Gymnázium, Praha 6, Arabská 14
Obor programování

Ročníkový projekt

Strategická tahová videohra

Vojtěch Nejedlý

srpen 2022

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne 25.srpna 2022 Vojtěch Nejedlý

Anotace

Cílem mé práce bylo vytvořit jednodušší bojový systém se strategickými prvky pro dva hráče u jednoho počítače za pomoci Javy. V dokumentaci práce najdete postup vytváření hry a popis důležitých prvků a mechanik.

Obsah

1. Úvod

Oproti práci z minulého školního roku, kde jsem pracoval s jednorozměrným polem, jsem se rozhodl, že si chci zkusit vytvořit něco za pomocí dvourozměrného pole. Také jsem se rozhodl přidat ručně kreslené obrázky.

1.1 Popis hry

Hra vykreslí herní pole o rozměrech 12 x 11. Hraje se ale pouze na poli 12 x 10, kde při startu první dva sloupce používá modrý hráč a poslední dva sloupce používá hráč červený. Hráči mohou vytvářet vlastní scénáře bitvy a mohou smazat obě počáteční předem nastavené armády. Hra má dva typy jednotek *lukostřelce* a *rytíře*. Rytíř je jednotka která umí bojovat pouze na blízkou vzdálenost a ujde maximálně 5 políček. Lukostřelci mohou útočit na dálku, ale vydrží méně než rytíři a také ujdou o 2/5 menší pohyb. Hráč, který přijde o všechny jednotky prohrává a vypíše se nápis, který hráč vyhrál. Hra se také zamrazí a jediná možnost je odejít zpět do menu, jelikož bude na tahu hráč, který už nemá žádnou jednotku, s kterou by mohl hýbat.

1.2 Použité technologie

Projekt jsem vytvořil v NetBeans 8.2 a později přešel na IntelliJ IDEA Community Edition od společnosti JetBrains. Obě vývojové prostředí podporují Windows, macOS i Linux.

Obrázky použité ve hře jsem prvně dělal v Malování, následně jsem přešel ke GIMPu verze 2.10.24, což vylepšilo kvalitu a i zjednodušilo práci.

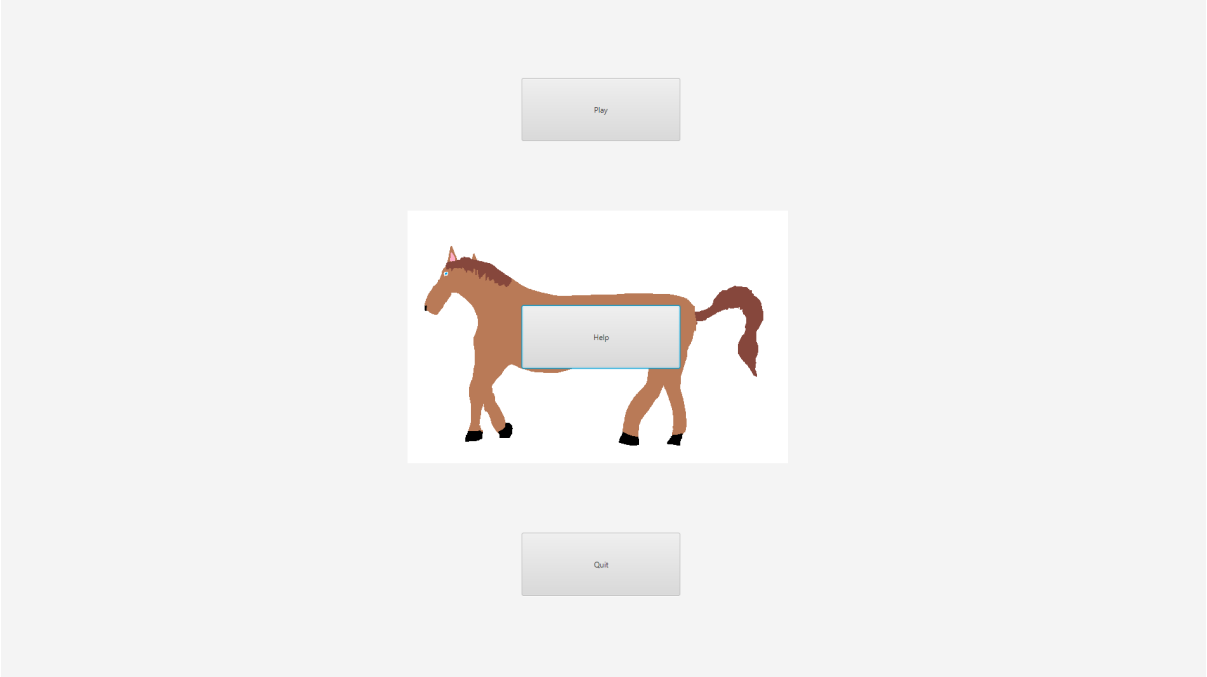
Program jsem dělal v Javě 8 za pomoci JavaFx Scene Builderu, který pár věcí komplikoval, ale najít řešení nebylo tak složité.

2. Grafické Rozhraní

2.1 Hlavní menu

Scéna, která se zobrazí po zapnutí programu. Hráč může vybrat jednu z možností: *Play* - otevře hrací pole, *Help* - otevře text s nápovědou, *Quit* - vypne hru. Pro ukázku kvality obrázku z malování jsem zde nechal obrázek koně.

Tlačítka Play a Help přepnout scénu na jiný GridPane, díky metodě která současný odebere a nastaví jiný GridPane na AnchorPanu, který je neměnný [1].

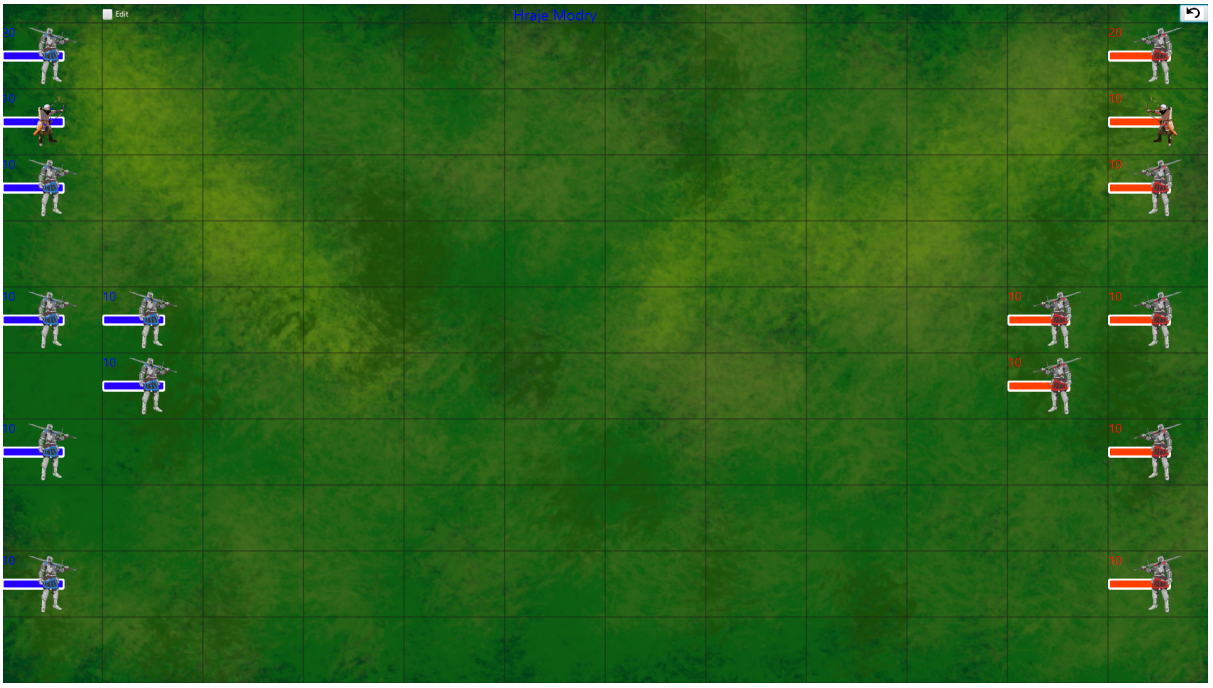


Obr. 1

2.2 Herní pole

Herní pole se skládá z pole na kterém jsou jednotky, dále 30 pixelové lišty v horní části obrazovky a tlačítka *zpět* v pravém horním rohu.

Uprostřed nahoře je indikátor toho, který hráč je právě na řadě. V prvním řádku druhého sloupce je důležité zaškrtávací políčko, které umožňuje mazat, přetvářet a vytvářet nové jednotky.



Obr. 2

2.3 Herní postavy

Postavičky zrovna moc velkou kvalitu nemají, pouze při pohledu z větší vzdálenosti nabírají dojmu kvalitních postav. Každá skupina má svůj počet a současné životy, které indikuje ProgressBar. Postava lukostřelce je vidět navrchu a postava reprezentující rytíře vespod.

Obr. 3

3. Logika aplikace

3.1 Figures package

Obsahuje tři třídy, přičemž dvě dědí z třídy Figure. Další dvě třídy se jmenují Knight a Archer. Třída Figure obsahuje důležité proměnné jako *double* maximální životy, *double* brnění, *double* poškození, *int* vzdálenost pohybu, *int* počet, který udává počet jednotek ve skupině, a *boolean* barva, která když je *true* znamená, že se jedná o jednotku červeného hráče. V opačném případě, tedy *false*, se jedná o jednotku hráče modrého. Třída má také proměnnou *Image* z knihovny JavaFX. Třída má gettery na všechny své proměnné a na pár proměnných i settery, jako třeba současné životy nebo počet jednotek. V parametrech konstruktoru se nastaví barva a počet jednotky. Konstruktor pak inicializuje pouze některé parametry.

Třída Knight dědí z třídy Figure a obsahuje pouze konstruktor, který nastavuje proměnné, jaké konstruktor třídy Figure nenastavuje. Již nastavené proměnné z konstruktoru třídy Figure dědí. Navíc, podle toho, je-li *boolean* color pravda nebo lež nastaví, který obrázek se má do objektu při vytvoření nahrát.

Třída Archer dědí ze třídy Knight a také obsahuje pouze konstruktor. Od ostatních dvou tříd se mění tím, že nastavuje poškození na dálku z -1 na platnou hodnotu. Také narozdíl od třídy Knight má menší hodnotu brnění a hodnotu vzdálenosti pohybu.

3.2 Core package

Tento package obsahuje dvě třídy, které se zabývají logickým zastoupením pole. Třída Square má proměnnou xPos a yPos typu *int*, které mají určovat, o které políčko se jedná, pak má třída také proměnnou *Figure*, aby si pamatovala, která postava se zde zrovna nachází. V konstruktoru se tyto proměnné nastaví. Dále třída obsahuje setter na postavu a také getter pro postavu, kterou objekt obsahuje. Též má metodu isEmpty, která vrací *true* pokud se currentFigure rovná null.

Třída Field toho obsahuje o něco víc. Tato třída má pevně nastavené rozměry pole, jakožto WIDTH a HEIGHT. Jedná se o statické pevně dané čísla 12 a 10, což jsou rozměry herního pole. Třída má jedinou proměnnou a to dvourozměrné pole Squareů, které nese název field. Konstruktor této třídy je bez parametrů a obsahuje 2 do sebe vnořené *for-loopy*, které vytvoří pole o rozměrech 12 x 10, tak že pokaždé vytvoří nový objekt typu Square a nastaví současné I, J a null do parametrů nového políčka.
Tato třída dále obsahuje veškeré metody, které mohou hráči zahrát, návratová hodnota těchto metod je boolean. Každá metoda nejdříve ověří zda je pohyb, který chce hráč provést legální tedy, jestli splňuje vše co má, když ne vrací *false*.
Mějme metodu move, která má za úkol hýbat s hráčem po mapě. Metoda potřebuje pro zavolání nastavit do parametrů výchozí pozici postavy a pozici kam se chce pohnout. Metoda prvně ověří, zda se vůbec na vybraném políčku, z kterého chceme provést pohyb postava nachází. Potom ověří, že políčko kam jdeme, není to, ze kterého jdeme. Následně si uloží vybranou postavu a zjistí, zda je vzdálenost mezi body menší nebo rovna vzdálenosti, jakou může ujít postava. To je zjištěno pomocí výpočtu: $\sqrt{((x2-x1) * (y2-y1))}$, kdy: x1 a y1 jsou počáteční souřadnice a x2 a y2 jsou cílové souřadnice. Teprve poté se zjišťuje zda na cílových souřadnicích není žádná jednotka, jelikož pokud se zde nachází jednotka stejné barvy a stejného typu, tak se jednotky spojí tak, že se počet jednotky co se přesouvá přičte k jednotce v cílovém místě a původní jednotka zaniká a metoda move vrací *true*. Pokud nenastane žádný z případu výše, metoda move vrací *true*. Třída Field ještě obsahuje metodu attack a rangedAttack, které fungují podobě. Ještě je zde metoda mergeArmy, která se volá pouze metodě move.
Field obsahuje ještě metodu assignPos, které se zadají souřadnice x a y a nová postava, kterou chceme do pole přidat. Tato metoda zavolá setter ze třídy Square a nastaví novou postavu do příslušného políčka. Se zaškrtávacím tlačítkem edit jsem přidal ještě metodu deleteFig, která naopak smaže postavu z daného políčka.

3.3 Controller hry

Jak se mezi sebou jednotlivé scény mění jsem popsal již u 2.1 Hlavní menu. Třída GameGridController přidává k samotnému poli z třídy Field i pole grafické. Třída si pamatuje logické pole typu Field, dále si pamatuje pole obrázků, Labelů a Progress barů. Tyto pole se nastaví v metodě initializeBattlefield při startu hry. Metoda vytvoří posledním třem jmenovaným proměnným pole o rozměrech WIDTH a HEIGHT ze třídy Field.
Dále dva do sebe vnořené for-loopy nastaví a vloží proměnné do příslušných polí.
Pole pro příslušné proměnné jsou zde, aby se s nimi mohlo pracovat i později, nejen při jejich vytvoření.
Po dokončení přiřazování hodnot se vytvoří objekt Field a uloží se do výše zmíněného Fieldu. Potom skončení této metody se volá metoda initializeArmy, která pomocí metody assignPos nastaví výchozí rozpoložení jednotek a zavolá funkci updateMap.
UpdateMap je metoda, která se volá po každé změně v logickém poli a provede stejnou změnu i na poli grafickém. Metoda použije dva for-loopy, které podle toho, jestli je nebo není na daném místě postava nastaví GridPane. Navíc tato metoda vypíše, pokud nějaký z hráčů prohraje, a také vypisuje, který hráč je na řadě. Též nastavuje příslušné barvy příslušným políčkům podle toho, zda je zde červená nebo modrá jednotka. Pro odpovědi, jak změnit barvu progressbaru a jak změnit velikost textu, jsem se koukal na Stackoverflow[2][3], avšak jsem žádné z doporučených řešení nepoužil, jelikož se stačilo podívat do SceneBuilderu, kde jsem zjistil, že existuje jiné řešení, které využívá jiné třídy jako Effect nebo Font.
Nakonec je zde mousePressed event, který jsem nakonec vybral jako nejlepší možnost řešení interakce hráče. Event nastane kdykoliv hráč klikne pod prvních 30 pixelů obrazovky. Tato část je vyhrazena tlačítku zpět a také zaškrtávacímu tlačítku edit. Tato metoda využívá dvě *boolean* proměnné editing a moving. Pokud je editing *true* tak hráč může mazat postavy a vytvářet nové po dvaceti. Pokud je moving i editable *false* tak si třída zapamatuje souřadnice vybraného políčka a jednotky v ní. Pokud se zde nenachází jednotka nebo je to jednotka hráče, který není na řadě tak se nic neprovede. Pokud je zde jednotka hráče co je na řadě tak se jednotka stane poloprůhlednou a move se nastaví na *true*. Pokud je move *true*, tak může hráč útočit, popojít, sloučit jednotky případně zaútočit na dálku nebo zrušit tah tak, že klikne na průhlednou jednotku. Po každém úspěšném tahu(tedy po tom co nebyl zrušen) se přidá jednička do počtu provedených tahů. Když se bude počet provedených tahů rovnat maximálnímu číslu tahů tak bude na řadě druhý hráč a počet provedených tahů se nastaví zpět na nulu. Počet maximálních tahů je defaultně nastaven na 2 tahy za kolo.

4. Závěr

Myslím si, že by práce šla ještě vylepšit. Během psaní mě napadaly nové myšlenky, co bych mohl přidat. Jedna z nich bylo tlačítko pro možnost úpravy bojiště, které jsem nakonec přidal. Další je například omezení pohybu jednotek tak, aby se jednotka mohla pohnout jen jednou za kolo a hráč tak musel pohnout poté s jinou. Také jsem přemýšlel nad uložením pole do souboru a přidání načítání ze souboru. Tlačítko pro úpravy by se pak dalo využít k připravení scénáře, který by šel uložit. To se mi bohužel zatím nepovedlo, avšak nebál bych se říct, že hra je nyní v hratelném stavu. Myslím si, že i kdybych splnil tyto nedostatky, tak bych si našel další věci na doladění, či vylepšení.
Do budoucna bych chtěl zkusit vytvořit nějakou hru co nebude zrovna tahová, čili bude se moct obnovovat každý snímek místo tahu. Jako například nějakou 2D arkádovou plošinovku.

5. Zdroje

[1] JavaFX multiple screen app in 4 min. Dostupné z URL:
https://www.youtube.com/watch?v=C-ReKeKSQrw
[2] How to change the text font size ? Dostupné z URL:
https://stackoverflow.com/questions/22047457/how-to-change-the-text-font-size-in-javafx
[3]JavaFX ProgressBar: how to change bar color? Dostupné z URL:
https://stackoverflow.com/questions/13357077/javafx-progressbar-how-to-change-bar-color

6.Obrázky