**Můj deníček ke hře**

Obsah

[Zadání hry: 1](#_Toc66775515)

[Můj výkaz 2](#_Toc66775516)

[Datum: doba trvání 2](#_Toc66775517)

[Datum: doba trvání 2](#_Toc66775518)

[Datum: doba trvání 2](#_Toc66775519)

[Návod ke hře 3](#_Toc66775520)

# Zadání hry:

Chtěl jsem zařadit časomíru, ale po dvou dnech stále neúspěšně. Nakonec jsem se rozhodl časomíru vypustit.

# Můj výkaz

## Datum: 26.2. 13:00-14:00

Vymyšlení projektu – vymyšlení názvu, konceptu, popisu, úkolů

## Datum: 10.4. 13:00-22:00

Tvorba základů – vytvoření konceptů

Hledání grafiky – hledání obrázků, stahování a předání Kiszovi na vytvoření grafiky (z důvodu slabého výkonu ve starém pc (pc mi nerozjel žádný program, ani na stránce), na novém pc jsem měl zdroj na reklamaci a čekal na vyřízení)

Tvorba levelu jedna – tvorba levelu jedna skrz for cyklus

Tvorba playera – vytvoření třídy player, všech mechanik a konceptů

Tvorba neprůchodných bloků / zdí

Tvorba konceptu easter eggu

## Datum: 15.4. 23:00 – 1:00

Vytvoření levelu dva – kompletní tvorba druhého levelu

Vytvoření tříd a světů – vytvoření potřebných tříd (buttony, gatey) a světů (StartW, GOWorld, level 2, 3 a win obrazovka)

Dokončení přesunu dat – ze starého PC do nového PC

Doladění důležitých systémových prací – otevírání brán, přestup mezi levely

## Datum: 16.4. 9:30-12:00

Třetí level – kompletní tvorba třetího a závěrečného bludiště

Doladění easter eggu – přenastavení easter eggu z hráče na GateThree, úprava Gate a GateTwo, aby zde easter egg nefungoval

Úprava třídy hráče a světů – přenesení int keys z konstruktoru do povinného zadání Player, odstranění easter eggu z nextL, nextLTwo a win (easter egg je v GateThree.class), dopsání proměnné int do světů s hráčem

Dokončení textových dokumentů – zapsání deníku a soubor Readme.txt

## Datum: 18.4. 17:00-19:00

Dokončení hry – dodání buttonů do světů

Dodání grafiky – nastavení grafiky světů

Export projektu

# Návod ke hře

„Učební text“ popisující, jak jsem hru tvořil od začátku až po konec.

Po vytvoření projektu v greenfootu, vložíte do složky „sounds“ potřebné zvukové efekty.

Do složky „images“ všechny potřebné externí obrázky pro grafiku.

Vytvoříme světy:

**MyWorld**

import greenfoot.\*;

public class MyWorld extends World

{

private Player player;

public MyWorld()

{

super(600, 600, 1);

prepare();

}

public Player getPlayer()

{

return this.player;

}

private void prepare()

{

player = new Player(0);

addObject(player,197,61);

setPaintOrder(Player.class, Key.class, Gate.class, Block.class);

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block,1+30\*citac, 0);

}

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block,1+30\*citac, 600);

}

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block, 0, 1+30\*citac);

}

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block, 600, 1+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 271, 1+30\*citac);

}

for(int citac = 0; citac <= 6; citac++)

{

Block block = new Block();

addObject(block, 391+30\*citac, 151);

}

for(int citac = 0; citac <= 5; citac++)

{

Block block = new Block();

addObject(block, 1+30\*citac, 151);

}

for(int citac = 0; citac <= 8; citac++)

{

Block block = new Block();

addObject(block, 271, 241+30\*citac);

}

for(int citac = 0; citac <= 11; citac++)

{

Block block = new Block();

addObject(block, 121+30\*citac, 481);

}

for(int citac = 0; citac <= 11; citac++)

{

Block block = new Block();

addObject(block, 121+30\*citac, 241);

}

for(int citac = 0; citac <= 5; citac++)

{

Block block = new Block();

addObject(block, 1+30\*citac, 361);

}

for(int citac = 0; citac <= 6; citac++)

{

Block block = new Block();

addObject(block, 391+30\*citac, 361);

}

Key key = new Key();

addObject(key,82,533);

key.setLocation(68,537);

Key key2 = new Key();

addObject(key2,549,537);

key2.setLocation(540,545);

Key key3 = new Key();

addObject(key3,306,345);

key3.setLocation(268,202);

Gate gate = new Gate();

addObject(gate,514,74);

}

}

**NextL2**

import greenfoot.\*;

public class NextL2 extends World

{

private Player player;

public NextL2()

{

super(600, 600, 1);

prepare();

}

public Player getPlayer()

{

return this.player;

}

public void prepare()

{

player = new Player(0);

addObject(player,75,67);

player.setLocation(58,58);

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block,1+30\*citac, 0);

}

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block,1+30\*citac, 600);

}

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block, 0, 1+30\*citac);

}

for(int citac = 0; citac <= 50; citac++)

{

Block block = new Block();

addObject(block, 600, 1+30\*citac);

}

for(int citac = 0; citac <= 15; citac++)

{

Block block = new Block();

addObject(block, 121+30\*citac, 481);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 121, 0+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 121, 241+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 0+30\*citac, 241);

}

for(int citac = 0; citac <= 11; citac++)

{

Block block = new Block();

addObject(block, 241+30\*citac, 241);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 241, 241+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 361, 361+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 481, 241+30\*citac);

}

for(int citac = 0; citac <= 2; citac++)

{

Block block = new Block();

addObject(block, 241+30\*citac, 121);

}

for(int citac = 0; citac <= 2; citac++)

{

Block block = new Block();

addObject(block, 421+30\*citac, 121);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 301, 0+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 421, 121+30\*citac);

}

setPaintOrder(Player.class, Key.class, Gate.class, Block.class);

Key key = new Key();

addObject(key,551,316);

key.setLocation(541,303);

GateTwo gateTwo = new GateTwo();

addObject(gateTwo,539,541);

gateTwo.setLocation(544,537);

Key key2 = new Key();

addObject(key2,455,184);

key2.setLocation(495,178);

Key key3 = new Key();

addObject(key3,49,291);

key3.setLocation(57,297);

}

}

**NextL3**

import greenfoot.\*;

public class NextL3 extends World

{

private Player player;

public NextL3()

{

super(600, 600, 1);

prepare();

}

public Player getPlayer()

{

return this.player;

}

public void prepare()

{

for(int citac = 0; citac <= 20; citac++)

{

Block block = new Block();

addObject(block,1+30\*citac, 0);

}

for(int citac = 0; citac <= 20; citac++)

{

Block block = new Block();

addObject(block,1+30\*citac, 600);

}

for(int citac = 0; citac <= 20; citac++)

{

Block block = new Block();

addObject(block, 0, 1+30\*citac);

}

for(int citac = 0; citac <= 20; citac++)

{

Block block = new Block();

addObject(block, 600, 1+30\*citac);

}

for(int citac = 0; citac <= 2; citac++)

{

Block block = new Block();

addObject(block, 481, 511+30\*citac);

}

for(int citac = 0; citac <= 6; citac++)

{

Block block = new Block();

addObject(block, 391+30\*citac, 391);

}

for(int citac = 0; citac <= 3; citac++)

{

Block block = new Block();

addObject(block, 361, 391+30\*citac);

}

for(int citac = 0; citac <= 6; citac++)

{

Block block = new Block();

addObject(block, 241, 301+30\*citac);

}

for(int citac = 0; citac <= 12; citac++)

{

Block block = new Block();

addObject(block, 121, 121+30\*citac);

}

for(int citac = 0; citac <= 8; citac++)

{

Block block = new Block();

addObject(block, 241+30\*citac, 271);

}

for(int citac = 0; citac <= 8; citac++)

{

Block block = new Block();

addObject(block, 121+30\*citac, 151);

}

for(int citac = 0; citac <= 2; citac++)

{

Block block = new Block();

addObject(block, 121, 211+30\*citac);

}

for(int citac = 0; citac <= 4; citac++)

{

Block block = new Block();

addObject(block, 361, 1+30\*citac);

}

for(int citac = 0; citac <= 1; citac++)

{

Block block = new Block();

addObject(block, 241, 1+30\*citac);

}

for(int citac = 0; citac <= 3; citac++)

{

Block block = new Block();

addObject(block, 481, 151+30\*citac);

}

for(int citac = 0; citac <= 3; citac++)

{

Block block = new Block();

addObject(block, 481+30\*citac, 151);

}

for(int citac = 0; citac <= 1; citac++)

{

Block block = new Block();

addObject(block, 481, 1+30\*citac);

}

setPaintOrder(Player.class, Key.class, Gate.class, Block.class);

GateThree gateThree = new GateThree();

addObject(gateThree,558,506);

gateThree.setLocation(543,540);

player = new Player(0);

addObject(player,439,543);

player.setLocation(439,547);

Key key = new Key();

addObject(key,298,62);

key.setLocation(301,70);

Key key2 = new Key();

addObject(key2,541,212);

Key key3 = new Key();

addObject(key3,500,89);

key3.setLocation(546,53);

}

}

**StartW**

**WinW**

**Objekty:**

**Block**

import greenfoot.\*;

public class Block extends Actor

{

public Block()

{

this.getImage().scale(30, 30);

}

}

**Player**

**Gate**

import greenfoot.\*;

public class Gate extends Actor

{

private Player player;

private final int Volume = 100;

private GreenfootImage open = new GreenfootImage("images/GateOp.png");

private GreenfootImage closed = new GreenfootImage("images/GateCl.png");

public Gate()

{

this.getImage().scale(70, 80);

this.open.scale(70, 80);

}

public void openGate()

{

if(((MyWorld)this.getWorld()).getPlayer().getKeys() == 3)

{

GreenfootSound gs = new GreenfootSound("OpG.mp3");

gs.setVolume(Volume);

gs.play();

gs.stop();

this.setImage(open);

}

}

public void act()

{

this.openGate();

}

}

**GateTwo**

import greenfoot.\*;

public class GateTwo extends Actor

{

private Player player;

private final int Volume = 100;

private GreenfootImage open = new GreenfootImage("images/GateOp.png");

private GreenfootImage closed = new GreenfootImage("images/GateCl.png");

public GateTwo()

{

this.getImage().scale(70, 80);

this.open.scale(70, 80);

}

public void openGate()

{

if(((NextL2)this.getWorld()).getPlayer().getKeys() == 3)

{

GreenfootSound gs = new GreenfootSound("OpG.mp3");

gs.setVolume(Volume);

gs.play();

gs.stop();

this.setImage(open);

}

}

public void act()

{

this.openGate();

}

}

**GateThree**

import greenfoot.\*;

public class GateThree extends Actor

{

private Player player;

private final int Volume = 100;

private GreenfootImage open = new GreenfootImage("images/GateOp.png");

private GreenfootImage closed = new GreenfootImage("images/GateCl.png");

public GateThree()

{

this.getImage().scale(70, 80);

this.open.scale(70, 80);

}

public void openGate()

{

if(((NextL3)this.getWorld()).getPlayer().getKeys() == 3)

{

GreenfootSound gs = new GreenfootSound("OpG.mp3");

gs.setVolume(Volume);

gs.play();

gs.stop();

this.setImage(open);

}

}

public void act()

{

this.openGate();

}

}

**Key**

import greenfoot.\*;

public class Key extends Actor

{

public Key()

{

this.getImage().scale(30, 40);

}

}

**Player**

import greenfoot.\*;

public class Player extends Actor

{

private int speed;

private int keys;

private int counter;

private Player player;

private GreenfootImage stopMove = new GreenfootImage("images/stop.png");

private GreenfootImage upMove = new GreenfootImage("images/up.png");

private GreenfootImage downMove = new GreenfootImage("images/down.png");

private GreenfootImage leftMove = new GreenfootImage("images/left.png");

private GreenfootImage rightMove = new GreenfootImage("images/right.png");

private final int StepVol = 100;

private final int GOVol = 100;

private final int TKVol = 50;

public Player(int keys)

{

this.speed = 5;

this.counter = 0;

this.getImage().scale(50, 60);

this.stopMove.scale(50, 60);

this.leftMove.scale(50, 60);

this.rightMove.scale(50, 60);

this.upMove.scale(50, 60);

this.downMove.scale(50, 60);

}

public void control()

{

if (Greenfoot.isKeyDown("left")){

setLocation(getX()-speed,getY());

GreenfootSound gs = new GreenfootSound("step.wav");

gs.setVolume(StepVol);

gs.play();

this.setImage(leftMove);

}

if (Greenfoot.isKeyDown("right")){

setLocation(getX()+speed,getY());

GreenfootSound gs = new GreenfootSound("step.wav");

gs.setVolume(StepVol);

gs.play();

this.setImage(rightMove);

}

if (Greenfoot.isKeyDown("up")){

setLocation(getX(),getY()-speed);

GreenfootSound gs = new GreenfootSound("step.wav");

gs.setVolume(StepVol);

gs.play();

this.setImage(upMove);

}

if (Greenfoot.isKeyDown("down")){

setLocation(getX(),getY()+speed);

GreenfootSound gs = new GreenfootSound("step.wav");

gs.setVolume(StepVol);

gs.play();

this.setImage(downMove);

}

}

public boolean touching()

{

if(isTouching(Block.class))

{

if(Greenfoot.isKeyDown("right"))

{

this.setLocation(this.getX()-speed,this.getY());

}

if(Greenfoot.isKeyDown("left"))

{

this.setLocation(this.getX()+speed,this.getY());

}

if(Greenfoot.isKeyDown("down"))

{

this.setLocation(this.getX(),this.getY()-speed);

}

if(Greenfoot.isKeyDown("up"))

{

this.setLocation(this.getX(),this.getY()+speed);

}

return true;

} else {

return false;

}

}

public boolean getKey()

{

if(this.isTouching(Key.class))

{

Key key = (Key)this.getOneIntersectingObject(Key.class);

this.getWorld().removeObject(key);

GreenfootSound gs = new GreenfootSound("TK.wav");

gs.setVolume(TKVol);

gs.play();

this.keys ++;

return true;

}else{return false;}

}

public int getKeys()

{

return this.keys;

}

public void nextL()

{

if(this.isTouching(Gate.class) && keys==3)

{

Greenfoot.setWorld(new NextL2());

}

}

public void nextLTwo()

{

if(this.isTouching(GateTwo.class) && keys==3)

{

Greenfoot.setWorld(new NextL3());

}

}

public void win()

{

if(this.isTouching(GateThree.class) && keys==3)

{

Greenfoot.setWorld(new WinW());

}

}

public void act()

{

this.control();

this.touching();

this.getKey();

this.nextL();

this.nextLTwo();

this.win();

}

}

**HoBut**

**ResBut**

**ResButThree**

**ResButTwo**

**StBut**

**Vše bylo vytvořeno v Greenfootu, pokud to zkusíte v jiném vývojovém prostředí, nebude Vám to fungovat!!!**