

## DUM č. 14 v sadě

### 28. Inf-4 Jednoduchá hra Had ve Flashi (ActionScript)

Autor: Robert Havlásek

Datum: 07.06.2013

Ročník: 5AV

Anotace DUMu: Migrace kódu generování jablka do zvláštní funkce. Sněžení jablka hadem. Přidání další části těla hada.

Materiály jsou určeny pro bezplatné používání pro potřeby výuky a vzdělávání na všech typech škol a školských zařízení. Jakékoliv další využití podléhá autorskému zákonu.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## Migrace generování jablka do zvláštní funkce

V minulém DUMu jsme vysvětlili teorii k funkcím, v tomto DUMu ji využijeme – vyrobíme univerzální funkci, která na základě získaných parametrů bude jablko v ploše tvořit.

*Pedagogická poznámka: Migrace kusů kódu do samostatné funkce je vždycky nepříjemná, studenti ztrácejí přehled, kde co mají. Je vhodné nejprve jim celý postup ukázat na dataprojektoru a teprve pak je nechat jej zopakovat.*

Kód na vlastní tvorbu jablka (počínaje `//TVORBA JABLKA`, konče `} while (obsazeno==1);`) vyjmeme (CTRL-X) z místa, kde se vyskytuje, a namísto něj napíšeme jen volání funkce, kterou vyrobíme: `vytvor_jablko("jablko1", 41)`.

Pro vlastní funkci pak začneme psát hlavičku, ideálně před funkcí `drawCircle`:

```
function tvor_jablko(jmeno:String, hloubka:Number):Number {
    // do tohoto místa vložíme (CTRL-V) kód na vlastní tvorbu jablka
    return 0
}
```

Aktuální stav kódu plochy je tedy:

```
var zdix:Array = Array();
var zdiy:Array = Array();
var klonzdi:Array = Array();
// TVORBA POLE ZDI OKOLO:
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(0);} // horni rada
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(190);} // spodni rada
for (i=0; i<=190; i=i+10) {zdix.push(0);zdiy.push(i);} // leva rada
for (i=0; i<=190; i=i+10) {zdix.push(310);zdiy.push(i);} // prava rada
// FYZICKE KLONOVANI ZDI OKOLO:
for (k=0; k<zdix.length; k++){
    _root.vzorzdi._x=zdix[k];
    _root.vzorzdi._y=zdiy[k];
    duplicateMovieClip(_root.vzorzdi,"klonzdi["+k+"]",50+getNextHighestDepth());}
// NAHODNE ZDI A JEJICH KONTROLA:
for (k=0; k<20; k=k+1){
    _root.vzorzdi._x=10+10*random(30);
    _root.vzorzdi._y=10+10*random(18);
    // KONTROLA OBSAZENOSTI:
    var obsazeno=0;
    for (m=0; m<zdix.length; m=m+1)
        if ((_root.vzorzdi._x==zdix[m])&&(_root.vzorzdi._y==zdiy[m]))
            {obsazeno=1;break;}
    if ( (obsazeno==1) or
        ((_root.vzorzdi._x==160)&&(_root.vzorzdi._y==100)) ) {k=k-1}
    else {zdix.push(_root.vzorzdi._x);
        zdiy.push(_root.vzorzdi._y);
        duplicateMovieClip(_root.vzorzdi,"klonzdi["+zdix.length-1+"]",getNextHighestDepth());}
}
tvor_jablko("jablko1",41);

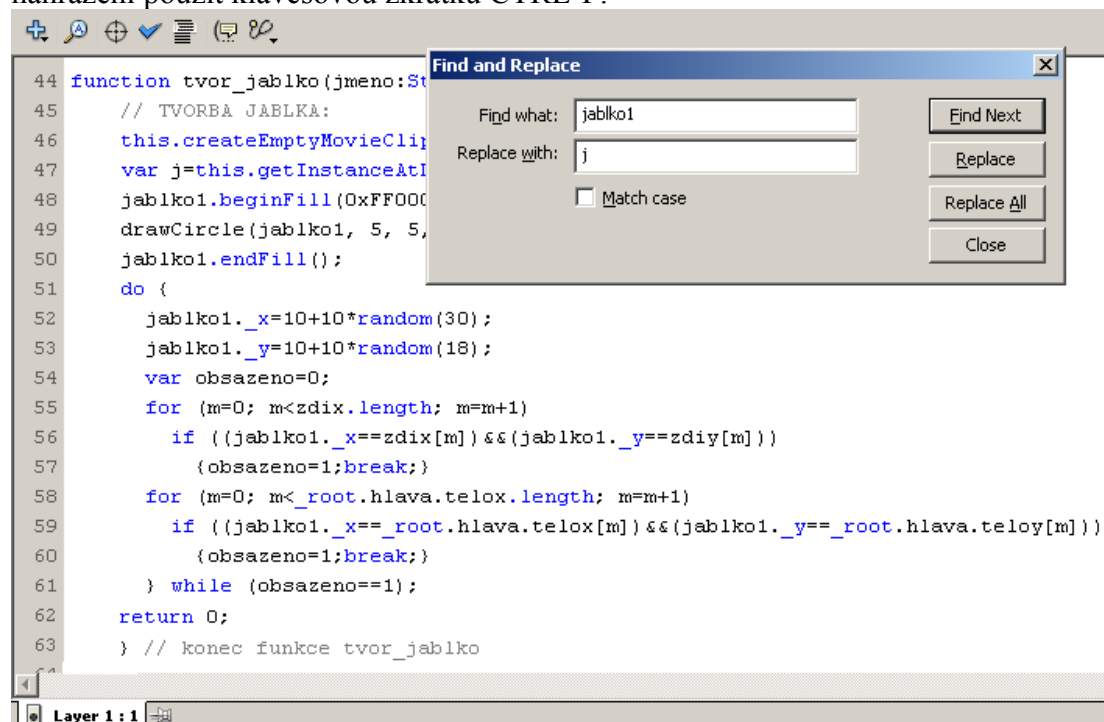
function tvor_jablko(jmeno:String, hloubka:Number):Number {
    // do tohoto místa vložíme (CTRL-V) kód na vlastní tvorbu jablka
    return 0
}

function drawCircle ...
```

Do těla funkce tvor\_jablko vložíme zpět kód, který máme ve schránce, předsadíme jej mezerami (aby logicky vypadal tak, že je uvnitř funkce):

```
function tvor_jablko(jmeno:String, hloubka:Number):Number {
    // TVORBA JABLKA:
    this.createEmptyMovieClip("jablko1", 41);
    jablko1.beginFill(0xFF0000);
    drawCircle(jablko1, 5, 5, 5);
    jablko1.endFill();
    do {
        jablko1._x=10+10*random(30);
        jablko1._y=10+10*random(18);
        var obsazeno=0;
        for (m=0; m<zdix.length; m=m+1)
            if ((jablko1._x==zdix[m])&&(jablko1._y==zdiy[m]))
                {obsazeno=1;break;}
        for (m=0; m<_root.hlava.telox.length; m=m+1)
            if ((jablko1._x==_root.hlava.telox[m])&&(jablko1._y==_root.hlava.teloy[m]))
                {obsazeno=1;break;}
        } while (obsazeno==1);
    return 0
}
```

Zatímco dosud bylo v kódu pro generování jablka přímo napsáno jméno instance, která má vzniknout a též hloubka, kde má vzniknout. Nově tyto informace budeme předávat do parametrů funkce, je tedy nutno text "jablko1" nahradit za jmeno a číslo 41 za hloubka. Aby nebyla situace tak jednoduchá, v dalších řádcích nepoužíváme "jablko1" (jako string), ale jablko1, což se chápe jako jméno instance. Zavedeme tedy novou lokální proměnnou j a do ní tu instanci necháme přiřadit, např. příkazem var j=this.getInstanceAtDepth(hloubka); Namísto proměnné jablko1 budeme ve funkci nadále používat proměnnou j. Doporučuji pro nahrazení použít klávesovou zkratku CTRL-F:



*Pedagogická poznámka: Čistě teoreticky jsme mohli místo j zavést lokální proměnnou jablko1 a nic neměnit, ale dle mého názoru by to bylo matoucí – mít proměnnou uvnitř funkce stejného jména, jako je hodnota parametru...*

Celá funkce bude tedy:

```
function tvor_jablko(jmeno:String, hloubka:Number):Number {
  // TVORBA JABLKA:
  this.createEmptyMovieClip(jmeno, hloubka);
  var j=this.getInstanceAtDepth(hloubka);
  j.beginFill(0xFF0000);
  drawCircle(j, 5, 5, 5);
  j.endFill();
  do {
    j._x=10+10*random(30);
    j._y=10+10*random(18);
    var obsazeno=0;
    for (m=0; m<zdix.length; m=m+1)
      if ((j._x==zdix[m])&&(j._y==zdiy[m]))
        {obsazeno=1;break;}
    for (m=0; m<_root.hlava.telox.length; m=m+1)
      if ((j._x==_root.hlava.telox[m])&&(j._y==_root.hlava.teloy[m]))
        {obsazeno=1;break;}
    } while (obsazeno==1);
  return 0
} // konec funkce tvor_jablko
```

## Sníme jablko

Vyřešme nyní situaci, kdy had (svojí hlavou) do místa jablka přijede. Měl by jej „sníst“. Důsledkem toho by se mělo objevit jablko někde jinde. Jiným důsledkem (vyřešeným později) bude zvětšení délky hada o jeden díl těla.

Kód pro sněžení jablka hlavou budeme psát do stejného místa, kde řešíme narážení hlavy do zdi a do sebe sama. Nejlépe tedy před řádek // KONTROLA NARAZENI HLAVY DO ZDI:

```
if ((_root.hlava._x==_root.jablko1._x)&&(_root.hlava._y==_root.jablko1._y))
  {_root.tvor_jablko("jablko1",41); }
```

neboli: Když máme stejnou pozici hlavy jako jablka, vyrob jablko nové, ve stejné hloubce, jakou mělo jablko původní.

*Pedagogická poznámka: Měli-li bychom jablek víc, museli bychom tuto podmínku dělat cyklicky, vždy přiřadit do speciální proměnné objekt, jenž se v dané hloubce nachází a jemu porovnávat souřadnice.*

## Po sněžení jablka zvětšíme délku těla

Přibude další díl těla, na stejné pozici, jako je poslední současný díl. Poslední současný díl má pozici `_root.hlava.telox[p]`, `_root.hlava.teloy[p]`, kde `p` je index posledního těla, tedy `p=_root.hlava.telox.length`. Do pomocné proměnné `p` doporučuji onen index posledního těla uložit dopředu, aby při přidávání nevznikly zmatky, co je vlastně poslední část...

Tedy:

```
if ((_root.hlava._x==_root.jablko1._x)&&(_root.hlava._y==_root.jablko1._y))
  {_root.tvor_jablko("jablko1",41);
  var p=_root.hlava.telox.length;
  _root.hlava.telox.push(_root.hlava.telox[p]);
  _root.hlava.teloy.push(_root.hlava.teloy[p]);
  }
```

*Pedagogická poznámka: Prefix `_root.hlava.` lze vynechávat, protože píšeme do kódu hlavy.*

## Pro přehlednost celý kód plochy:

```
var zdix:Array = Array();
var zdiy:Array = Array();
var klonzdi:Array = Array();
// TVORBA POLE ZDI OKOLO:
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(0);} // horni rada
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(190);} // spodni rada
for (i=0; i<=190; i=i+10) {zdix.push(0);zdiy.push(i);} // leva rada
for (i=0; i<=190; i=i+10) {zdix.push(310);zdiy.push(i);} // prava rada
// FYZICKE KLONOVANI ZDI OKOLO:
for (k=0; k<zdix.length; k++){
    _root.vzorzdi._x=zdix[k];
    _root.vzorzdi._y=zdiy[k];
    duplicateMovieClip(_root.vzorzdi,"klonzdi["+k+"]",50+getNextHighestDepth());}
// NAHODNE ZDI A JEJICH KONTROLA:
for (k=0; k<20; k=k+1){
    _root.vzorzdi._x=10+10*random(30);
    _root.vzorzdi._y=10+10*random(18);
    // KONTROLA OBSAZENOSTI:
    var obsazeno=0;
    for (m=0; m<zdix.length; m=m+1)
        if ((_root.vzorzdi._x==zdix[m])&&(_root.vzorzdi._y==zdiy[m]))
            {obsazeno=1;break;}
    if ( (obsazeno==1) or
        ((_root.vzorzdi._x==160)&&(_root.vzorzdi._y==100)) ) {k=k-1}
    else {zdix.push(_root.vzorzdi._x);
        zdiy.push(_root.vzorzdi._y);
        duplicateMovieClip(_root.vzorzdi,"klonzdi["+zdix.length-1+"]",getNextHighestDepth());}
    }

tvor_jablko("jablko1",41);

function tvor_jablko(jmeno:String, hloubka:Number):Number {
    // TVORBA JABLKA:
    this.createEmptyMovieClip(jmeno, hloubka);
    var j=this.getInstanceAtDepth(hloubka);
    j.beginFill(0xFF0000);
    drawCircle(j, 5, 5, 5);
    j.endFill();
    do {
        j._x=10+10*random(30);
        j._y=10+10*random(18);
        var obsazeno=0;
        for (m=0; m<zdix.length; m=m+1)
            if ((j._x==zdix[m])&&(j._y==zdiy[m]))
                {obsazeno=1;break;}
        for (m=0; m<_root.hlava.telox.length; m=m+1)
            if ((j._x==_root.hlava.telox[m])&&(j._y==_root.hlava.teloy[m]))
                {obsazeno=1;break;}
        } while (obsazeno==1);
    return 0;
} // konec funkce tvor_jablko

function drawCircle(mc:MovieClip, x:Number, y:Number, r:Number):Void {
    mc.moveTo(x+r, y);
    mc.curveTo(r+x, Math.tan(Math.PI/8)*r+y, Math.sin(Math.PI/4)*r+x,
    Math.sin(Math.PI/4)*r+y);
    mc.curveTo(Math.tan(Math.PI/8)*r+x, r+y, x, r+y);
    mc.curveTo(-Math.tan(Math.PI/8)*r+x, r+y, -Math.sin(Math.PI/4)*r+x,
    Math.sin(Math.PI/4)*r+y);
    mc.curveTo(-r+x, Math.tan(Math.PI/8)*r+y, -r+x, y);
```

v ploše:

```
mc.curveTo(-r+x, -Math.tan(Math.PI/8)*r+y, -Math.sin(Math.PI/4)*r+x,
-Math.sin(Math.PI/4)*r+y);
mc.curveTo(-Math.tan(Math.PI/8)*r+x, -r+y, x, -r+y);
mc.curveTo(Math.tan(Math.PI/8)*r+x, -r+y, Math.sin(Math.PI/4)*r+x, -
Math.sin(Math.PI/4)*r+y);
mc.curveTo(r+x, -Math.tan(Math.PI/8)*r+y, r+x, y);
} // konec funkce drawCircle

function smrt() {
    _root.hlava._x=160;
    _root.hlava._y=100;
    _root.hlava.smer=4;
    for (var m=0;m<_root.hlava.telox.length;m++)
        {_root.hlava.telox[m]=_root.hlava._x;
        _root.hlava.teloy[m]=_root.hlava._y;
        }
    }
}
```

A i celý kód hlavy:

```
on (keyPress "<Left>") {smer=2;}
on (keyPress "<Right>") {smer=0;}
on (keyPress "<Up>") {smer=1;}
on (keyPress "<Down>") {smer=3;}

onClipEvent (enterFrame)
{ if (smer==0) {_root.hlava._x = _root.hlava._x + 10;}
  if (smer==1) {_root.hlava._y = _root.hlava._y - 10;}
  if (smer==2) {_root.hlava._x = _root.hlava._x - 10;}
  if (smer==3) {_root.hlava._y = _root.hlava._y + 10;}
  if (_root.hlava._x>=320) {_root.hlava._x=0;}
  if (_root.hlava._y>=200) {_root.hlava._y=0;}
  if (_root.hlava._x<0) {_root.hlava._x=310;}
  if (_root.hlava._y<0) {_root.hlava._y=190;}
  // KONTROLA, ZDA PRAVE NEJIME JABLKO:
  if ((_root.hlava._x==_root.jablkol._x)&&(_root.hlava._y==_root.jabl
kol._y))
    {_root.tvor_jablko("jablko1",41);
    var p=_root.hlava.telox.length;
    _root.hlava.telox.push(_root.hlava.telox[p]);
    _root.hlava.teloy.push(_root.hlava.teloy[p]);
    }
  // KONTROLA NARAZENI HLAVY DO ZDI:
  var obsazeno=0;
  for (m=0; m<_root.zdix.length; m=m+1)
    if ((_root.hlava._x==_root.zdix[m])&&(_root.hlava._y==_root.zdiy[m]))
      {obsazeno=1;break;}
  // A I KONTROLA NARAZENI DO SVEHO TELA (JEN POKUD SE HYBEME):
  if (smer<4) for (m=0; m<_root.hlava.telox.length; m=m+1)
    if ((_root.hlava._x==_root.hlava.telox[m])&&(_root.hlava._y==_root.hla
va.teloy[m]))
      {obsazeno=1;break;}
  if (obsazeno==1) _root.smrt();
/*   if ((_root.hlava._x>=320) || (_root.hlava._y>=200) ||
    (_root.hlava._x<0) || (_root.hlava._y<0)) _root.smrt();*/
  // VYPOCET NOVEHO TELA:
  for (m=telox.length-2;m>=0;m--) {telox[m+1]=telox[m];
    teloy[m+1]=teloy[m];}

  telox[0]=_root.hlava._x;
  teloy[0]=_root.hlava._y;
```

v objektu hlava:

```
// KRESLENI TELA:
for (var m=0;m<telox.length;m++)
    {removeMovieClip("klontelo["+m+"]");
      _root.telovzor._x=telox[m];
      _root.telovzor._y=teloy[m];
      duplicateMovieClip(_root.telovzor,"klontelo["+m+"]",m)
    }
} // konec onClipEvent(enterFrame)

onClipEvent(load) {
  var telox:Array = Array();
  var teloy:Array = Array();
  var defaultnidelka:Number = 4;
  _root.telovzor._x=_root.hlava._x; //posuneme vzor do pozice hlavy
  _root.telovzor._y=_root.hlava._y;
  for (var i=0; i<defaultnidelka; i++) //zduplikujeme (defaultnidelka)-krat
    {telox.push(_root.hlava._x);
      teloy.push(_root.hlava._y);
      duplicateMovieClip(_root.telovzor,"klontelo"+i+"",i)
    } //konec cyklu for
} // konec onClipEvent(load)
```