

# DOKUMENTÁCIA ZÁPOČTOVÉHO PROGRAMU GAME31

AREK ANTONIEWICZ

## UŽIVATEĽSKÁ DOKUMENTÁCIA

### PRAVIDLÁ HRY

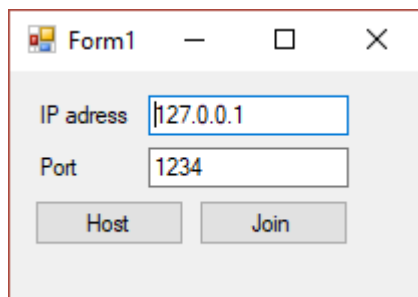
Každý hráč na začiatku ovláda určitý počet postáv. Počas svojho ťahu deklaruje, ako sa postavy budú hýbať, kam zaútočia a v akom poradí. To isté spraví súper a následne sa vyhodnotí a zobrazí čo sa v danom ťahu stalo. Cieľom hry je zabiť všetky súperove postavy (znížiť im život na 0).

Deklarovanie ťahu – Každá postava má určitý počet krokov, ktoré môže v každom ťahu spraviť. Postava sa najskôr hýbe, potom útočí. Útočiť môže iba na miesta, na ktoré má dosah. V prípade, že došlo ku kolízii (pozri nižšie) postava zaútočí na relatívnu pozíciu oproti miestu kam chcela ísť.

Poradie - Postavy sa hýbu v deklarovanom poradí, súčasne s protivníkovou postavou(prvá s prvou, druhá s druhou atď.)

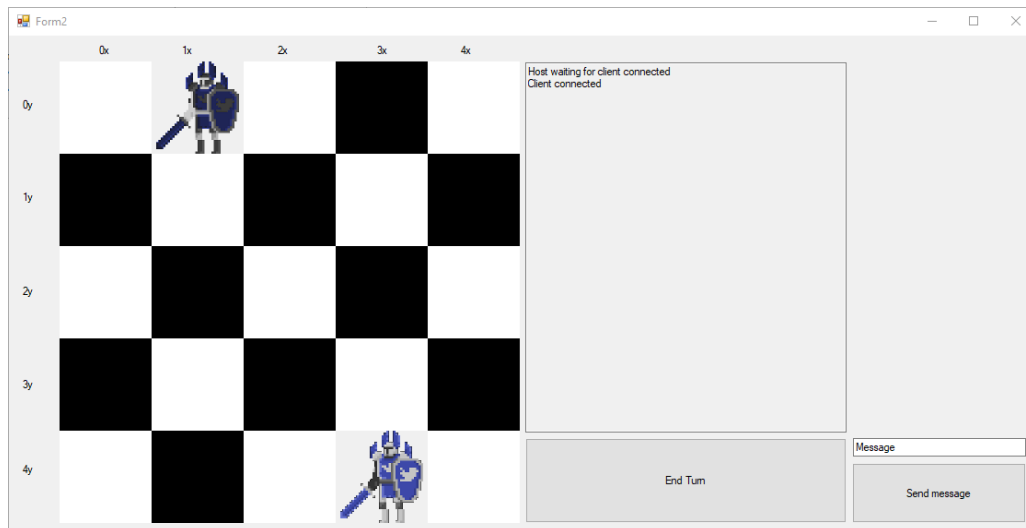
Kolízia – Ak sa postava chce hýbať na miesto, kde už sa nachádza iná postava, postavy sa zrazia. Ak sú protivníci tak si dajú collision damage. Postava sa po kolízii prestane hýbať. Ak sa súčasne dve postavy chcú pohnúť na rovnaké miesto presunie sa tam náhodne jedna postava a následne sa zrazia. Zrazia sa aj v prípade, že súčasne sa jedna chce pohnúť na miesto kde stojí druhá a druhá sa chce pohnúť preč.

### OVLÁDANIE



The screenshot shows a standard Windows application window with a title bar containing the text 'Form1' and standard minimize, maximize, and close buttons. The window's content area has a light gray background. It features two text input fields stacked vertically. The first field is labeled 'IP adress' and contains the text '127.0.0.1'. The second field is labeled 'Port' and contains the text '1234'. Below these input fields, there are two rectangular buttons side-by-side. The left button is labeled 'Host' and the right button is labeled 'Join'. Both buttons have a light gray background and a thin border.

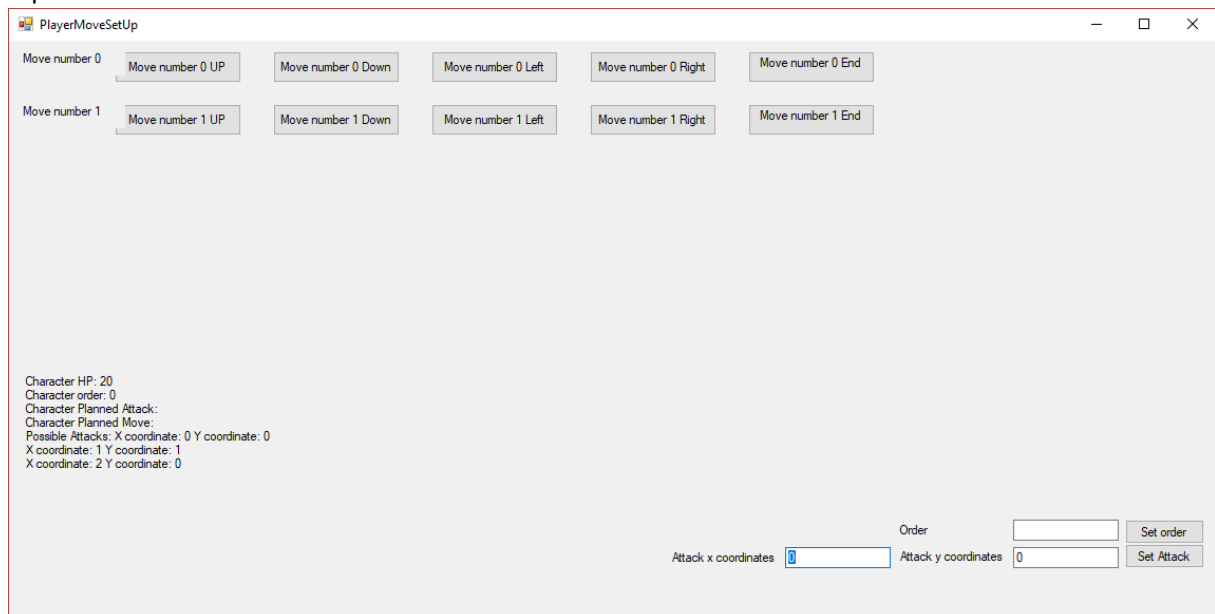
Hru hrajú dvaja hráči. Spoločne sa do hry pripoja tak, že jeden hru vytvorí a druhý sa pripojí.



End Turn – koniec kola, počká sa, kým skončí aj súper, následne sa zobrazia výsledky kola a nasleduje ďalšie kolo

Send Message – pošle správu súperovi s obsahom, ktorý je v textBoxe nad týmto tlačítkom

Pri kliknutí na svoju postavu sa otvorí okno na zadávanie ťahu danej postavy s informáciami o postave.



Pomocou horných tlačítkov sa nastaví pohyb postavy. Následne sa poradie a kam zaťochí nastaví v dolnej časti obrazovky.

## PROGRAMÁTORSKÁ DOKUMENTÁCIA

Hra je rozdelená na GUI a logiku hry. GUI sa stará o zobrazovanie a zadávanie ťahu, logika hry o komunikáciu a počítanie kola.

### LOGIKA HRY

#### GAMEMANAGER

Informácie o hre drží GameManager. Z GUI sa zavolá konštruktor a následne podľa toho čo sa deje na GUI alebo v komunikácii sa volajú metódy GameManager-a.

#### MAP

Dvojmerné pole GameSquare, vyplnené Empty square alebo potomkom Character. Metódy na prácu s mapou

#### TURN LOGIC

Metóda resolve turn, dostane deklaráciu prvého a druhého hráča a mapu, Vypočíta zmeny na mape a vráti List Action (to čo sa má zobraziť na GUI)

#### GAMECOMMUNICATION

Stará sa o komunikáciu medzi hráčmi. Obsahuje Abstraktné metódy ThisPlayerTurn a SendMessage ktoré sa implementujú podľa toho či je GameClient alebo GameHost. Typ správy sa určí podľa prvého bytu. /\*

```
* 0 - sending TurnResults
* 1 - sending Map
* 2 - sending DeclarePlayerTurn
* 3 - sending message
*/
```

#### SERIALIZER

Serializuje a deserializuje classy, ktoré sú potrebné aby sa posielali.

### GUI

Skladá sa z troch formulárov. Prvý slúži na pripojenie druhý zobrazuje hru a tretí na deklarovanie ťahu postavy.

#### FORM1

Pripojenie sa k súperovi

#### FORM2

Vytvorí gamemanagera podľa toho ako ho zavolá form1. Zobrazuje resulty, správy, pri kliknutí na postavu sa vytvorí tretí formulár

#### PLAYERMOVESETUPFORM

Informácie o jednej postave, zadávanie ťahu danej postavy.

### POUZIVANIE KNIZNICE GAMELOGIC

Používa sa pre ovládanie hry class GameManager

Vytvorí sa

```
public GameManager(List<Character> chars, int size, Action<TurnResults> sr, Action<string> sm,
Action<List<Character>> actualizeCharacters, bool host, string ipAdress, string port)
```

List<Character> chars - vytvorené počiatočné charactery  
int size – veľkosť mapy

Action<TurnResults> sr – metoda ktora sa bude volat na zobrazenie tahu  
Action<string> sm – metoda pomocou ktorej sa budu zobrazovat spravy  
Action<List<Character>> actualizeCharacters – metoda pomocou ktorej sa aktualizuju  
charactery na gui, vrati list aktualnych characterov a gui si ich moze ulozit k sebe ak ich potrebuje  
bool host, string ipAdress, string port – na vytvorenie komunikacie medzi hracmi

Metody volane z GUI:

DeclarePlayerTurn(PlayerTurn pt)

state of game v gamemanagery musi byt PlayerDeclaringTurn

zmeni sa state of game na WaitingForOpponent

sluzi na zadavanie hracovho tahu

DoneShowingResults()

state of game v gamemanagery musi byt ShowingResults

zmeni sa state of game na PlayerDeclaringTurn

oznamy gamemanagerovi ze sa uz dozobrazoval koniec tahu

EndOfGame()

vola sa pre ukoncenie hry