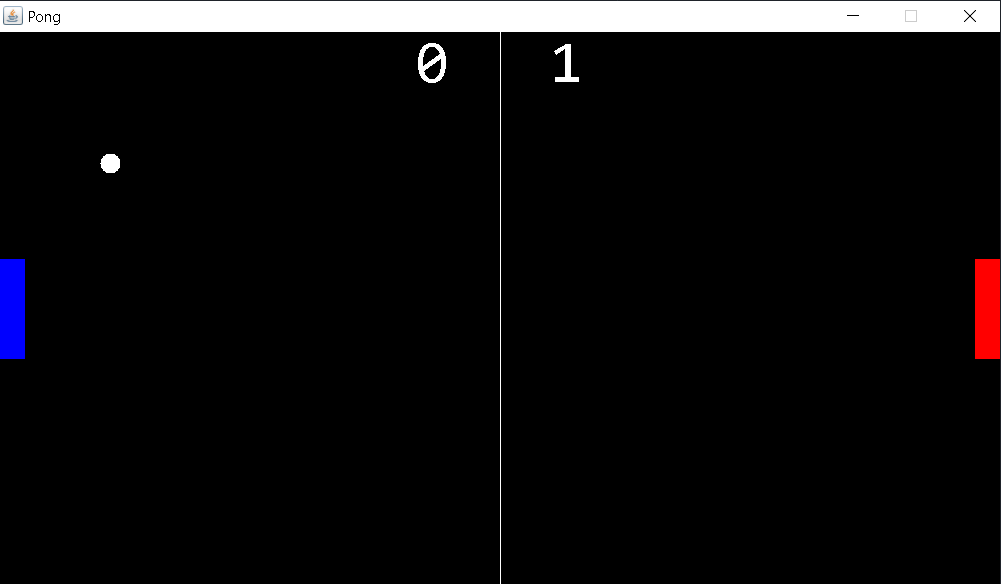
Hra Pong

ročníkový projekt



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 bezúplatně škole Gymnázium, Praha 6, Arabská 14 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………………. Pavelka Petr………………………

Obsah

[1 Anotace 3](#_Toc70332109)

[2 Zadání projektu 4](#_Toc70332110)

[3 Vlastní práce 5](#_Toc70332111)

[3.1 Struktura 5](#_Toc70332112)

[3.2 Game 5](#_Toc70332113)

[3.3 Frame 5](#_Toc70332114)

[3.4 Panel 6](#_Toc70332115)

[3.4.1 Check Collision 6](#_Toc70332116)

[3.5 Paddle 7](#_Toc70332117)

[3.6 Ball 9](#_Toc70332118)

[3.7 Score 9](#_Toc70332119)

[4 Závěr 10](#_Toc70332120)

[5 Zdroje 11](#_Toc70332121)

# 1 Anotace

Cílem tohoto projektu je naprogramovat hru Pong v programovacím jazyce Java, za použití knihoven Java.awt a Java.swing.

Hra je určena pro dva hráče hrající proti sobě na jednom počítači. Hra má neomezené skóre a herní dobu. Jejím účelem je procvičit pozornost, soustředění a rychlost reakce jednotlivých hráčů.

# 2 Zadání projektu

Dva hráči hrají na jednom počítači. V hracím poli lítá míček ze strany na stranu. Pokud míček narazí do vrchní nebo spodní části, tak se odrazí a jde dál. V případě, že hráč míček neodrazí, a ten proletí, tak protivník získává bod.

# 3 Vlastní práce

## 3.1 Struktura

Práce je rozdělena do následujících tříd:

Game

Frame

Panel

Check Collision

Paddle

Ball

Score

## 3.2 Game

Prvně jsem vytvořil třídu Game.java, ve které jsem použil main metodu, do níž jsem zavolal svůj Frame.

public static void main(String[] args) {

GameFrame frame = new GameFrame();

}

## 3.3 Frame

Ve třídě GameFrame.java jsem si udělal JFrame, mající v sobě Panel, jehož velikost je přizpůsobena velikosti panelu pomocí metody pack. Nastavil jsem název na Pong, vypnul jsem možnost změny velikosti framu, zafixoval jsem jeho zobrazení uprostřed pomocí setLocationRelativeTo(null) a zvolil jsem černou barvu.

GameFrame(){

panel = new GamePanel();

this.add(panel);

this.setTitle("Pong");

this.setResizable(false);

this.setBackground(Color.black);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.pack();

this.setVisible(true);

this.setLocationRelativeTo(null);

}

## 3.4 Panel

Po té jsem si vytvořil JPanel, do kterého jsem vložil funkci Runnable, díky které se program stále opakuje. V konstruktoru jsem si zavolal metodu newBall a newPaddle, díky kterým vykreslím míček a pálky. Pak jsem nechal vykreslit skóre. Nastavil jsem focusable na true, pomocí kterého se snímá stlačení kláves. Po té jsem přidal třídu KL, která umožňuje odezírání z klávesnice. A nakonec jsem zapnul thread, jež zajišťuje, že program dělá více věcí naráz.

GamePanel(){

newPaddle();

newBall();

score = new Score(GAME\_WIDTH, GAME\_HEIGHT);

this.setFocusable(true);

this.addKeyListener(new KL());

this.setPreferredSize(SCREEN\_SIZE);

thread = new Thread(this);

thread.start();

}

### 3.4.1 Check Collision

Bylo třeba zabezpečit, aby se pálky pohybovaly po stranách shora dolů, a aby nezmizely z hracího pole. Toho jsem dosáhl tím, že pokud pálka byla na okraji, tak jsem její y pozici nastavil znovu na 0 v horní pozici, a na GAME\_HEIGHT v dolní pozici.

if(paddle1.y <= 0){

paddle1.y = 0;

}

if(paddle1.y >= (GAME\_HEIGHT - PADDLE\_HEIGHT)){

paddle1.y = GAME\_HEIGHT - PADDLE\_HEIGHT;

}

if(paddle2.y <= 0){

paddle2.y = 0;

}

if(paddle2.y >= (GAME\_HEIGHT - PADDLE\_HEIGHT)){

paddle2.y = GAME\_HEIGHT - PADDLE\_HEIGHT;

}

Další věc, kterou jsem řešil, byl míček a jeho odrážení od horních okrajů, od pálek a také připisování bodů, když je míček na postranních okrajích. Pomocí podmíněného prohlášení if jsem nastavil:

1. Pokud míček dosáhne horního okraje, tak se jeho y rychlost obrátí.

if(ball.y <= 0){

ball.setYDirection(-ball.yVelocity);

}

1. Pokud míček dosáhne spodního okraje, tak se jeho y rychlost obrátí.

if(ball.y >= GAME\_HEIGHT - BALL\_DIAMETER){

ball.setYDirection(-ball.yVelocity);

}

1. Odražení míčku od pálek. Pokud dojde ke kontaktu míčku a pálky, míček změní trajektorii na opačnou stranu a zároveň se o trochu zrychlí.

if(ball.intersects(paddle2)){

ball.xVelocity = Math.abs(ball.xVelocity);

ball.xVelocity++;

if(ball.yVelocity > 0){

ball.yVelocity++;

}

else{

ball.yVelocity--;

}

ball.setXDirection(-ball.xVelocity);

ball.setYDirection(ball.yVelocity);

}

1. Posledním úkolem k řešení bylo přičítat body hráčům. Pokud se y pozice míčku rovnala 0, tak se přičetl bod hráči 2, a pokud se rovnala GAME\_WIDTH, tak se bod přičetl hráči 1. Poté se vykreslí nové pálky a míček uprostřed pole.

if(ball.x <= 0){

score.player2++;

newPaddle();

newBall();

}

if(ball.x >= GAME\_WIDTH - BALL\_DIAMETER){

score.player1++;

newPaddle();

newBall();

}

## 

## 3.5 Paddle

Ve třídě Paddle jsem v konstruktoru nastavil parametry toho, na jaké pozici a jak velká se pálka vykreslí. Každé pálce jsem přiřadil její id pro hráče 1 a 2.

Poté jsem použil metodu keyPressed, a v té jsem vytvořil switch podle id. V case 1(levá pálka) jsem nastavil podmínku - pokud hráč zmáčkne klávesu W, tak se nastaví záporná y rychlost, pálka jede nahoru. Pokud hráč zmáčkne klávesu S, tak y rychlost bude kladná a pálka jede dolu. To samé jsem udělal pro druhou pálku s použitím šipky nahoru a dolů.

public void keyPressed(KeyEvent e){

switch(id){

case 1:

if(e.getKeyCode() == KeyEvent.VK\_W){

setYDirection(-speed);

move();

}

if(e.getKeyCode() == KeyEvent.VK\_S){

setYDirection(speed);

move();

}

break;

Stejný postup jsem opakoval pro keyReleased, kde se y rychlost nastaví na 0, aby se pálka po uvolnění klávesy zastavila.

public void keyReleased(KeyEvent e){

switch(id){

case 1:

if(e.getKeyCode() == KeyEvent.VK\_W){

setYDirection(0);

move();

}

if(e.getKeyCode() == KeyEvent.VK\_S){

setYDirection(0);

move();

}

break;

V metodě draw jsem nastavil podmínku - pokud se id = 1, tak bude vybarvena modře, jinak se vybarví červeně. A vyplnil jsem obdélník.

public void draw(Graphics g){

if (id == 1) {

g.setColor(Color.red);

}

else{

g.setColor(Color.blue);

}

g.fillRect(x, y, width, height);

}

## 

## 3.6 Ball

Ve třídě Ball jsem v konstruktoru nadefinoval parametry toho, na jaké pozici a jak velký se míček vykreslí. Dále jsem zvolil náhodnou x a y rychlost.

Ball(int x, int y, int width, int height){

super(x, y, width, height);

random = new Random();

int randomXDirection = random.nextInt(2);

if(randomXDirection == 0){

randomXDirection--;

}

setXDirection(randomXDirection \* speed);

int randomYDirection = random.nextInt(2);

if(randomYDirection == 0){

randomYDirection--;

}

setYDirection(randomYDirection \* speed);

}

## 3.7 Score

Ve třídě Score jsem vytvořil konstruktor, do kterého jsem zadal parametr GAME\_WIDTH a GAME\_HEIGHT .

Score( int GAME\_WIDTH, int GAME\_HEIGHT){

Score.GAME\_WIDTH = GAME\_WIDTH;

Score.GAME\_HEIGHT = GAME\_HEIGHT;

}

V metodě draw jsem nastavil barvu, styl a velikost písma. Vykreslil jsem čáru uprostřed hry a počet bodů každého hráče.

public void draw(Graphics g){

g.setColor(Color.white);

g.setFont(new Font("Consolas", Font.PLAIN, 60));

g.drawLine((GAME\_WIDTH / 2), 0, (GAME\_WIDTH / 2), GAME\_HEIGHT);

g.drawString(String.valueOf(player1), (GAME\_WIDTH / 2) - 85, 50);

g.drawString(String.valueOf(player2), (GAME\_WIDTH / 2) + 50, 50);

}

# 4 Závěr

Při práci na ročníkovém projektu jsem se naučil používat knihovny Java.awt a Java.swing, a celkově jsem výrazně rozšířil svoji znalost programovacího jazyku Java. Návod k práci jsem čerpal z veřejně dostupných zdrojů, především lekcí na itnetwork a youtube, a osobní konzultací s IT specialistou. Poprvé mám možnost vidět ucelený výsledek své programátorské práce.

Myslím si, že se mi ročníkový projekt celkem povedl. Do budoucna bych rád omezil dobu hraní.

# 5 Zdroje

[**https://www.itnetwork.cz/cs/java/swing/hry/tutorial-java-gui-hry-dil01/**](https://www.itnetwork.cz/cs/java/swing/hry/tutorial-java-gui-hry-dil01/)

[**https://www.itnetwork.cz/**](https://www.itnetwork.cz/)

[**https://www.w3schools.com/java/¨**](https://www.w3schools.com/java/¨)

[**https://www.youtube.com/watch?v=r59xYe3Vyks&list=PLS1QulWo1RIbfTjQvTdj8Y6yyq4R7g-Al**](https://www.youtube.com/watch?v=r59xYe3Vyks&list=PLS1QulWo1RIbfTjQvTdj8Y6yyq4R7g-Al)

**osobní konzultace s Karlem Bartošem (IT specialista)**