

# Kurz LSL skriptování

Shiny Iceberg

2009



součást cyklu  
**Nejsme jelita**



# Kurz LSL skriptování

## Shiny Iceberg

v Second Life od roku 2006  
*shiny.iceberg@virtualmagazine.cz*

### Aktuální projekty

virtualmagazine.cz  
Urbanica, Shinyland  
Bwindi Orphans  
cyklus Nejsme jelita





# Organizační body

## Průběh lekce

- bude trvat zhruba 90-120 minut
- pokud nekladete dotazy, vypněte si mikrofon
- příklady si klidně zkoušejte přímo v hledišti

## Vaše otázky

- můžete se ptát na konci každého snímku nebo na konci celé přednášky
- dotazy mohou být přes voice nebo IM
- při psaní IM z posledních řad použijte Shout



# Plán přednášky

1. Fungování a struktura skriptu
2. Z čeho se skládá skript
3. Vlastnosti objektu
4. Pohyb objektu
5. Pose bally
6. Komunikace skriptu
7. Inventory objektu
8. Detektory
9. Particles
10. Příklady a dokumentace



# Co jsou particles 1/2

- každý prim v SL může generovat „částice“
- částice mohou simulovat různé vizuální efekty jako oheň, kouř, mlhu, déšť, vodní proud, odlesky na objektech, stopy za avatarem, padající listí, trávu či květiny na zemi, záplavu srdíček létacící mezi avatary, krvácení aj.
- částice jsou vykreslovány jako 2D objekty (sprites), tedy jako obdélníky s texturou, které jsou vždy natočeny čelem k avatarovi



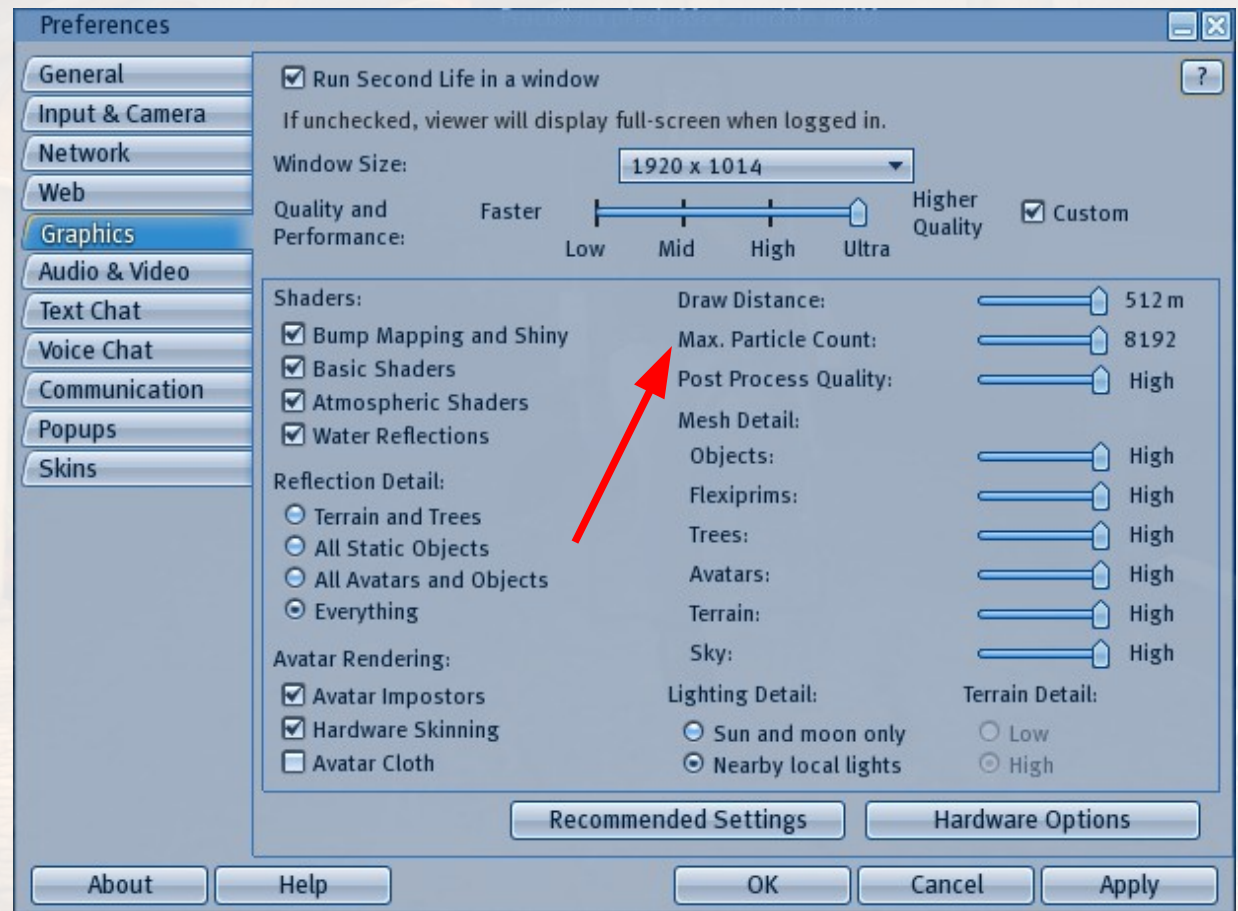
## Co jsou particles 2/2

- generování částic je vlastnost primu, tj. generování pokračuje i po smazání skriptu
- generování lze zapnout/vypnout skriptem
- částice mají omezenou dobu životnosti (max 30 sekund) a nepočítají se do prim limitu
- částice jsou vykreslovány na straně klienta, zatěžují tedy váš počítač, nikoliv simulátor
- částice se nechovají jako primy, jsou průchozí a nelze na ně zaostřit kameru



# Viditelnost particles

Maximální množství viditelných particles v jeden okamžik si každý může nastavit v klientovi v rozmezí 0 - 8192. Vyšší hodnotu lze nastavit manuálně v *settings.ini*.





# Informace, příklady a nástroje

## llParticleSystem

<http://www.lslwiki.net/lslwiki/wakka.php?wakka=llParticleSystem>

## Particle Laboratory

<http://slurl.com/secondlife/Teal/200/60/21/>

## Particles LSL generátor

<http://particles-lsl-generator.bashora.com/index.php>

## Particle Organ na Xstreet SL

<https://xstreetsl.com/modules.php?name=Marketplace&file=item&ItemID=1057705>

## Particle Library

<http://secondlife.mitsi.com/cgi/llscript.plx?Category=Particles>



```
lParticleSystem(list Parameter) ;
```

**VSTUP:** seznam parametrů a jejich hodnot ve tvaru

```
[ parametr1, hodnota1,  
  parametr2, hodnota2,  
  ...  
  parametrX, hodnotaX ]
```

**VÝSTUP:** nic

**POZNÁMKA:** vypnutí particles se vynutí příkazem

```
lParticleSystem([]) ;
```



# Parametry `llParticleSystem` 1/6

```
// =====  
// VZHLED CASTIC  
// =====  
  
// nazev nebo UUID textury na casticich  
// nastaveni nize vezme prvni texturu z inventory primu  
// pokud neni textura nalezena, pouzije se defaultni  
PSYS_SRC_TEXTURE,          llGetInventoryName (INVENTORY_TEXTURE, 0) ,  
  
// pocatecni a konecna velikost castic v metrech, maximalne 4 metry  
PSYS_PART_START_SCALE,    <0.5, 0.5, FALSE> ,  
PSYS_PART_END_SCALE,      <0.2, 0.2, FALSE> ,  
  
// pocatecni a konecna barva castic  
// zadavane jako barevny vektor se slozkami RGB  
PSYS_PART_START_COLOR,    <0.0, 1.0, 1.0> ,  
PSYS_PART_END_COLOR,      <1.0, 0.5, 0.0> ,  
  
// pocatecni a konecna pruhlednost castic  
// rozmezi 0.0 (zcela pruhledny) az 1.0 (zcela nepruhledny)  
PSYS_PART_START_ALPHA,    0.5 ,  
PSYS_PART_END_ALPHA,      0.8 ,
```



# Parametry IIParticleSystem 2/6

```
// =====  
// GENEROVANI CASTIC  
// =====  
  
// pocet castic generovanych pri kazdem vytrysku  
PSYS_SRC_BURST_PART_COUNT, 5,  
  
// kolik sekund mezi jednotlivymi vytrysky  
PSYS_SRC_BURST_RATE, 0.2,  
  
// kolik sekund bude kazda castice existovat, maximalne 30 sekund  
PSYS_PART_MAX_AGE, 5.0,  
  
// kolik sekund se maji castice generovat od okamziku, kdy se  
// zdroj dostane do zorneho pole avatara poprvé, 0 = neomezene  
PSYS_SRC_MAX_AGE, 0.0,
```



# Parametry IIParticleSystem 3/6

```
// =====  
// ROZMISTOVANI CASTIC  
// =====  
  
// typ rozmistovani castic  
// 1 = DROP: jen vytvorit  
// 2 = EXPLODE: rozmetat nahodne na vsechny strany  
// 4 = ANGLE: generovat v 2D kruhove vyseci  
// 8 = CONE: generovat v 3D kulove vyseci  
// 16 = CONE EMPTY: generovat vsude kolem krome 3D kulove vysece  
PSYS_SRC_PATTERN,          4,  
  
// polomer umisteni generovanych castic v metrech  
// parametr je neplatny pro typ DROP  
// priznak PSYS_PART_FOLLOW_SRC_MASK nesmi byt zapnuty (viz nize)  
PSYS_SRC_BURST_RADIUS,    3.0,  
  
// pocatecni a koncovy uhel v radianech definujici vysec  
// parametr je neplatny pro typ DROP a EXPLODE  
// konstanta PI odpovida generovani vsude kolem  
PSYS_SRC_ANGLE_BEGIN,    0.4 * PI,  
PSYS_SRC_ANGLE_END,     0.5 * PI,  
  
// rotace vysece, do ktere se umistuji castice  
// rotace je v radianech podle os X, Y a Z  
// parametr je neplatny pro typ DROP a EXPLODE  
PSYS_SRC_OMEGA,         <0.0, 0.0, 3.0>,
```



# Parametry llParticleSystem 4/6

```
// =====  
// POHYB CASTIC  
// =====  
  
// minimalni a maximalni pocatecni rychlost castic  
// parametr je neplatny pro typ DROP  
PSYS_SRC_BURST_SPEED_MIN,    3.0,  
PSYS_SRC_BURST_SPEED_MAX,    5.0,  
  
// zrychlovani castic po vygenerovani v osach X, Y a Z  
PSYS_SRC_ACCEL,              <0.0, 0.0, 2.0 >,  
  
// UUID objektu nebo avatara, ke kteremu maji castice smerovat  
// pouziti "llGetKey()" znamena, ze se castice vraci do objektu  
// pouziti "llGetOwner()" znamena, ze castice miri na vlastnika  
// priznak PSYS_PART_TARGET_POS_MASK nesmi byt vypnuty (viz nize)  
PSYS_SRC_TARGET_KEY,         llGetOwner(),
```



# Parametry IIParticleSystem 5/6

```
// =====  
// DALSI CHOVANI CASTIC  
// =====  
  
// parametr obsahujici priznaky, ktere ovlivnuji chovani castic  
// priznaky se zadavaji klicovym slovem a oddeluji se znakem |  
// deaktivace priznaku se provadi uvedenim znaku // pred radkem  
// prvni radek PSYS_PART_FLAGS musi zustat vzdy aktivni  
PSYS_PART_FLAGS, ( 0  
  
    // postupna zmena barvy castic z pocatecni na konecnou  
    | PSYS_PART_INTERP_COLOR_MASK  
  
    // postupna zmena velikosti castic z pocatecni na konecnou  
    | PSYS_PART_INTERP_SCALE_MASK  
  
    // castice vyzaruji svetelnou zari (GLOW)  
    | PSYS_PART_EMISSIVE_MASK  
  
    // svisla osa textury na casticich se nataci dle smeru pohybu  
    | PSYS_PART_FOLLOW_VELOCITY_MASK  
  
    // castice jsou unasene SL vetrem  
    | PSYS_PART_WIND_MASK
```



# Parametry `ParticleSystem` 6/6

```
// castice se nedostanou nize nez Z souradnice zdroje castic
// chovaji se, jako by v teto vysce byla pruhledna deska,
// na ktere castice poskakují a zustavaji lezet
| PSYS_PART_BOUNCE_MASK

// castice se tvori uprostred zdroje
// pri deaktivaci se tvori podle parametru PSYS_SRC_BURST_RADIUS
| PSYS_PART_FOLLOW_SRC_MASK

// castice se po vygenerovani pohybují smerem k vybranemu
// avatarovi nebo objektu dle parametru PSYS_SRC_TARGET_KEY
// pri deaktivaci se castice pohybují volne do prostoru
| PSYS_PART_TARGET_POS_MASK

// doplnkova funkce k priznaku PSYS_PART_TARGET_POS_MASK
// aktivace zpusobi, ze castice se pohybují k cili po primce
| PSYS_PART_TARGET_LINEAR_MASK
```

```
)
```



# Příklad: Kouř (krátké)

```
default
{
    state_entry()
    {
        llParticleSystem([
            PSYS_PART_START_SCALE, <0.1, 0.5, 0.0>, PSYS_PART_END_SCALE, <2.0, 2.0, 0.0>,
            PSYS_PART_START_COLOR, <0.7, 0.7, 0.7>, PSYS_PART_END_COLOR, <0.7, 0.7, 0.7>,
            PSYS_PART_START_ALPHA, 0.3, PSYS_PART_END_ALPHA, 0.0,
            PSYS_SRC_BURST_PART_COUNT, 5, PSYS_SRC_BURST_RATE, 0.2,
            PSYS_PART_MAX_AGE, 5.0, PSYS_SRC_MAX_AGE, 0.0,
            PSYS_SRC_PATTERN, 8,
            PSYS_SRC_ANGLE_BEGIN, 0.0, PSYS_SRC_ANGLE_END, 0.05*PI,
            PSYS_SRC_BURST_SPEED_MIN, 0.5, PSYS_SRC_BURST_SPEED_MAX, 1.0,
            PSYS_SRC_ACCEL, <0.0, 0.0, 0.5 >,
            PSYS_PART_FLAGS, (PSYS_PART_INTERP_COLOR_MASK | PSYS_PART_INTERP_SCALE_MASK
                            | PSYS_PART_WIND_MASK | PSYS_PART_FOLLOW_SRC_MASK)
        ]);
    }
}
```



# Příklad: Kouř (s detaily) - 1/6

```
// Některé parametry jsou záměrně deaktivovány
// Jejich aktivace se provede odstraněním // na začátku řádky

default
{
    state_entry()
    {
        llParticleSystem([
            PSYS_SRC_TEXTURE,          llGetInventoryName(INVENTORY_TEXTURE, 0),

            // počáteční a konečná velikost částic v metrech, maximálně 4 metry
            PSYS_PART_START_SCALE,    <0.1, 0.5, FALSE>,
            PSYS_PART_END_SCALE,      <2.0, 2.0, FALSE>,

            // počáteční a konečná barva částic
            // zadává se jako barevný vektor se složkami RGB
            PSYS_PART_START_COLOR,    <0.7, 0.7, 0.7>,
            PSYS_PART_END_COLOR,      <0.7, 0.7, 0.7>,

            // počáteční a konečná průhlednost částic
            // rozsah 0.0 (zcela průhledný) až 1.0 (zcela neprůhledný)
            PSYS_PART_START_ALPHA,    0.3,
            PSYS_PART_END_ALPHA,      0.0,
        ])
```



# Příklad: Kouř (s detaily) - 1/6

```
// =====  
// GENEROVANI CASTIC  
// =====  
  
// pocet castic generovanych pri kazdem vytrysku  
PSYS_SRC_BURST_PART_COUNT, 5,  
  
// kolik sekund mezi jednotlivymi vytrysky  
PSYS_SRC_BURST_RATE, 0.2,  
  
// kolik sekund bude kazda castice existovat, maximalne 30 sekund  
PSYS_PART_MAX_AGE, 5.0,  
  
// kolik sekund se maji castice generovat od okamziku, kdy se  
// zdroj dostane do zorneho pole avatara poprve, 0 = neomezene  
PSYS_SRC_MAX_AGE, 0.0,
```



# Příklad: Kouř (s detaily) - 1/6

```
// =====  
// ROZMISTOVANI CASTIC  
// =====  
  
// typ rozmistovani castic  
// 1 = DROP: jen vytvorit  
// 2 = EXPLODE: rozmetat nahodne na vsechny strany  
// 4 = ANGLE: generovat v 2D kruhove vyseci  
// 8 = CONE: generovat v 3D kulove vyseci  
// 16 = CONE EMPTY: generovat vsude kolem krome 3D kulove vysece  
PSYS_SRC_PATTERN,          8,  
  
// polomer umisteni generovanych castic v metrech  
// parametr je neplatny pro typ DROP  
// priznak PSYS_PART_FOLLOW_SRC_MASK nesmi byt zapnuty (viz nize)  
PSYS_SRC_BURST_RADIUS,    0.0,  
  
// pocatecni a koncovy uhel v radianech definujici vysec  
// parametr je neplatny pro typ DROP a EXPLODE  
// konstanta PI odpovida generovani vsude kolem  
PSYS_SRC_ANGLE_BEGIN,    0.00 * PI,  
PSYS_SRC_ANGLE_END,     0.05 * PI,  
  
// rotace vysece, do ktere se umistuji castice  
// rotace je v radianech podle os X, Y a Z  
// parametr je neplatny pro typ DROP a EXPLODE  
PSYS_SRC_OMEGA,         <0.0, 0.0, 0.0>,
```



# Příklad: Kouř (s detaily) - 1/6

```
// =====  
// POHYB CASTIC  
// =====  
  
// minimalni a maximalni pocatecni rychlost castic  
// parametr je neplatny pro typ DROP  
PSYS_SRC_BURST_SPEED_MIN, 0.5,  
PSYS_SRC_BURST_SPEED_MAX, 1.0,  
  
// zrychlovani castic po vygenerovani v osach X, Y a Z  
PSYS_SRC_ACCEL, <0.0, 0.0, 0.5 >,  
  
// UUID objektu nebo avatara, ke kteremu maji castice smerovat  
// pouziti "llGetKey()" znamena, ze se castice vraci do objektu  
// pouziti "llGetOwner()" znamena, ze castice miri na vlastnika  
// priznak PSYS_PART_TARGET_POS_MASK nesmi byt vypnuty (viz nize)  
// PSYS_SRC_TARGET_KEY, llGetOwner(),
```



# Příklad: Kouř (s detaily) - 1/6

```
// =====  
// DALSI CHOVANI CASTIC  
// =====  
  
// parametr obsahujici priznaky, ktere ovlivnuji chovani castic  
// priznaky se zadavaji klicovym slovem a oddeluji se znakem |  
// deaktivace priznaku se provadi uvedenim znaku // pred radkem  
// prvni radek PSYS_PART_FLAGS musi zustat vzdy aktivni  
PSYS_PART_FLAGS, ( 0  
  
    // postupna zmena barvy castic z pocatecni na konecnou  
    | PSYS_PART_INTERP_COLOR_MASK  
  
    // postupna zmena velikosti castic z pocatecni na konecnou  
    | PSYS_PART_INTERP_SCALE_MASK  
  
    // castice vyzaruji svetelnou zari (GLOW)  
    // | PSYS_PART_EMISSIVE_MASK  
  
    // svisla osa textury na casticich se nataci dle smeru pohybu  
    // | PSYS_PART_FOLLOW_VELOCITY_MASK  
  
    // castice jsou unasene SL vetrem  
    | PSYS_PART_WIND_MASK
```



# Příklad: Kouř (s detaily) - 1/6

```
// castice se nedostanou nize nez Z souradnice zdroje castic
// chovaji se, jako by v teto vysce byla pruhledna deska,
// na ktere castice poskakuji a zustavaji lezet
// | PSYS_PART_BOUNCE_MASK

// castice se tvori uprostred zdroje
// pri deaktivaci se tvori podle parametru PSYS_SRC_BURST_RADIUS
| PSYS_PART_FOLLOW_SRC_MASK

// castice se po vygenerovani pohybují smerem k vybranemu
// avatarovi nebo objektu dle parametru PSYS_SRC_TARGET_KEY
// pri deaktivaci se castice pohybují volne do prostoru
// | PSYS_PART_TARGET_POS_MASK

// doplnkova funkce k priznaku PSYS_PART_TARGET_POS_MASK
// aktivace zpusobi, ze castice se pohybují k cili po primce
// | PSYS_PART_TARGET_LINEAR_MASK
)

]); // ukoncení parametru funkce llParticleSystem
}
}
```



# Otázky a diskuze

