Gymnázium, Praha 6, Arabská 14

Obor programování

Ročníková práce

Miroslav Sklenář

Hrdinský tahový simulátor boje

Květen 2020

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 ……. dne …… Miroslav Sklenář…….

Název práce: Hrdinský tahová simulátor boje

Autoři: Miroslav Sklenář

Abstrakt: Cílem projektu bylo naprogramovat hrdinskou tahovou simulaci boje s monstry. Uživateli měl být dán prostor k volbě strategie boje. Uživatel má tak na výběr možnosti jestli se má bránit či útočit nebo dokonce může utéct. Monstra se generují náhodně a jejich statistiky se odvíjejí od jejich levelu.

Klíčová slova: tahová, boj , javaFX, java

Obsah

1. Úvod

V tomto dokumentu se budeme zabývat hrdinskou tahovou simulací boje. Tato simulace (později už jen jako „hra“) měla za úkol vytvořit bojový systém, do kterého jsou implementovány statistiky hrdiny a zároveň i jeho nepřátel. Hra také měla mít možnost vybrat si, jaký přístup boje si zvolíte např. útok/obrana. Monstra jsou však nastavena, aby vždy útočily. Původní projekt měl být jen v textovém formátu, ale za pomocí JavaFX se mnohé usnadnilo a otevřely se autorovi nové možnosti na vizualizaci.

Zadání

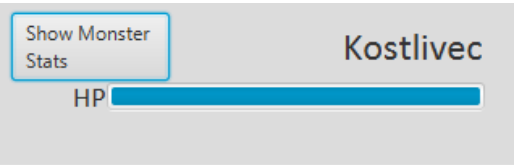
Hra má mít následující prvky:

1. Hrdinu se statistikami
2. Náhodně generovaná monstra se statistikami
3. Základní možnosti rozhodování

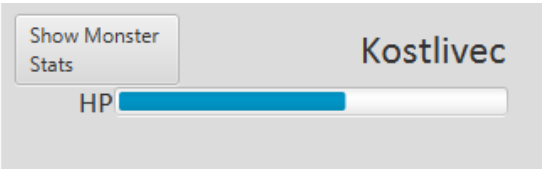
2. Vzhled rozhraní

2.1 Health bar

Health bar je vizuální reprezentace zdraví hrdiny i jeho nepřátel. Když hrdina zaútočí na nepřítele a zraní ho, tak nepříteli klesne zdraví a v závislosti na tom se upraví health bar do pozice přímo úměrné zdraví nepřítele. To samé platí i pro hrdinu.



Obr. 1 Reprezentace plného HP baru



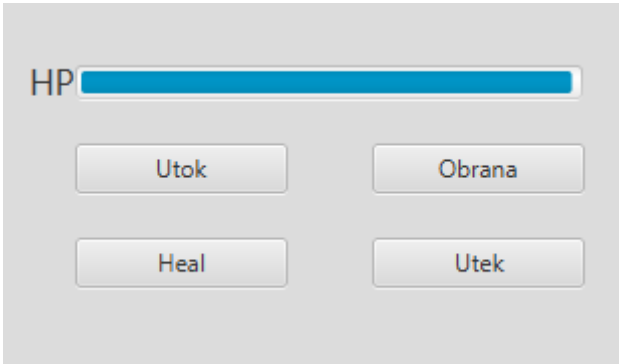
Obr. 2 Reprezentace HP baru po útoku

2.2 Tlačítka

2.2.1 Akce Hrdiny

Tlačítka vizuálně reprezentují akce hrdiny. Můžete si tedy vybrat, co bude hrdina dělat v tomto kole. Na výběr jsou čtyři tlačítka akcí hrdiny:

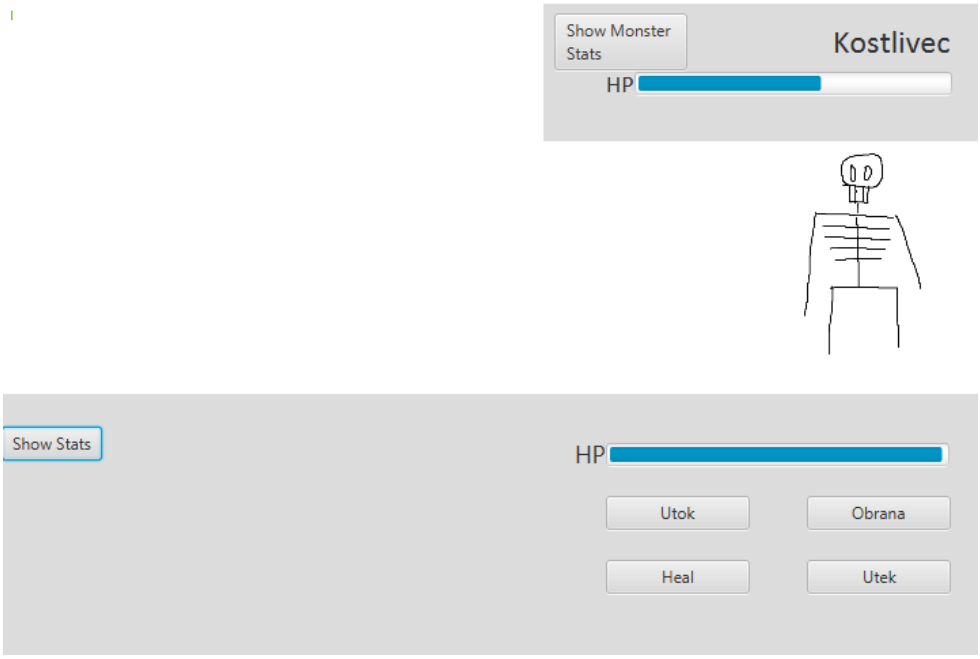
Útok
Obrana
Heal
Útěk
Co každé tlačítko dělá,
se dozvíte v kapitole **3. Funkčnost.**



Obr. 3 Akce hrdiny

2.2.2 Zobrazení statistik

V levé dolní polovině lze nalézt také tlačítko, které se jmenuje *Show Stats*
To zobrazí statistiky hrdiny v textovém formátu. To samé se dá nalézt v horním středu, akorát pro monstra.

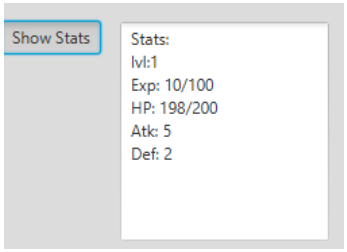


Obr. 4 Uživatelské rozhraní

2.3 Okna se statistikami

Po rozkliknutí tlačítka *Show Stats* nebo *Show Monster Stats* se zobrazí okno, kde jsou uvedeny všechny statistiky. Statistiky jsou následující:
lvl-level hrdiny/monstra
Exp - zkušenosti hrdiny
HP - zdraví hrdiny/monstra
Atk - útok hrdiny/monstra
Def - obrana hrdiny/monstra

Obr. 5 Statistiky monstra



Obr. 6 Statistiky hrdiny

3. Funkčnost

3.1 Útok

Útok se stane, pokud kliknete na tlačítko útok. Útok monster se stane pokaždé a jediná možnost jak mu předejít je zvolit obranu avšak i ta má nějaké své limity. Útok hrdiny je však ovlivněn obranou monster. A tak se od útoku hrdiny odečte obrana monstra. Jelikož pak mohl nastat problém, pokud obrana monstra byla vyšší než hrdinův útok, pak by se monstru přičítaly životy. Což je nesmysl. A tak je tento problém vyřešen přes podmínku. Pokud je rozdíl útoku hrdiny a obrany monstra menší nebo roven nule, pak se tento rozdíl bude vždy rovnat nule. A díky této podmínce se nebudou přičítat životy a vše funguje tak jak má.

```
private void Utok(ActionEvent event) {
    int kkk = Hrdina.atk - LocalDef;
    if (kkk <= 0) {
        kkk = 0;
    } else {
        LocalHP = LocalHP - kkk;
        double HPBar = (double) LocalHP / (double) LocalMaxHP;
        HPMonstraBar.setProgress(HPBar);
        Hrdina.HP = Hrdina.HP - LocalAtk;
        double k = (double) Hrdina.HP / (double) Hrdina.MaxHP;
        HPHrdinaBar.setProgress(k);
    }
```

Obr. 7 Kus kódu reprezentující útok hrdiny

3.2 Obrana

Obranný systém má 2 části – pasivní a aktivní. Pasivní systém funguje pro monstra a aktivní je pro hrdinu/hráče.

Pasivní systém odečítá od útoku hrdiny obranu monster. Tady to je nastaveno tak že nemůže nastat situace, kdy by se tento rozdíl dostal do mínusu. Takže není nutné tento problém ošetřovat.

Aktivní systém funguje tak, že si hráč zvolí *Obranu* a pokud má vyšší *Def* než monstrum *Atk* pak nedostane poškození. Tady už bylo nutné ošetřit minusový rozdíl, a tudíž vznikla podmínka, která tento rozdíl nastavila na nulu.

```
private void Obrana(ActionEvent event) {
    int ll = LocalAtk - Hrdina.def;
    if (ll <= 0) {
        Hrdina.HP = Hrdina.HP;
        double k = (double) Hrdina.HP / (double) Hrdina.MaxHP;
        HPHrdinaBar.setProgress(k);
        LocalHP = LocalHP - 5;
        double HPBar = (double) LocalHP / (double) LocalMaxHP;
        HPMonstraBar.setProgress(HPBar);
    } else {
        Hrdina.HP = Hrdina.HP - (int) (ll / 4);
        double k = (double) Hrdina.HP / (double) Hrdina.MaxHP;
        HPHrdinaBar.setProgress(k);
        LocalHP--;
        double HPBar = (double) LocalHP / (double) LocalMaxHP;
        HPMonstraBar.setProgress(HPBar);

    }
```

Obr. 8 Kus kódu reprezentující obranu hrdiny

3.3 Heal

Heal system(léčivý systém) funguje na podobném principu jako obrana. Zásadní rozdíl je pak v tom, že tento systém je jen dočasný a má sloužit jako předchůdce inventáře.

Jakmile si zvolíte *Heal*, tak to vyléčí hrdinu pomocí šance za nějaký počet zdraví. Tady mohl nastat problém. Pokud by se hráč rozhodl zvolit *Heal* při maximálním počtu zdraví, pak by jeho aktuální zdraví překročilo jeho maximální. A to se stát nemá a tak tu je podmínka, která tomu zabrání. Pokud by hrdinovo zdraví bylo vyšší než jeho maximální, pak se jeho zdraví nastaví na maximální.

Aby hráč nemohl do nekonečna používat *Heal* byl zde nastaven limit dvou použití za jeden souboj.

```java
@FXML
private void Heal(ActionEvent event) {
    int kk = (int) (Math.random() * 9);
    if (kk <= 4) {
        Hrdina.HP = Hrdina.HP + 10;
    }
    if (kk > 4 && kk <= 7) {
        Hrdina.HP = Hrdina.HP + 15;
    }
    if (kk > 7) {
        Hrdina.HP = Hrdina.HP + 35;
    }
    if (Hrdina.HP > Hrdina.MaxHP) {
        Hrdina.HP = Hrdina.MaxHP;
    }
    Hrdina.HP = Hrdina.HP - (LocalAtk / 2);
    double k = (double) Hrdina.HP / (double) Hrdina.MaxHP;
    HPHrdinaBar.setProgress(k);
    LocalHeal++;
    if (LocalHeal == 2) {
        BtnHeal.setDisable(true);
    }
```

Obr. 9 Kus kódu reprezentující léčení

3.4 Útěk

Systém útěku, je tu jako poslední možnost. Lze jej využít jako poslední záchranu, když prohráváte souboj nebo také pokud hráč není spokojen s úrovní jeho protivníka.

Když si hráč zvolí možnost útěku tak se mu doplní životy na maximum, nedostane žádné XP a vygeneruje se nový protivník.

3.5 Generování monster

Generování monster je řešeno pomocí náhody a switche.
Každé monstrum má předdefinované číslo ve switchi, a když je zvoleno vygeneruje se mu akorát level. A podle něj se přepočítají hodnoty jeho statistik.

```
int kk = (int) (Math.random() * 3);
String jmenol = "";
switch (kk) {
    case 0:
        jmenol = "Slime";
        Locallvl = (int) (Math.random() * 3 + Hrdina.lvl);
        ImgSlime.setVisible(true);
        break;
    case 1:
        jmenol = "Kostlivec";
        Locallvl = (int) (Math.random() * 3 + Hrdina.lvl * 2);
        ImgKostlivec.setVisible(true);
        break;
    case 2:
        jmenol = "Zombie";
        Locallvl = (int) (Math.random() * 4 + Hrdina.lvl);
        ImgZombie.setVisible(true);
        break;
}
```

Obr. 10 Reprezentace generování monster

Závěr

Myslím si, že projekt se celkem povedl. Je zde však obrovský prostor pro vylepšení. Projektu by velmi prospělo, kdyby byl zakomponován do nějaké menší hry. Kde by se dalo využít funkce inventáře, který má nahradit funkci *Heal* nebo naplno využít funkce Útěk. Dále by nebylo špatné zpestřit onen boj např. přidáním animací nebo speciálních efektů. Projekt má velký potenciál do budoucna, ale s nynějšími znalostmi autora jej autor nemůže zcela naplnit.

Bibliografie

V této práci nebylo použito žádného jiného díla jiného autora.

Seznam obrázků