která zajišťuje správně hraní jednotlivých karet.

PiskvorkyHelpClasses.cs

Soubor obsahuje třídu Board s feldem **List<Button> buttons** - tedy pole pro hraní piškvorek. Třída obsahuje asynchronní metodu **async Task StartTournamentThreePlay()** , kde po odehrání třech symbolů se pošle signal na server, který pošle signal protihráči, že se zahajující hráč odehrál tři symboly a může si vybírat barvu.

Implementace class CheckPiskvorkyWinner, kontroluje zda jeden z hráčů nedokázal dát 5 svých symbolů k sobě. Pokud ano, server dostane signál, že jeden z hráčů vyhrál a proto příslušné metody jsou asynchronní Task.

Poslední třídou je class TournamentPiskvorky, která obsahuje proměnné pro správný chod turnajové hry.

DamaHelpClasses.cs

Počáteční metoda **FigurePressed : ButtonPressed** obsahuje dvě virtuální metody public virtual void SetLeftGreenButton(int index, bool mustJump) { }, public virtual void SetRightGreenButton(int index, bool mustJump) { } které zobrazí hrací políčka, kam stisknutá figurka může jít.

public class GreenGameButtonShort : FigurePressed, třída, která zobrazí levé a pravé tlačítko, pokud vyhovují podmínkám. Jsou nato dvě metody override void

SetLeftGreenButton(int index, bool mustJump) a **override void SetRightGreenButton(int index, bool mustJump)**

GreenGameButtonLong : FigurePressed – Třída která zobrazí tlačítka na které může figurka vstoupit v případě vyhození figurky

public class EmptyGameButton : ButtonPressed - třída která vrátí příslou barvu tlačítka zpět.

public class PressGreenButtonGame : ButtonPressed – Třída, která reaguje na stisknutí zeleného tlačítka. Třída obsahuje metody **DeleteRightFigure(int indexFrom, int indexTo)** – Odebere figurku z plochy (i protihráčovi, jelikož hráč pošle signal na server). Dále obsahuje async Task Jump(int indexFrom, int indexTo, object button), která zajišťuje správný skok figurky a kontrolou, zda nemůže figurka dále vyhazovat, také pokud figurka dojde na protihráčovu stranu, pěšec se stává dámou. (Metoda NextJump kontrolu potenciální skoky – vyhození).

public class QueenGreen : ButtonPressed – Třída, která naopak obarví tlačítka zeleně, pokud se dáma může pohybovat.

public class DeleteQueenJump : ButtonPressed – Stejně jako předtím odebírá figurku z pole. Obsahuje metody asynchronní metody, které posílají signal na server.

public class BoardFigure : UsesOfProject – Opět herní deska pro dámu.

void MustPlay() – Metoda, která zajišťuje, že může skákat pouze figurky, které ohou vyhodit protihráčovu figurku.

async Task EndOpponent() – konec hry.

EndGame.cs

public class EndGameButton : UsesOfProject – Pokud uživatel stiskne tlačítko endGame, pak se hráč dostane na Page1.

public class EndUserGame : UsesOfProject – Pokud hráč předčasně ukončí hru, pak se také musí oberat z Page1 i hrací lajna pro připojení jiného hráče.

GameClasses.cs

public class StartEndGame – Pokud uživatel založí hru třída zavolá metodu a pošle signal na server, že hra byla založena **await project.signalR.Connection.InvokeAsync("SetGame", typeGame, project.userInfo.name);**

Třída obsahuje metodu **async Task StartGame(string typeGame)**, která se zavolá pokud hráč založí hru a proto vznikají třídy, které od této třídi dědí,

public class PrsiGame : StartEndGame **public class DamaGame : StartEndGame**

public class PiskvorkyGame : StartEndGame

PageHelpClasses.cs

public class GreenButton : UsesOfProject – Pokud uživatel stiskne zelené tlačítko, bude poslán signal serveru, že hráč s epřiponil a bylo stisknuté zelené tlačítko.

public class LineOfTheGame – Třída obsahující **public GreenButton greenButtons { get; set; }** **public List<Label> typeGamelabels = new List<Label>();** po načtení všichni uživatelé uvidí jaké hry si mohou zahrát.

public class AddButtons : UsesOfProject – Obsahuje metody, které zobrazí modrou a červenou barvu

public class ConnectedToGame : UsesOfProject – Pokud se hráč připojí metoda obsahuje metody, která pošle serveru signal, že se připojil hráč a protihráč může hrát. (Zobrazí se mu zelené tlačítko)

Form1.cs

Obsahuje Field:

public UserInformation userInformation = new UserInformation(); - informace o hráči jméno atd.

public SignalR signalR = new SignalR(); - Pro posílání signálů na server.

public List<Page> pages = new List<Page>(); - pages list jednotlivých stránek

public SignalCommands signalCommands; - Třída obsahující příkazy ppro zaslání na signálu na server.

public System.Windows.Forms.Timer timer1 = new System.Windows.Forms.Timer(); - Časovač pro dámu.

public int timeLeft = 0;

public int plusMove = 0;

public bool classical = true;

➤ Proměnné pro správný chod dámy (čas)

public TournamentPiskvorky tournamentPiskvorky = new TournamentPiskvorky(); - Jendá se o turnajovou hru – Piskvorky

Následně soubor obsahuje sadu příkazů, které slouží k obdržení signalu od Serveru.

signalR.Connection.StartAsync().ContinueWith(task =>{ });

Signal.cs

public class SignalR – Uživatel se připojí na server.

public class SignalCommands : UsesOfProject – Příkazy, které jsou poslány od serveru, který následně uživatel vykoná, typu – změnit barvu tlačítka atd.

UsesOfProject.cs

public class UsesOfProject – Několik tříd dědí právě od této třídy, jelikož se jedná o referenci z Form.

public class UserInformation – Informace o uživateli a nastavení příslušné hry.

Program.cs (Server)

Obsahuje fieldy: public static ConcurrentDictionary<string, string> tables = new ConcurrentDictionary<string, string>(); Kde se ukládají jednotlivé jména hráčů se kterými hrajou.

public static List<string> nameServer = new List<string>(); - obsahuje jména, které byly už zadány.

public static readonly SemaphoreSlim _semaphore = new SemaphoreSlim(1, 1); - proto aby nedošlo k chybě, kde dva hráči najednou pošlou stejné jméno.

public WebApplicationBuilder applicationBuilder { get; set; }

public WebApplication application { get; set; }

Poslední dvě slouží k webovému rozhraní serveru.

Závěrem program obsahuje sadu asynchronních Task metod, které á třída

public class ServerRequests : Hub, kde příkazy dostává právě od připojených uživatelů.

{

public async Task Pass(string namePlayed)

{

await Server._semaphore.WaitAsync();

try { new Pass(namePlayed, Context, Clients);}

finally {Server._semaphore.Release();}

}

Ukázka, kde je využit Semaphore aby nedošlo k chybě, díky současnému požadavku hráčů.

ClassesForTask.cs

public class PLayerInformation – Třída která je děděna od ostatních tříd, které vykonávají příslušný úkol obsahuje jméno uživatele (zakladatele) a hráče připojeného.

Dále obsahuje sadu tříd, které zpracovávají příslušný úkol

Síťování a více vláknové programování

Naší multiplayer hru vytvoříme pomocí klient-server metody, kde použijeme knihovnu Microsoft.AspNetCore.SignalR; kde se několik klientů může připojit na lokální webový server.

Nejdříve se uživatelé připojí na server a potom posílají serveru sadu příkazů, kde se vytvářejí několik vláken, jak na serverovské, klientské i klientské protihráčovy straně, jelikož

Klient > server > minimálně jednomu hráči.

Vláknování je řešené asynchronně a proto některé metody obsahují this.InvokeRequired, aby nedošlo k chybnému přepsání proměnné, které nebylo vytvořené z hlavního vlákna.

V programu na serverovské straně je využit jeden semaphore.

Vývojové prostředí

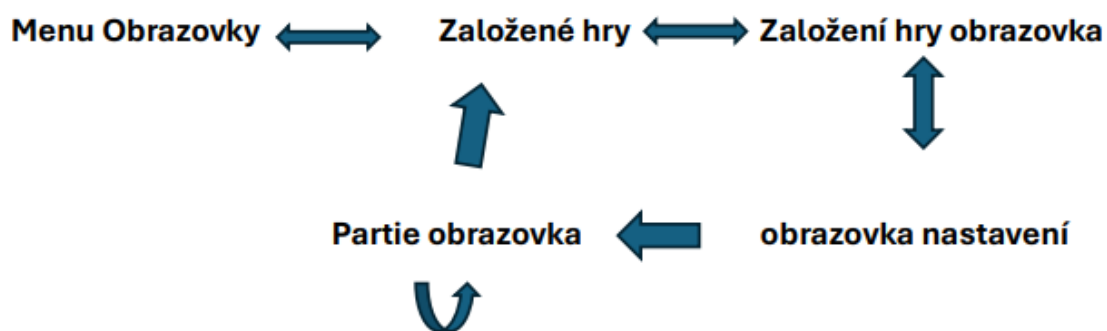
Program je napsán v jazyce C# ve Visual Studiu, kde klient je naprogramován ve windowsForms a server ve webové aplikaci ASP.NET Core.

Operační systém "

Program je napsán pro operační systém Windows.

Schéma obrazovek

Schéma Obrazovek



Zdroje:

https://cz.pinterest.com/pin/283726845249720328/?amp_client_id=CLIENT_ID%28%29&mweb_unauth_id=%7B%7Bdefault.session%7D%7D&simplified=true

<https://www.wallpaperflare.com/angry-birds-video-games-sky-nature-no-people-green-color-wallpaper-qdjej>

<https://angrybirds.fandom.com/f/p/4400000000000083586>

<https://www.shutterstock.com/cs/image-vector/crown-blue-vector-icon-eps-10-2170238625>

<https://vyhledavac.mujup.cz/132047>

<https://angrybirds.fandom.com/wiki/Bomb>

<https://cz.pinterest.com/pin/478789004105312778/>