|  |
| --- |
| Erik Urban |
| Semestrálna práca  Hra Slayer |
| Skupina 5ZI1B |

|  |
| --- |
| 23.5.2021 |

**Slayer**

Tento projekt predstavuje Roguelike RPG Adventúru, ktorú som nazval Slayer.  
Hráčovi je postavený v ceste nepriateľ, prípadne iný interaktívny objekt. Hráčov pohľad je z prvej osoby.

Hráč interaguje s prostredím klikaním ľavého tlačidla myši na príslušný interaktívny objekt. Na začiatku každej scény, hra čaká na zaútočenie (kliknutie) hráča na nepriateľa.

Následne sa hráč dostáva do boja a je mu umožnené používať špeciálne schopnosti, ktoré vylepšujú jeho sily. Tieto špeciálne schopnosti majú cenu tzv. energiu. Energia sa dopĺňa na predvolenú hodnotu vždy po zmene scény.

Hra má momentálne implementované dve sily – Sila útoku a Sila liečenia.  
Obe hodnoty týchto síl môžu byť dočasne navyšované. Po zaútočení, alebo liečení sa tieto hodnoty vrátia na predvolenú hodnotu. Obe sily sú od seba nezávisle.

Útok nepriateľa prebieha hneď po útoku alebo liečení hráča.  
Používanie špeciálnych schopností nevyvolá odpoveď od nepriateľa.  
Po každom útoku alebo liečení sa zvýši aktuálna energia o jeden, až po maximálnu predvolenú hodnotu.

Za porazenie nepriateľa hráč dostáva odmenu, predstavujúcu hernú menu tzv. peniaze. Tieto peniaze môže hráč využiť na vylepšenie svojich atribútov.

Koniec hry nastáva po porazení posledného nepriateľa alebo po dosiahnutí nulového zdravia hráča.

Hráč vyhrá hru, ak porazí posledného nepriateľa v hre.

**Vytvorené Objekty**

Program sa skladá z 4 packages, menovite sú to :

Manazeri : Obsahuje triedy, ktoré manažujú interakciu medzi objektami, sú to: ManazerHrania, ManazerTlacidiel, ManazerZobrazenia.  
  
Nepriatelia: Obsahuje triedy, ktoré predstavujú nepriateľov v hre, sú to: Cerberus, Goblin, Lich, Nepriatel, Ork.

Obrazky : Táto package obsahuje obrázky použité v hre.

Povolania : Obsahuje triedy, ktoré predstavujú zosobnenie hráča v hre, sú to Bojovník, Kuzelnik, Lukostrelec, Postava.

Mimo package-u sa nachádzajú triedy :  
GUI – predstavuje grafické prostrenie, využíva hlavne knižnicu Swing.  
Hra – slúži na spustenie hry a vytvorenie inštancií Manažérov a GUI.  
Main – táto trieda obsahuje metódu main, v ktorej sa vytvorí inštancia hra triedy Hra.  
Nachádza sa tu aj obrázok, použitý ako ikona hry.

**Trieda Main**

Táto trieda obsahuje statickú metódu main. V nej sa nám inštancia triedy Hra.

**Trieda Hra**

Táto trieda slúži na spustenie hry a vytvorenie inštancií Manažérov a GUI.

**Konštruktor :**

Konštruktor, ktorý inicializuje atribúty gui, manazerZobrazenia a manazerTlacitiek. Taktiež vytvorý lokálnu premennú manazerHrania a zavolá metódu spustiHru();

**Metódy :**

public void spustiHru() - Metóda, ktorá spustí novú hru.

**Trieda GUI**

Táto trieda reprezentuje Graphical User Interface.

**Konštruktor :**

Konštruktor, v ktorom sa inicializuje atribút okno - JFrame, špecifikujú sa jeho vlastnosti. Inicializjú sa i atribúty typu ArrayList , ktoré budú obsahovat JPanel-y, JLabel-y a JButton-y.

**Metódy :**

**vytvorHlavneMenu(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky hlavného menu. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**vytvorMeno(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky tvorby mena . Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**vyberPovolaniaScreen(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky výberu povolia . Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**statusHUD()** - Táto metóda vytvorí grafické prvky HUD, ktoré predstavuje aktuálny stav postavy.

**les(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky scény les . Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**nastavNepriatelaActionCommand()** - Táto metóda nastaví príkaz akcie, ktoré posielajú tlačidla, konkrétne na akého nepriateľa sa kliká.

**hlbokoVLese(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické scény hlboko v lese. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**chodDozaduGUI(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky, ktoré umožňujú vracanie scénamy. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**vytvorTextHUD(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky textového HUD. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**predJaskynou(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky scény pred Jaskyňou. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**vJaskyni(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky scény pred Jasykňou. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**lukaSObchodnikom(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky scény lúky s obchodníkom. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**menuObchodnik(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky, ktoré umožňujú nákup u obchodníka. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**vytvorKoniecTlacitka(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky, ktoré umožňujú koniec hry. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**smrtScreen(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky, ktoré umožňujú zobraziť posmrtnú scénu. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**vyhraScreen(ManazerTlacidiel manazerTlacidiel)** - Táto metóda vytvorí grafické prvky, ktoré umožňujú zobraziť výhernú scénu. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**nastavTlacitkaTextACommand()** - Táto metóda nastaví príkaz akcie, ktoré posielajú tlačítká, konkrétne pre tlačítka schopností postavy.

**tvorbaGUIOstatne(ManazerTlacidiel manazerTlacidiel)** - Táto metóda zavolá všetky metódy tvorby grafiky PO vytvorení postavy. Parameter manazerTlacidiel je manažér tlačidiel, prijíma input, na ktorý následne reaguje volaním konkrétnych metód.

**zobrazGUI()** - Táto metóda vytvorí zobrazí okno, v ktorom sa zobrazuje hra.

**getHlavnyText()** - Táto metóda vráti hlavný text, kde sa vypisuje priebeh hry.

**getPanely()** - Táto metóda vráti ArrrayList panelov.

**getLabely()** - Táto metóda vráti ArrrayList labelov.

**getTextMeno()** - Táto metóda vráti textové pole, kde sa píše meno postavy.

**getTlacidla()** - Táto metóda vráti ArrrayList tlačidiel.

**getOkno()** - Táto metóda vráti okno hry.

**nastavLabel(JLabel label)** - Táto metóda nastaví label. Parameter label predstavuje konkrétny label, ktorý chceme nastaviť.

**nastavLabelHUD(JLabel label)** - Táto metóda nastaví HUD label. Parameter label predstavuje konkrétny label, ktorý chceme nastaviť.

**nastavPanelDefault(JPanel panel)** - Táto metóda nastaví panel na predvolené vlastnosti. Parameter panel predstavuje konkrétny panel, ktorý chceme nastaviť.

**nastavPozadieLabel(JLabel label, String adresaObr)** - Táto metóda nastaví label ako pozadie. Parameter label predstavuje konkrétny label, ktorý chceme nastaviť. Parameter adresaObr predstavuje adresu obrázka.

**nastavLabelObrazok(JLabel label)** - Táto metóda nastaví labelu obrázok. Parameter label predstavuje konkrétny label, ktorý chceme nastaviť.

**nastavButtonObrazok(JButton button, String adresaObr, ActionListener actionListener)** - Táto metóda nastaví tlačidlo a pridá mu obrázok. Parameter button predstavuje konkrétne tlačidlo, ktoré chceme nastaviť. Parameter adresaObr predstavuje adresu obrázka. Parameter actionListener predstavuje manažéra, ktorý načuvá akciu tlačídla.

**nastavLabelNadpis(JLabel label**) - Táto metóda nastaví label ako nadpis. Parameter label predstavuje konkrétny label.

**nastavButtonDefault** **(JButton button, JPanel panel, ActionListener actionListener, boolean malePismo)** - Táto metóda nastaví tlačidlo na predvolené vlastnosti. Parameter button predstavuje konkrétne tlačidlo. Parameter panel predstavuje do ktorého panelu je tlačítko pridané. Parameter actionListener predstavuje manažéra, ktorý načuvá akciu tlačídla. Parameter malePismo predstavuje, veľkosť písma tlačítka.

**menuTlacidlo(JButton button)** - Táto metóda nastaví tlačídlo na vlastnosti tlačítka pre menu. Parameter button predstavuje konkrétne tlačítko .

Trieda Postava

Táto trieda predstavuje rodiča povolaní.

**Konštruktor :**

**Postava(int maxZivoty, int peniaze, int silaUtoku, int maximalnaEnergia, int silaLiecenia )** - Konštruktor, ktorý nastaví atribúty postavy podľa parametrov. Parameter maxZivoty reprezentuje základné maximálne životy postavy. Parameter peniaze reprezentuje základné peniaze postavy. Parameter silaUtoku reprezentuje základnú silu útoku postavy. Parameter maximalnaEnergia reprezentuje základnú maximálnu energiu postavy. Parameter silaLiecenia reprezentuje základnú silu liečenia postavy.

**Metódy :**

**nastavSily(int silaUtoku, int silaLiecenia) -** Táto metóda nastaví sily postavy podľa parametrov. Parameter silaUtoku reprezentuje nastavovanú hodnotu sily útoku. Parameter silaLiecenia reprezentuje nastavovanú hodnotu sily liečenia.

**nastavPlnuEnergiu()** - Táto metóda nastaví plnú energiu postave.

**nastavPlneZivoty()** - Táto metóda nastaví plné životy postave.

**getSilaLiecenia()** - Táto metóda nastaví plné životy poTáto metóda vráti aktuálnu silu liečenia postavy.stave.

**getPeniaze()** - Táto metóda vráti aktuálne peniaze postavy.

**getSilaUtoku()** - Táto metóda vráti aktuálnu silu útoku postavy.

**getZivotyAktualne()** - Táto metóda vráti aktuálne životy postavy.

**znizZivoty(int hodnota)** - Táto metóda zníži životy postave. Parameter hodnota reprezentuje o koľko sa atribút zníži.

**zvysZivoty(int hodnota)** - Táto metóda zvýši životy postave. Parameter hodnota reprezentuje o koľko sa zvýši atribút.

**getMaxZivoty()** - Táto metóda vráti hodnotu maximálnych životov.

**zvysSiluLiecenia(int hodnota)** - Táto metóda zvyši silu liečenia postave. Parameter hodnota reprezentuje o koľko sa zvýši atribút.

**znizEnergiu(int hodnota)** - Táto metóda zníši energiu postave. Parameter hodnota reprezentuje o koľko sa zníši atribút.

**zvysEnergiu(int hodnota)** - Táto metóda zvýši energiu postave. Parameter hodnota reprezentuje o koľko sa zvýši atribút.

**zvysSiluUtoku(int hodnota)** - Táto metóda zvýši silu utoku. Parameter hodnota reprezentuje o koľko sa zvýši atribút.

**getEnergia()** - Táto metóda vráti atribút energie postavy.

**zvysPeniaze(int hodnota)** - Táto metóda zvýši peniaze. Parameter hodnota reprezentuje o koľko sa zvýši atribút.

**znizPeniaze(int hodnota)** - Táto metóda zníž peniaze. Parameter hodnota reprezentuje o koľko sa zníži atribút.

**zvysMaximalneZivoty(int hodnota)** - Táto metóda zvýši maximálne životy postave. Parameter hodnota reprezentuje o koľko sa zvýši atribút.

Trieda Bojovnik

Táto trieda Bojovnik predstavuje potomka triedy charakter.

**Konštruktor :**Konštruktor, ktorý nastaví postavu podľa daných konštánt.

Trieda Lukostrelec

Táto trieda Lukostrelec predstavuje potomka triedy charakter.

**Konštruktor :**Konštruktor, ktorý nastaví postavu podľa daných konštánt.

Trieda Kuzelnik

Táto trieda Kuzelnik predstavuje potomka triedy charakter.

**Konštruktor :**Konštruktor, ktorý nastaví postavu podľa daných konštánt.

Trieda Nepriatel

Táto trieda predstavuje rodiča nepriateľov.

**Konštruktor :**

**Nepriatel(String nazovNepriatela, int maximalneZivoty, int silaUtoku, int odmena) -** Konštruktor, ktorý nastaví postavu podľa daných konštánt. Parameter nazovNepriatela reprezentuje názov nepriateľa. Parameter maximalneZivoty reprezentuje maximálne životy nepriateľa. Parameter silaUtoku reprezentuje silu útoku nepriateľa. Parameter odmena reprezentuje odmenu za porazenie nepriateľa.

**Metódy :**

**getZivoty()** - Táto metóda vráti aktuálne životy nepriateľa.

**znizZivoty(int hodnota)** - Táto metóda zníži aktuálne životy nepriateľa.

**getNazovNepriatela()** - Táto metóda vráti pomenovanie nepriateľa.

**getSilaUtoku()** - Táto metóda vráti silu nepriateľa.

**getOdmena()** - Táto metóda vráti odmenu za nepriateľa.

Trieda Goblin

Táto trieda Goblin predstavuje potomka triedy nepriatel.

**Konštruktor :**

Konštruktor, ktorý nastaví nepriateľa podľa daných konštánt**.**

Trieda Ork

Táto trieda Ork predstavuje potomka triedy nepriatel.

**Konštruktor :**

Konštruktor, ktorý nastaví nepriateľa podľa daných konštánt**.**

Trieda Cerberus

Táto trieda Cerberus predstavuje potomka triedy nepriatel.

**Konštruktor :**

Konštruktor, ktorý nastaví nepriateľa podľa daných konštánt**.**

Trieda Lich

Táto trieda Lich predstavuje potomka triedy nepriatel.

**Konštruktor :**

Konštruktor, ktorý nastaví nepriateľa podľa daných konštánt**.**

Trieda ManazerTlacidiel

Táto trieda implementuje Interface ActionListener, načúva akciám tlačidla - JButton .

**Konštruktor :**

**ManazerTlacidiel(ManazerZobrazenia manazerZobrazenia, ManazerHrania manazerHrania**) - Konštruktor, ktorý nastaví atribúty podľa prijatých parametrov. Parameter manazerZobrazenia reprezentuje manažéra zobrazenia. Parameter manazerHrania eprezentuje manažéra hrania.

**Metódy :**

**actionPerformed(ActionEvent e)** - Táto metóda je metóda, ktorú musí trieda implementovať, lebo je implementovaná Interface-om ActionListener. \* Načúva akciám tlačidla. Parameter e predstavuje akčný event - udalosť.

**prepinajButtony() -** Táto metóda zneprístupní tlačidlo, ak postava nemá dosť energie.

Trieda ManazerZobrazenia

Táto Trieda slúži na zobrazovanie Swing komponentov - GUI.

**Konštruktor :**

**ManazerZobrazenia(GUI gui) -** Konštuktor, ktorý inicializuje atribút GUI. Parameter gui reprezentuje parameter, na ktorý sa nastaví atribút gui.

**Metódy :**

**zobrazNovuHru()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu novej hry.

**zobrazTvorbuMena()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu tvorby mena.

**zobrazTvorbuPovolania()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu tvorby povolania.

**zobrazLes()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu lesa.

**hlbokoVLese()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu hlbokov v lese.

**zobrazPredJaskynou()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu pred jaskynou.

**zobrazVJaskyni()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu v jaskyni.

**zobrazSmrt()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu smrti hráča.

**zobrazVyhru()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu výhry hráča.

**zobrazLukuSObchodnikom()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu lúky s obchodníkom.

**zobrazPonuku()** - Táto metóda zobrazí panel, ktorý drží komponenty pre scénu obchodnej ponuky.

**zobrazStatusHUD(boolean zobraz**) - Táto metóda zobrazí panel, ktorý drží komponenty status HUD. Parameter zobraz , ak je parameter pravdivý a zároveň je atribút zvacsiOkno nepravdivý, tak zväčší okno a nastaví atribút zvacsiOkno na pravdu.

**vypniVsetko()** - Táto metóda vypína všetky prvky hry.

**zobrazGoblin(boolean uzRazPorazeny)** - Táto metóda zobrazuje Goblin nepriateľa. Parameter uzRazPorazeny, ak je pravdivý, tak nepriateľ sa nezobrazí. Slúži pri cestovaní naspäť v hre.

**smrtGoblina()** - Táto metóda zobrazí smrt Goblina.

**zobrazOrka(boolean uzRazPorazeny)** - Táto metóda zobrazuje Ork nepriateľa . Parameter uzRazPorazeny, ak je pravdivý, tak nepriateľ sa nezobrazí. Slúži pri cestovaní naspäť v hre.

**smrtOrka()** - Táto metóda zobrazí smrt Orka.

**zobrazCerberusa(boolean uzRazPorazeny)** - Táto metóda zobrazuje Cerberus nepriateľa . Parameter uzRazPorazeny, ak je pravdivý, tak nepriateľ sa nezobrazí. Slúži pri cestovaní naspäť v hre.

**smrtCerberusa()** - Táto metóda zobrazí smrt Cerberusa.

**zobrazLich()** - Táto metóda zobrazuje Lich nepriateľa.

**dajZoznamTlacidiel()** - Táto metóda vráti ArrayList tlačidiel.

**dajZoznamLabelov()** - Táto metóda vráti ArrayList labelov.

**zmenaHlavnehoTextu(String zmena)** - Táto metóda nastaví hlavný text hry podľa parametra. Parameter zmena predstavuje zmenu textu.

**getMeno()** - Táto metóda vracia meno, ktoré zadal hráč pri zadávaní mena. Ak hráč nezadal meno, nastaví sa na "noName". \* Return vráti meno postavy hráča.

Trieda ManazerHrania

Táto trieda predstavuje manažéra hrania hry.

**Konštruktor :**

**ManazerHrania(ManazerZobrazenia manazerZobrazenia)** - Konštruktor, ktorý nastaví atribút manazerZobrazenia podľa prijatého parametra. Parameter manazerZobrazenia reprezentuje prijatý parameter.

**Metódy :**

**vytvorPociatocnyStav(String meno, String povolanie)** - Táto metóda nastavý vybraté povolanie a meno hráča. Parameter meno reprezentuje meno hráča. Parameter povolanie reprezentuje povolanie hráča.

**nastavNepriatela(int cislo**) - Táto metóda nastavý nepriateľa podľa prijatého int parametra. Parameter cislo reprezentuje číselnú reprezentáciu nepriateľa.

**bojSNepriatelom()** - Metóda, ktorá reprezentuje boj s nepriateľom.

**smrtNepriatela(Nepriatel nepriatel)** - Metóda, ktorá reprezentuje smrť nepriateľom. Parameter nepriatel predstavuje nepriateľa, ktorý zomrel.

**nakupovanie(String nakup)** - Metóda nakupovania u obchodníka. Príjma parameter String, ktorý reprezentuje príkaz akcie tlačítka. Parameter nakup predstavuje akciu tlačítka.

**vyliecSa()** - Metóda reprezentuje liečenie hráča.

**zvysovanieSil(String actionCommand) -** Metóda slúži na zvyšovanie síl hráča. Parameter actionCommand reprezentuje príkaz akcie tlačidla.

**vyresetujSily(boolean ibaUtok, boolean ibaLiecenie)** - Táto metóda vracia sily na predvolené hodnoty. Parametre slúžia na identifikáciu toho, či sa má resetovať utok, liečenie alebo obe. Táto metóda sa použije pri použití útoku - iba útok sa resetne, liečnia - iba liečenie sa resetne, prechod do ďalšej scény - obe sa resetnu. Parameter ibaUtok , ak je pravdivý, tak sa resetuje iba útok. Parameter ibaLiecenie , ak je pravdivý, tak sa resetuje iba liečenie.

**screenSmrti()** - Táto metóda zavolá obrazovku smrti.

**aktualizujHUD()** - Táto metóda aktualizuje HUD. Vykonáva sa vždy po akcii v hre.

**getAktulanuEnergiu()** - Táto metóda vráti aktuálnu energiu postavy.