JSR 184 - tutorial 2

Transformace v JSR-184

Cíl tutorialu: - naučit vás měnit pozici/ rotaci/ velikost objektu

Tak jdeme na to:

V minulém tutorialu (Úvod do JSR-184) jsme si ukázali, jak námi vytvořenou scénu ve formátu *.m3g můžeme načíst do virtuálního světa, ukázali jsme si, jak z tohoto světa vybereme 1 objekt a řekli jsme si, jak naší scénu vykreslíme na display. V dnešním tutorialu máme na programu si říct něco málo o tom, jakými způsoby můžeme měnit polohu, rotaci i velikost podle všech os (X/ Y/ Z). Budeme pokračovat v kódu, který jsme si napsali minule, ale pro běh aplikace, kde se něco hýbe (což je tedy většina), je moudřejší mít k dispozici vlákno. To znamená, že si na začátku kódu (hned za „extends Canvas“) napíšeme „implements Runnable“:

```
public class packman3DCanvas extends Canvas implements Runnable
{
```

Stejně, jako když rozšiřujete Canvas, musí vaše třída obsahovat metodu „paint(Graphics g)“, musí třída implementující Runnable obsahovat metodu „run()“, která se sama volá během běhu aplikace (zatím necháme prázdnou):

```
public void run()
{
}
```

Také si na začátku aplikace musíme deklarovat vlákno s počáteční nulovou hodnotou, které si pojmenujeme „myThread“:

```
Thread myThread = null;
```

Ještě budeme potřebovat něco, čím budeme pozastavovat hru atp. Na to je samozřejmě nejvhodnější proměnná typu „boolean“ (zpočátku s hodnotou false) a pojmenujeme si ji „running“:

```
boolean running = false;
```

V konstruktoru třídy ještě vytvoříme nové vlákno, nastavíme proměnnou kontrolující běh aplikace na true a zapneme běh vlákna:

```
myThread = new Thread(this);
running = true;
myThread.start();
```

A teď už se konečně dostáváme k tématu tohoto tutorialu. Nejdříve si představíme základní metody, pomocí kterých měníme polohu, rotaci a velikost objektů v JSR-184 a poté za pomoci některých z nich naučíme našeho (zatím statického) packmana pohybu. Všechny metody, které nyní budu popisovat jsou z třídy „javax.microedition.m3g.Transformable“.

Poloha

Nejzákladnějším příkazem na změnu polohy nějakého objektu (nejenom Meshe) je „setTranslation(float x, float y, float z)“. Jistě jste ho všichni pochopili… Je to metoda se 3 parametry, které udávají polohu v každém rozměru našeho světa.

Velmi blízkou metodou k metodě „setTranslation“ je metoda „translate(float x, float y, float z)“. Rozdíl mezi touto a předchozí metodou je v tom, že metoda „setTranslation“ umístí daný objekt přímo na zadané souřadnice, zatímco metoda „translate“ POSUNE daný objekt o určenou vzdálenost podle jednotlivých os (zadanou v parametrech).

Poloha určeného objektu se dá zjistit pomocí příkazu „getTranslation(float[] xyz)“, která vrací pole 3 čísel typu float (x, y, z).

Rotace

U rotace (natočení) objektu je takovým základním příkazem „setOrientation(float angle, float ax, float ay, float az)“. První parametr udává úhel (udávaný ve stupních; po směru hodinových ručiček), zbylé parametry udávají, kolikanásobnou hodnotu má úhel mít podle určené osy.

Další možností, jak změnit rotaci objektu je příkaz „preRotate(float angle, float ax, float ay, float az)“. Mezi příkazy „setOrientation“ a „preRotate“ je to stejné jako mezi „setTranslation“ a „translate“. U „setOrientation“ se zadává, jak se má daný objekt natočit, „preRotate“ znamená, O KOLIK se má daný objekt natočit. Ale pozor!!! Nepleťte si tento příkaz s následujícím. Proč, to si vysvětlíme přímo u něho.

Nyní se dostáváme k příkazu „postRotate(float angle, float ax, float ay, float az)“. Jeho parametry jsou stejné jako u předchozích dvou příkazů. Tento příkaz se však často plete s příkazem zmiňovaným dříve, s příkazem „preRotate“. Řeknu to asi takto: Každý svět má své osy, od kterých se určuje vzdálenost objektu apod. Ale stejně tak má i každý objekt své osy (jejich průsečík se bere jako poloha objektu). A právě v tom spočívá rozdíl příkazů „postRotate“ a „preRotate“. Používáme-li „preRotate“, objekt se otáčí kolem vlastních os, zatímco při „postRotate“ se objekt otáčí kolem globálních os.

Nakonec se dostáváme k tomu, jak zjistit orientaci objektu. To se dělá téměř stejně jako u zjištění polohy. Jen na zjištění orientace objektu nám k tomu slouží příkaz „getOrientation(float[] angleAxis)“, který vrací pole 4 čísel typu float (angle, ax, ay, az).

Velikost

Poslední věc, kterou se bude zabývat tento tutorial je změna velikosti objektů. Pro změnu velikosti se používají příkazy „setScale(float sx, float sy, float sz)“ a „scale(float sx, float sy, float sz)“. Mezi těmito příkazy je to stejné jako mezi příkazy „setTranslation“ a „translate“. U příkazu „setScale“ se opět jedná o určení hodnoty, na kterou se má objekt zvětšit (zmenšit), zatímco u příkazu „scale“ se zadávají hodnoty O KOLIK se má objekt zvětšit (zmenšit). Počáteční velikost je 1.0f, 0.5f je tedy velikost poloviční a 2.0f dvojnásobná. Pro lepší představu si tato čísla můžete vynásobit stem a pracovat s nimi jako s procenty.

Samozřejmě i u velikosti můžeme zjistit, jak je objekt veliký oproti původní hodnotě a to příkazem „getScale(float[] xyz)“ který vrací pole 3 čísel typu float (sx, sy, sz).

Zpět k našemu packmanovi

Když už jsme si vysvětlily, jak měnit polohu, rotaci a velikost objektů, můžeme si to vyzkoušet v praxi. Budeme sice používat jen velmi málo z dnes představených metod, vy si však můžete samostatně vyzkoušet všechny. Chtěl bych s vámi docílit toho, aby se náš packman pohyboval po světě, aby při zmáčknutí čísla 4 (kurzoru vlevo) zahnul o 90° vlevo (o –90°), při zmáčknutí čísla 6 (kurzoru vpravo) zahnul o 90° vpravo, při 1 zmáčknutí čísla 8 (kurzoru dolu) zastavil a při druhém se otočil o 180° a dále pokračoval opačným směrem než předtím popřípadě stejným směrem po zmáčknutí čísla 2 (kurzoru nahoru). To znamená, že budeme potřebovat metodu starající se o to, nebyla-li stisknuta nějaká klávesa. Předpokládám, že když se pouštíte do 3D javy, tak jste se s tímto už setkali. A rovnou si tuto metodu předpřipravíme pro budoucí akce, u kterých budeme potřebovat čísla 2, 4, 6, 8, resp. Tlačítka UP, LEFT, RIGHT, DOWN. Nyní to bude vypadat asi takto:

```
protected void keyPressed(int keyCode)
{
	if(keyCode == KEY_NUM4 || keyCode == getKeyCode(Canvas.LEFT))
	{
	}
	else if(keyCode == KEY_NUM6 || keyCode == getKeyCode(Canvas.RIGHT))
	{
	}
	else if(keyCode == KEY_NUM8 || keyCode == getKeyCode(Canvas.DOWN))
	{
	}
	else if(keyCode == KEY_NUM2 || keyCode == getKeyCode(Canvas.UP))
	{
	}
}
```

Také budeme potřebovat nějakou proměnnou, která nám bude říkat, jakým směrem se zrovna náš packman pohybuje. Já na takovéto věci používám proměnné typu int, za které si jako komentář pro jistotu píši, jaká hodnota co znamená. Ostatně tohoto „intu“ využijeme i později pro pohyb… Tuto proměnnou si nazveme „smerPackmana“ (mám raději delší a přehlednější názvy) a ze začátku jí přiřadíme hodnotu 30. Hodnoty větší než 4 budou znamenat, že se packman nepohybuje a budou to hodnoty 10, 20, 30 a 40 (to kvůli tomu, abychom věděli, jakým směrem se packman pohyboval před zastavením), hodnota 1 že se pohybuje tak, že mu klesá pozice v ose X, hodnota 2, že mu přibývá hodnota v ose Y, hodnota 3, že mu přibývá hodnota v ose X a hodnota 4, že mu ubývá hodnota v ose Y. Takovéto věci si zapisuji takto:

```
int smerPackmana = 30;  // 1 = x-, 2 = y+, 3 = x+, 4 = y-, >4 = stoji
```

Možná jste zvyklí, že osa Y udává výšku. Na to jsem byl z jiných modelačních programů také zvyklý, ovšem v 3ds max je tomu tak, že výškou je osa Z. Dá se na to zvyknout a já jsem si už zvykl, takže to budu psát takto. Snad vám to nebude dělat moc velké problémy.

Dále si vytvoříme proměnnou s rychlostí packmana, abychom ji mohli kdykoli snadno změnit. Bude typu int, pojmenujeme si ji „rychlostPackmana“ a nastavíme ji na 1 (zatím).

```
int rychlostPackmana = 1;
```

Teď si uděláme dvojrozměrné pole typu int, které nám v budoucnu ulehčí život, protože nebudeme muset vypisovat spoustu podmínek. Nazveme si ho „smeryPohybu“ a budou v něm 4 další pole (protože máme 4 směry) a v každém tomto poli 3 hodnoty (x, y, z) podle daného směru (jak jsme si říkali že např. smerPackmana 1 je s ubývající hodnotou X, tak to zde také napíšeme):

```
int[][] smeryPohybu = new int[][] {{-1, 0, 0}, {0, 1, 0}, {1, 0, 0}, {0, -
1, 0}};
```

Nyní si poupravíme na začátku zmiňovanou metodu „run“ tak, aby se volaly potřebné metody pouze pokud naše hra běží (když je „running“ true) a aby na rychlejších zařízeních neběžela hra příliš rychle, „uspíme“ vlákno na 40 milisekund (což by pak mělo být 25 snímků za sekundu – akorát plynulá a ne přehnaně rychlá hra) – „Thread.sleep(40)“:

```
public void run()
{
       while(running) {
            pohybPackmana();
            try {
                  Thread.sleep(40);
            } catch(Exception e) {
                  System.out.println(e);
            }
      }
}
```

Jak vidíte, volal jsem metodu pohybPackmana, kterou si teď představíme. Nenachází se v ní nic moc, ale stará se to o pohyb packmana ve všech směrech:

```
public void pohybPackmana()
{
	if(smerPackmana >= 1 && smerPackmana <= 4 )
	{
		packman.translate(smeryPohybu[smerPackmana-
1][0]*rychlostPackmana, smeryPohybu[smerPackmana-1][1]*rychlostPackmana,
smeryPohybu[smerPackmana-1][2]*rychlostPackmana);
	}
}
```

V této metodě jsme použili metodu „translate“, kterou jsme se naučili výše. V každém parametru této metody jsme vyvolali hodnotu z dvojrozměrného pole, s prvním podpolem odpovídajícímu směru packmana a indexem odpovídajícímu buď ose X, Y nebo Z, a nakonec jsem tuto hodnotu vynásobil rychlostí packmana.

Jako poslední věc, kterou ještě musíme udělat je při zmáčknutí některé klávesy změnit směr packmana, zastavit ho, opět ho rozjet určitým směrem atp. To zde nebudu popisovat, jelikož je to moc zdlouhavé a ten, kdo si ten kód přečte to snad pochopí (není tam snad nic nového). Jen jsem tam párkrát použil metodu „preRotate“, kterou jsme si popisovali zezačátku.

```
protected void keyPressed(int keyCode)
{
	if(keyCode == KEY_NUM4 || keyCode == getKeyCode(Canvas.LEFT))
	{
		if(smerPackmana <= 4)
		{
			smerPackmana = smerPackmana>1 ? smerPackmana-1 : 4;
		}
		else
		{
			smerPackmana = smerPackmana==10 ? 4 : (smerPackmana-
10)/10 ;
		}

		packman.preRotate(-90, 0, 0, 1);
	}
	else if(keyCode == KEY_NUM6 || keyCode == getKeyCode(Canvas.RIGHT))
	{
		if(smerPackmana <= 4)
		{
			smerPackmana = smerPackmana<4 ? smerPackmana+1 : 1;
		}
		else
		{
			smerPackmana = smerPackmana==40 ? 1 :
(smerPackmana+10)/10 ;
		}

		packman.preRotate(90, 0, 0, 1);
	}
	else if(keyCode == KEY_NUM8 || keyCode == getKeyCode(Canvas.DOWN))
	{
		if(smerPackmana <= 4)
		{
			smerPackmana *= 10;
```

```
			}
			else
			{
				switch(smerPackmana)
				{
					case 10:
						smerPackmana = 3;
					break;

					case 20:
						smerPackmana = 4;
					break;

					case 30:
						smerPackmana = 1;
					break;

					case 40:
						smerPackmana = 2;
					break;
				}
			}

			packman.preRotate(180, 0, 0, 1);
		}
		else if(keyCode == KEY_NUM2 || keyCode == getKeyCode(Canvas.UP))
		{
			smerPackmana = smerPackmana>4 ? smerPackmana/10 : smerPackmana;
		}
}
```

Po dnešním tutorialu by měl kód naší třídy „packmanCanvas3D“ vypadat nějak takto:

```
import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Graphics;
import javax.microedition.m3g.World;
import javax.microedition.m3g.Camera;
import javax.microedition.m3g.Graphics3D;
import javax.microedition.m3g.Loader;
import javax.microedition.m3g.Mesh;
import javax.microedition.m3g.Object3D;

public class packman3DCanvas extends Canvas implements Runnable
{
        Thread myThread = null;
    boolean running = false;

        Graphics3D g3d = null;

        World world = null;
        Mesh packman = null;
        Camera cam = null;

        int packmanID = 27;

        int smerPackmana = 30;   // 1 = x-, 2 = y+, 3 = x+, 4 = y-, >4 = stoji
        int rychlostPackmana = 1;
        int[][] smeryPohybu = new int[][] {{-1, 0, 0}, {0, 1, 0}, {1, 0, 0}, {0, -1, 0}};

        public packman3DCanvas()
        {
                setFullScreenMode(true);
                loadWorld("/packman.m3g");
                loadObjects();

                myThread = new Thread(this);
       running = true;
                myThread.start();
        }

        public void run()
        {
       while(running) {
```

```
                        pohybPackmana();
                        try {
                                Thread.sleep(40);
                        } catch(Exception e) {
                                System.out.println(e);
                        }
                }
        }

        public void loadWorld(String path)
        {
                try
                {
                        Object3D[] buffer = Loader.load(path);
                        for(int i = 0; i < buffer.length; i++)
                        {
                                if(buffer[i] instanceof World)
                                {
                                        world = (World)buffer[i];
                                        break;
                                }
                        }
                }
                catch (Exception e)
                {
                        System.out.println(e);
                }
        }

        public void loadObjects()
        {
                cam = world.getActiveCamera();
                packman = (Mesh)world.find(packmanID);
        }

        public void pohybPackmana()
        {
                if(smerPackmana >= 1 && smerPackmana <= 4 )
                {
                        packman.translate(smeryPohybu[smerPackmana-1][0]*rychlostPackmana,
smeryPohybu[smerPackmana-1][1]*rychlostPackmana, smeryPohybu[smerPackmana-1][2]*rychlostPackmana);
                }
        }

        public void paint(Graphics g)
        {
                g3d = Graphics3D.getInstance();
                try {
                        g3d.bindTarget(g, true, 0);
                        g3d.render(world);
                } catch(Exception e) {
                        System.out.println(e);
                } finally {
                        g3d.releaseTarget();
                }
                repaint();
        }

        protected void keyPressed(int keyCode)
        {
                if(keyCode == KEY_NUM4 || keyCode == getKeyCode(Canvas.LEFT))
                {
                        if(smerPackmana <= 4)
                        {
                                smerPackmana = smerPackmana>1 ? smerPackmana-1 : 4;
                        }
                        else
                        {
                                smerPackmana = smerPackmana==10 ? 4 : (smerPackmana-10)/10 ;
                        }

                        packman.preRotate(90, 0, 0, 1);
                }
                else if(keyCode == KEY_NUM6 || keyCode == getKeyCode(Canvas.RIGHT))
                {
                        if(smerPackmana <= 4)
                        {
                                smerPackmana = smerPackmana<4 ? smerPackmana+1 : 1;
                        }
                        else
                        {
                                smerPackmana = smerPackmana==40 ? 1 : (smerPackmana+10)/10 ;
                        }

                        packman.preRotate(-90, 0, 0, 1);
                }
                else if(keyCode == KEY_NUM8 || keyCode == getKeyCode(Canvas.DOWN))
                {
```

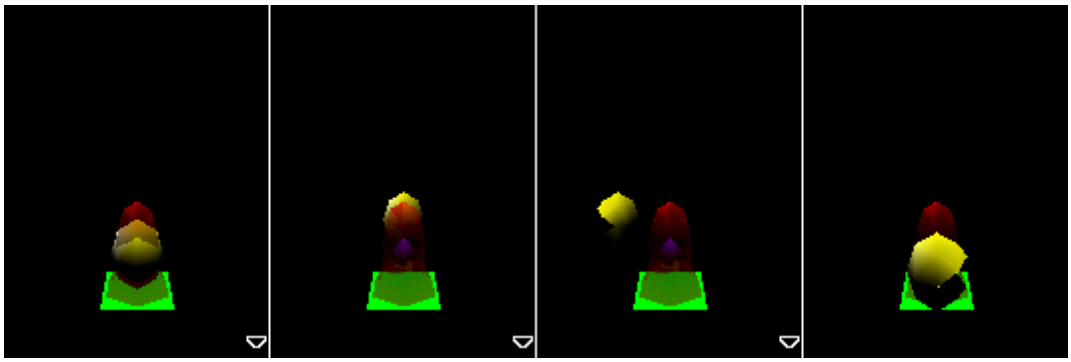
```
                        if(smerPackmana <= 4)
                        {
                                smerPackmana *= 10;
                        }
                        else
                        {
                                switch(smerPackmana)
                                {
                                        case 10:
                                                smerPackmana = 3;
                                        break;

                                        case 20:
                                                smerPackmana = 4;
                                        break;

                                        case 30:
                                                smerPackmana = 1;
                                        break;

                                        case 40:
                                                smerPackmana = 2;
                                        break;
                                }
                                packman.preRotate(180, 0, 0, 1);
                        }
                }
                else if(keyCode == KEY_NUM2 || keyCode == getKeyCode(Canvas.UP))
                {
                        smerPackmana = smerPackmana>4 ? smerPackmana/10 : smerPackmana;
                }
        }
}
```

A při spuštění hry byste měli být schopni se s packmanem pohybovat po (zatím prakticky prázdném) světě podle os X a Y a váš packman by se měl podle směru pohybu i natáčet:



Pokud budete mít nějaké potíže, naleznete v tutorialu chybu, budete mít nějaký nápad na vylepšení, budete se chtít na něco zeptat, budete se chtít pochlubit svou prací, budete se chtít k tutorialu vyjádři (ať už kladně či záporně) nebo cokoli podobného, budu velmi rád, když napíšete do fóra, popřípadě přímo mně na mail nebo na ICQ.

Kompletní zdrojový kód tohoto tutorialu se všemi soubory naleznete ke stažení v sekci Download.