

Fred – Animation

Na to, aby ste vytvorili dojem plynulého pohybu Freda, potrebujete vytvoriť jeho animáciu, pozostávajúcu zo sledu obrázkov. Každý špecializovaný pohyb by mal mať vlastnú animáciu (napr. pohyb vľavo, pohyb vpravo), na to poslúži trieda `Animation`.

```
Animation currentAnimation, leftAnimation, rightAnimation;
```

Je viac spôsobov, ako vytvoriť animáciu postavy v hre:

a) pomocou viacerých obrázkov

- každý z nich reprezentuje jeden krátky úsek z pohybu

b) pomocou jedného obrázka, ktorý je vlastne vyskladaný z obrázkov z možnosti a)

Variant a)

Pre tento účel vytvorte polia obrázkov pre jednotlivé animácie. Na vytvorenie objektu obrázka použite konštruktor

`Image(String ref)`

- parameter **`ref`** je cesta ku obrázku, ktorý chcete použiť
- koľko obrázkov do poľa pridáte, toľko ich bude použitých pri vytvorení animácie, takže čím viac obrázkov, tým plynulejšie bude pohyb vyzerat'

```
Image [] moveLeft = {new Image("path/move_left1.png"),
                      new Image("path/move_left2.png"),
                      ...};
Image [] moveRight = {new Image("path/move_right1.png"),
                      new Image("path/move_right2.png"),
                      ...};
Image [] moveUp = ...

...etc
```

Pozn.: Vo vašom projekte máte k dispozícii obrázky len pre animáciu v jednom smere pohybu. Aby ste nemuseli vytvárať ďalšie prevrátené kópie, použite na vytvorených objektoch obrázkov metódu

```
getFlippedCopy(boolean flipHorizontal, boolean flipVertical)
```

Zatiaľ si postačíte s animáciami pre pohyb vľavo a vpravo.

Pre vytvorenie animácií použite konštruktor

`Animation(Image [] frames, int duration, boolean autoUpdate)`.

- parameter **frames** je pole obrázkov - rámečkov, z ktorých má animácia pozostávať
- parameter **duration** udáva trvanie vykreslenia jedného obrázka z animácie na obrazovku, v milisekundách
- parameter **autoUpdate** udáva, či sa má animácia obnovovať automaticky alebo nie (vyskúšajte, ako sa animácia bude správať, ak tento parameter nastavíte na true a potom na false)

napr.:

```
leftAnimation = new Animation(moveLeft, 100, false);
```

Ak ste tak ešte neurobili, nezabudnite pridať potrebné importy:

```
import org.newdawn.slick.Animation;
import org.newdawn.slick.Image;
```

Variant b)

V tomto prípade budete potrebovať jeden obrázok vyskladaný zo všetkých častí (rámečkov), z ktorých má animácia pozostávať. Príkladom takého obrázka je **fred.png**, nachádzajúci sa v priečinku *resources/tiled* vášho projektu.

Vytvorte inštanciu triedy `SpriteSheet`, použite pri tom konštruktor

```
SpriteSheet(String ref, int tw, int th)
```

ktorý

- prvý parameter **ref** typu `String` udáva prístupovú cestu k obrázku, bude použitý
- parameter **tw** udáva šírku jedného rámečka animácie v pixloch
- parameter **th** udáva výšku jedného rámečka animácie v pixloch
- zadaný obrázok bude vlastne neskôr rozdelený na rámečky rozmerov `tw` a `th`, ktoré sa budú pri animácii vykresľovať na obrazovku

príklad použitia:

```
SpriteSheet spriteSh = new SpriteSheet("data/fred.png", 104, 170);
```

Nezabudnite pridať potrebný import:

```
import org.newdawn.slick.SpriteSheet;
```

Vytvorte inštanciu triedy `Animation` pre prvý z pohybov. Použite pri tom bezparametrický konštruktor. Teraz je potrebné do vytvorenej animácie pridať rámečky, z ktorých sa bude skladať. použite pri tom tento fragment kódu:

```
leftAnimation = new Animation();

for(int frame = 0; frame < number_of_frames; frame++) {
    leftAnimation.addFrame(spriteSh.getSprite(frame, 0), duration);
}
```

- namiesto **number_of_frames** použite číslo udávajúce počet rámcikov, pre vytváranú animáciu
- ako **duration** použite číslo reprezentujúce čas v milisekundách, počas ktorého bude vykreslený jeden rámček animácie na obrazovku (čiže trvanie jedného rámčeka animácie)

- - -

Teraz, keď máte vytvorené animácie pre Fredov pohyb, je potrebné ich patrične použiť. Pri inicializácii Freda nastavte ako jeho aktuálnu animáciu jednu z vytvorených animácií. To bude jeho východisková animácia (udáva aj to, na ktorú stranu bude Fred na začiatku otočený).

Animácia by sa mala obnoviť vždy s vykonaním nejakého pohybu Freda, čiže ak bol stlačený kláves pre pohyb vľavo, ako aktuálnu animáciu (`currentAnimation`) nastavte patričnú Fredovu animáciu (`leftAnimation`).

- Ak ste pri vytváraní animácie zvolili variantu **a)** a nastavili **autoUpdate** na hodnotu `true`, ďalšie obnovovanie animácie nebude potrebné.
- V opačnom prípade je ešte potrebné zavolať nad patričnou inštanciou triedy `Animation` metódu

```
update(long delta)
```

pričom parameter **delta** je množstvo času, ktorý ubehol od posledného obnovenia (`update-u`). **delta** (možno pod iným názvom) je jedným z parametrov metódy `update` triedy `BasicGame`, takže nie je problém ho získať a použiť

Ako posledné je potrebné spraviť rendrovanie (vykreslenie) animácie v metóde `render()`.

- Ak pri animáciách máte nastavený **autoUpdate** na `true` alebo ste postupovali pomocou variantu **b)**, na objekte triedy `Graphics` zavolajte metódu

```
drawAnimation(Animation anim, float x, float y),
```

kde **anim** je aktuálna animácia, ktorú chcete vykresliť, **x** a **y** udávajú súradnice pozície, na ktorú chcete animáciu vykresliť.

- Ak animácie nevyužívajú `autoUpdate`, na objekte aktuálnej animácie zavolajte metódu

```
draw(float x, float y),
```

kde **x** a **y** udávajú súradnice pozície, na ktorú chcete animáciu vykresliť.

K dispozícii je viacero konštruktorov triedy `Animation`. Vyskúšajte aj ďalšie z nich, prípadne meňte hodnoty ich parametrov, aby ste videli, ako sa v závislosti od nich animácie menia.