

DUM č. 20 v sadě

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

Autor: Robert Havlásek

Datum: 12.06.2013

Ročník: 5AV

Anotace DUMu: Flash - příklad: Zavedení jednotlivých levelů hry. Level editor. Framy před začátkem hry a po konci hry.

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

Zavedení jednotlivých levelů hry

Nejlépe samostatně v rámci domácího úkolu nebo o přestávce před hodinou dostanou studenti úkol na papír/tabuli/do obrázku poslaného mailem nakreslit a souřadnicemi popsat levely, které by v naší hře chtěli mít. Při zadávání zdůrazníme rozměry plochy i nutnost mít vše dělitelné deseti.

Kromě základního, řekněme prvního levelu (obsahujícího čtyři rovné zdi okolo a 20 náhodně umístěných cihlíček) můžeme vyrobit spoustu dalších, podle časových možností a podle toho, co vznikne z diskuze. Příklad výsledku diskuze najdete na druhé straně tohoto DUMu.

Level editor

S jednou z dřívějších skupinek jsme (s použitím kusů již hotových kódů) vyrobili i jednoduchý level editor, který zde předkládám k volnému použití, případně k rozšíření. Postačí prázdná plocha, kód si sám vyrobí objekty typu MovieClip. Klikáním na jednotlivá kolečka tato mění barvu a pozice se ukládají do polí zdix, zdiy, jež lze vypsat (trace) kliknutím na kolečko dole.

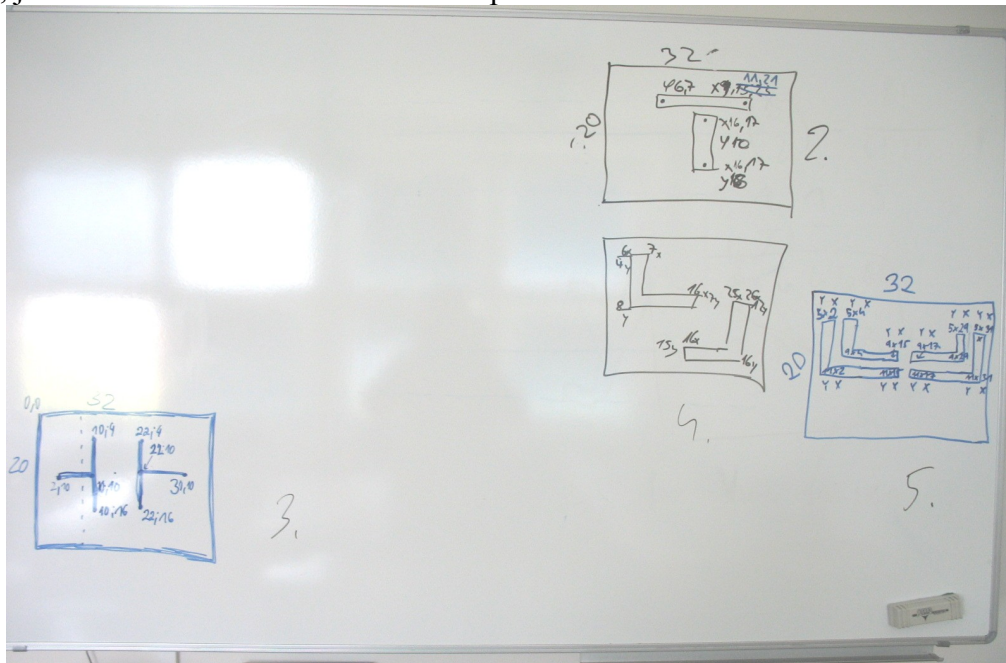
```
var zdix:Array = Array();
var zdiy:Array = Array();
for (i=0;i<32;i++)
  for (j=0;j<20;j++) {
    this.createEmptyMovieClip("jablko"+i+", "+j, 20*i+j);
    jablkoobj=getInstanceAtDepth(20*i+j);
    jablkoobj.beginFill(0x000000);
    drawCircle(jablkoobj, 5, 5, 5);
    jablkoobj._x=10*i;
    jablkoobj._y=10*j;
    jablkoobj.endFill();
    jablkoobj.onPress=function()
      {_root.zdix.push(this._x);_root.zdiy.push(this._y);
        this.beginFill(0xFF0000);
        _root.drawCircle(this, 5, 5, 5);
      }
  }
this.createEmptyMovieClip("vypis", 1000);
vypis.beginFill(0x00FF00);
drawCircle(vypis, 20, 20, 20);
vypis._x=0;
vypis._y=200;
vypis.onPress=function()
  {trace("_root.zdix=Array("+_root.zdix+");");
    trace("_root.zdiy=Array("+_root.zdiy+");");
  }

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);
  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
```

Než zadávat levely klikáním pomocí výše uvedeného level editoru, je dle mého názoru lepší nechat studentům za úkol jednotlivé levely vyrobit pomocí for-cyklů – rozmyslí si, kde jaká zeď začíná, končí, která souřadnice cihliček se v ní mění, ...

Obvykle studenty nechám si levely vymyslet a pak jim řeknu: „A teď si jeden z nich naprogramujte pomocí cyklů“.

Příklad, jak taková diskuze může na tabuli dopadnout:



Implementace levelů

Aktuální stav kódu plochy je ten, že hned na začátku programu nadeklarujeme pole `zdx`, `zdy` a poté do nich rovnou nasypeme data v pasážích `// TVORBA POLE ZDI OKOLO;`, `// FYZICKE KLONOVANI ZDI OKOLO;`, `// NAHODNE ZDI A JEJICH KONTROLA: a` `// KONTROLA OBSAZENOSTI:`. A tato data o zdech v nich (zatím) zůstávají po celou dobu běhu programu.

Tyto čtyři pasáže chápeme jako „tvoření zdí prvního levelu“, odmigrujeme je do speciální funkce `tvor_zdi`, která bude mít navíc parametr `kterylevel: Number`. Bude-li

`kterylevel==1`, zdi se vytvoří podle již naprogramovaných pasáží, bude-li `kterylevel==2` nebo případně vyšší číslo, kód pro tvoření zdí nově naprogramujeme.

Kromě toho napíšeme funkci `vymaz_zdi()`, bez parametrů, která smaže jak proměnné `zdx`, `zdy` (resp. je nastaví na prázdné pole), tak smaže (`removeMovieClip`) objekty na ploše vzniklé klonováním vzorové cihličky.

Na začátku programu, za deklaraci polí `zdx` a `zdy`, zavoláme `tvor_zdi(1)`.

V kódu hlavy zavedeme proměnnou `pocetjablekprolevel` (ideálně v místě, kde zavádíme `pocetjablekprobonus`), přiřadíme do ní vhodnou hodnotu, po kolika jablkách přejde had do dalšího levelu. (Při testování můžeme ponechat třeba `var pocetjablekprolevel: Number = 2;`)

Při sněžení jablka `mj` otestujeme, zda `if ((jablek % pocetjablekprolevel) == 0)`

a když ano, zavoláme `_root.vymaz_zdi()` a `_root.tvor_zdi` s vhodným parametrem, tedy `_root.tvor_zdi(1+Math.floor(jablek/pocetjablekprolevel))`. Při přechodu do

vyššího levelu je též vhodné hada „resetovat“ (pozici, směr, tělo) – to lze „obejít“ zavoláním `this.zivoty++;_root.smrt(this);` Případně při vstupu do nového levelu lze resetovat i délku hada (kód ale není součástí této sady DUMů).

V kódu plochy tedy nově bude:

```
function vymaz_zdi():Void {
  for (k in _root)
    if (_root[k]._name.substr(0,7)=="klonzdi")
      {_root[k].swapDepths(40);
      removeMovieClip(_root[k]);}
  _root.zdix=Array();
  _root.zdiy=Array();
} // of vymaz_zdi
```

Funkce vymaz_zdi využívá cyklus for-in (viz DUM č. 9) a pojmenování jednotlivých klonů zdi jako klonzdi[číslo]. Porovnáním části jména odlišíme zdi od ostatních objektů v ploše.

Před removeMovieClip píšeme swapDepths, který objekt s cihličkou posune do nižší výšky, aby šel příjemněji smazat (viz bug zmiňovaný v DUMu č. 8, kapitola Výmaz klonu).

Na konci funkce vymaz_zdi vidíme vynulování obou polí zdix, zdiy.

A též:

```
function tvor_zdi(kterylevel:Number):Void {
  switch (kterylevel) {
  case 1:
    // 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+"]",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());}
      }
      break; // of case 1

  case 2: // zde bude tvorba druheho levelu
      break; // of case 2

  default: // zde budou vsechny vyssi levely
      break; // of default
  } // of switch
} // function tvor_zdi
```

Popis příkazu switch viz DUM č. 5, při jeho používání je nutno psát za každou variantou „case“ příkaz break;, aby se poté neprováděly příkazy další varianty.

Samostatný frame před hrou

Aby na nás při spuštění vlastní hry nevyskočil rovnou had v bludišti zdí, ale spíše uvítací obrazovka, vyrobíme před vlastním hracím frame ještě jeden, v němž bude uvítání, dotaz na jméno hráče, obtížnost, atp. Studentům obvykle nechávám vybrat, co by čekali, že se do úvodní obrazovky bude zadávat za informace. Někteří zvolí „počet náhodných zdí“, někteří „počet jablek nutných pro nový život“, někteří třeba „rychlost pohybu hada“.

Ona „rychlost pohybu hada“ jde ve flashi ovlivnit poměrně komplikovaně – vlastní fps animace za chodu této animace měnit nelze. Jedinou možností je nastavit fps velmi vysoké a při jednotlivých onEnterFrame vždy zjišťovat, zda je aktuální frame dělitelný obtížností a pokud ne, vůbec jej neprovádět.

V tomto DUMu si do prvního frame zavedeme jméno a „počet jablek pro nový život“. Do časové osy vložíme nový prázdný frame (možností je několik, preferuji Drag&Dropnout aktuální herní frame z prvního políčka do druhého):



Do prvního frame nakreslíme:

- uvítání a texty okolo,
- textInput pro jméno hráče (s InstanceName `jmeno`),
- CheckBox s otázkou, zda chceme přidávat životy (s InstanceName `pridavat`),
- NumericStepper s číslem, po kolika jablkách chceme životy přidávat (s InstanceName `pridavatza`), vhodné je nastavit mu rozumné minimum a maximum v panelu Properties, v části Parameters
- Button s nápisem Start.



Nezapomeneme do kódu plochy napsat

```
stop();  
_root.txtjablek.visible=false;  
_root.pridavatza.visible=false;
```

aby animace zůstala stát na prvním snímku a NumericStepper s textem txtjablek nebyly vidět.

V kódu CheckBoxu bude (povšimněte si, jak elegantně přiřazujeme hodnotu typu Boolean):

```
on(click) {  
    _root.txtjablek.visible=(_root.pridavat.selected);  
    _root.pridavatza.visible=(_root.pridavat.selected);  
}
```

V kódu tlačítka bude:

```
on(click) {_root.zadane_jmeno_hrace="Hráč: "+_root.jmeno.text;  
    if (_root.pridavat.selected)  
        _root.zadane_pridavat_zivot=_root.pridavatza.value  
    else _root.zadane_pridavat_zivot=1000;  
    _root.nextFrame();  
}
```

Neboli: Je-li CheckBox zaškrtnutý, použijeme hodnotu NumericStepperu, není-li zaškrtnutý, použijeme hodnotu 1000. (Pozn.: Není předpoklad, že by had dokázal sníst 1000 jablek...)

Budeme chtít v hlavní hře zadaná data používat.

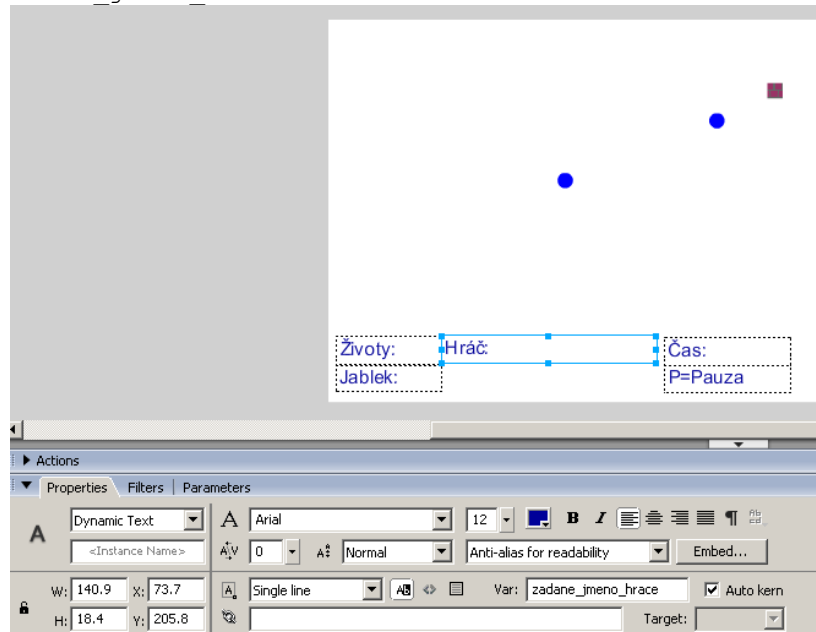
V případě počtu jablek, za něž se přidává život, musíme sáhnout do kódu hlavy a zaměnit

```
var pocetjablekprobonus:Number = 5;
```

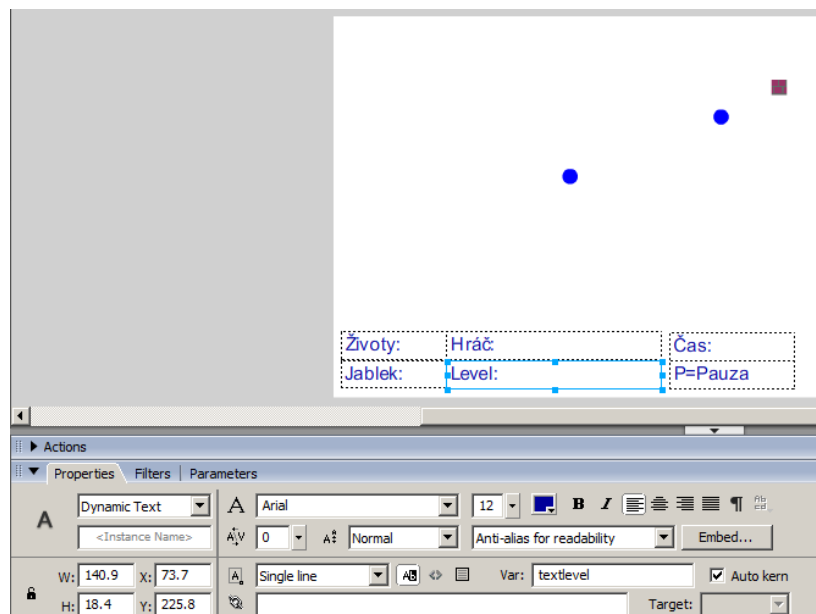
za

```
var pocetjablekprobonus:Number = _root.zadane_pridavat_zivot;
```

V případě jména bychom asi vyrobili dynamický text a v panelu Properties mu jako Var: rovnou zadali `zadane_jmeno_hrace`:



Při této příležitosti bych rovnou vyrobil dynamický text „Level:“, spojil jej s proměnnou `textlevel`:



a nechal jej rovněž aktualizovat, nejlépe na začátku funkce `tvor_zdi`:

```
function tvor_zdi(kterylevel:Number):Void {  
textlevel="Level: "+kterylevel;  
switch (kterylevel) {  
...  
}
```

Samostatný frame za hrou

Možností, jak jej zpracovat, je několik, v tomto DUMu vyrobíme speciální frame za herním framem. Do něj nakreslíme nápis „KONEC HRY“ (s InstanceName `gameover`), nápis „HRÁT ZNOVU“ (s InstanceName `gamerestart`) a prázdné textové pole spojené s proměnnou `textsummary` (to bude mít InstanceName `gameresult`). Abychom zajistili, že se v ploše třetího frame nebudou vidět žádné objekty z druhého či prvního frame, je nejjednodušší v ploše třetího frame napsat kód, který všechny objekty smaže (až na výše uvedené objekty začínající písmeny „game“):



Pro mazání všech objektů použijeme kopii kódu z funkce `vymaz_zdi`, pouze upravíme délku a znění porovnávacího řetězce pro `._name` – do plochy třetího frame napíšeme:

```
for (k in _root)
  if (_root[k]._name.substr(0,4)!="game")
    {_root[k].swapDepths(40);
     removeMovieClip(_root[k]);}
```

Text „HRÁT ZNOVU“ bude mít

```
on(press) {_root.prevFrame();}
```

A to je vše, přátelé

Hra Had vyrobená v této sadě není jedinou možnou variantou; za cca pět pokusů, v nichž jsme s různými skupinkami studentů hru vyráběli, jsme dospěli k pěti různým verzím – níže předkládám náměty, čemu se lze věnovat:

- zlepšení ovládání (aktuálně při průběhu `onEnterFrame` zkoumáme, zda je stisknuta klávesa; může se ale stát, že hráč klávesu stiskne mimo průběh `onEnterFrame`; ta se pak ignoruje)
- nemožnost kousnout se do krku
- zkracování délky hada při přechodu do nového levelu
- víc jablek
- reset času při znovunačtení hry (implementace proměnné `caszacatek`, kterou máme de facto připravenou; její odečítání při refreshi časového údaje)
- jemnější animace, kdy had neskáče po násobcích deseti pixelů, ale animuje se průběžně; změna směru je ale možná právě jen v násobcích deseti pixelů
- animace hlavy hada (bude na nás mrkat očičkem nebo otvírat ústa)
- další levely

Budete-li chtít tuto sadu DUMů ověřit, kontaktujte autora, který Vám rád pošle zdrojový `.fla` soubor. Jeho kód lze též poskládat z níže uvedených (finálních) zdrojových kódů:

Zdrojové kódy prvního frame

v ploše:

```
stop();
_root.txtjablek.visible=false;
_root.pridavatza.visible=false;
```

v CheckBoxu
pridavat:

```
on(click) {
    _root.txtjablek.visible=( _root.pridavat.selected);
    _root.pridavatza.visible=( _root.pridavat.selected);
}
```

v tlačítku
START:

```
on(click) {_root.zadane_jmeno_hrace="Hráč: "+_root.jmeno.text;
    if ( _root.pridavat.selected)
        _root.zadane_pridavat_zivot=_root.pridavatza.value
    else _root.zadane_pridavat_zivot=1000;
    _root.nextFrame();}
```

Zdrojové kódy druhého (herního) frame

v ploše:

```
onEnterFrame=function() {
if (getTimer())>=60000)
    textcas="Čas: "+(Math.floor(getTimer()/1000/60))+" m "+
        (Math.floor(getTimer()/1000)%60)+" s";
    else textcas="Čas: "+Math.floor(getTimer()/1000)+" s";
}

caszacatek=getTimer();
textcas = "Čas: "+Math.floor(getTimer()/1000)+" s";
var snd:Sound = new Sound();
snd.attachSound("tada");
var zdix:Array = Array();
var zdiy:Array = Array();
var klonzdi:Array = Array();
tvor_zdi(1);

tvor_jablko("jablko1",41);

function vymaz_zdi():Void {
    for (k in _root)
        if ( _root[k]._name.substr(0,7)=="klonzdi")
            {_root[k].swapDepths(40);
            removeMovieClip(_root[k]);}
    _root.zdix=Array();
    _root.zdiy=Array();
} // of vymaz_zdi

function tvor_zdi(kterylevel:Number):Void {
textlevel="Level: "+kterylevel;
switch (kterylevel) {
case 1:
// 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
```



```

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);
    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(koho:Object) {
    koho.zivoty--;_root.textzivoty='Životy: '+koho.zivoty;
    koho._x=160;
    koho._y=100;
    koho.smer=4;
    for (var m=0;m<koho.telox.length;m++)
        {koho.telox[m]=koho._x;
        koho.teloy[m]=koho._y;
        }
    if (koho.zivoty<1) {textsummary=zadane_jmeno_hrace+newline+
        textlevel+newline+
        textcas+newline+
        textjablek;
        nextFrame();}
}

```

v plóše:

```

on (press) {pauza();}
onClipEvent(enterFrame) {if (Key.isDown(80)) pauza();}
// kod 80 odpovida fyzicke klavese P
onClipEvent(load) {
    pauza=function() {_root.hlava.smer=5;
    }
}

```

v tlačítku P=Pauza:

v objektu hlava:

```
onClipEvent(enterFrame)
{
    if (Key.isDown(Key.LEFT)) smer=2;
    if (Key.isDown(Key.RIGHT)) smer=0;
    if (Key.isDown(Key.UP)) smer=1;
    if (Key.isDown(Key.DOWN)) smer=3;
    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.jablko1._x)&&(_root.hlava._y==_root.jablko1._y))
    {
        jablek++;_root.textjablek='Jablek: '+jablek;
        if ((jablek % pocetjablekprobonus) == 0)
            {zivoty++;_root.textzivoty='Životy: '+zivoty;
            _root.snd.start();}
        if ((jablek % pocetjablekprolevel) == 0)
            {_root.vymaz_zdi();
            _root.tvor_zdi(1+Math.floor(jablek/pocetjablekprolevel));
            this.zivoty++;_root.smrt(this);
            }
        _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.zdix[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.hlava.teloy[m]))
            {obsazeno=1;break;}
    if (obsazeno==1) _root.smrt(this);
    /* if ((_root.hlava._x>=320) || (_root.hlava._y>=200) ||
    (_root.hlava._x<0) || (_root.hlava._y<0)) _root.smrt();*/
    // VYPOCET NOVEHO TELA:
    if (_root.hlava.smer<4)
        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;
    // 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)
```

v objektu hlava:

```
onClipEvent(load) {
    var telox:Array = Array();
    var teloy:Array = Array();
    var defaultnidelka:Number = 4;
    var zivoty:Number = 5;_root.textzivoty='Životy: '+zivoty;
    var jablek:Number = 0;_root.textjablek='Jablek: '+jablek;
    var pocetjablekprobonus:Number = 5;
    var pocetjablekprolevel:Number = 3;
    _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)
```

Zdrojové kódy třetího frame

v ploše:

```
for (k in _root)
    if (_root[k]._name.substr(0,4)!="game")
        {_root[k].swapDepths(40);
        removeMovieClip(_root[k]);}
```

v tlačítku
HRÁT ZNOVU:

```
on(press) {_root.prevFrame();}
```