

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

To nejdůležitější z minula

- `llGetStatus` / `llSetStatus`
- `llGetPos` / `llGetLocalPos` / `llGetRootPosition`
- `llSetPos`
- `llGetRot` / `llGetLocalRot` / `llGetRootRotation`
- `llSetRot`
- `llSetPrimitiveParams`
- `llTargetOmega`
- `llRot2Up` / `llRot2Fwd` / `llRot2Left`
- `llGetMass` / `llGetVel`
- `llGetForce` / `llSetForce`
- `llApplyImpulse`

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

Animace v SL

Animace = sekvence statických póz, které hýbou avatarem a napodobují pohyb, v SL se používá BHV formát

Animací mohu avatara rozhýbat

- přímým spuštěním z inventory (jen sebe)
- skriptem v attachmentu nebo v objektu, na kterém avatar sedí
- skriptem v objektu, pokud znám *key* avatara a pokud avatar odsouhlasí své animování

Priorita animací

V Second Life je celá řada vestavěných animací (na sezení, chůzi, létání atd., které se přehrávají standardně, pokud není skriptem určeno jinak.

Každá animace má svou prioritu 0-4 definovanou při importu do SL. Animace s vyšší prioritou přebere kontrolu pohybu avatara. Animace, která hýbe pouze s částí těla, nepřebírá kontrolu zbývajících částí avatara.

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

Žádaná poloha sezení

Jsou dvě obvyklé možnosti, jak nastavit, kde přesně má avatar sedět (místo a natočení):

- pomocí „kuličky“ - poseballu - který posunu a natočím dle potřeby
- pomocí poseball skriptu přímo v objektu, který nastaví místo a natočení pomocí *llSitTarget* funkce

Jak animovat při sezení

- 1) skript nastaví polohu a rotaci sezení
- 2) avatar usedne na objekt
- 3) skript zjistí *key* tohoto avatara
- 4) skript požádá o práva na animování avatara
- 5) po automatickém obdržení práv začne skript animovat avatara
- 6) při ukončení sezení zastavím animaci

Funkce a event pro poseball

- `ISitTarget`
- `IAvatarOnSitTarget`
- `IGetAnimation / IGetAnimationList`
- `IStartAnimation / IStopAnimation`
- `IRequestPermissions`
- `IGetPermissionsKey`
- `run_time_permissions (event)`

`llSitTarget(vector Posun, rotation Rot);`

VSTUP:

- **Posun:** místo sezení vzhledem ke středu primu
- **Rot:** pootočení avatara v místě sezení

VÝSTUP: nic

POZNÁMKA:

- zadáním `llSitTarget(ZERO_VECTOR, ZERO_ROTATION)` dojde k odstranění místa sezení

`ZERO_VECTOR = <0.0, 0.0, 0.0>`

`ZERO_ROTATION = <0.0, 0.0, 0.0, 1.0>`

Příklad llSitTarget

```
// Avatar bude sedet 0.1 metru nad stredem objektu
// a nebude nijak natoceny
llSitTarget(<0.0,0.0,0.1>, ZERO_ROTATION);

// Avatar bude sedet 0.1 metru nad stredem objektu
// a bude otocen o 90 stupnu na ose Z
vector RotaceRad = <0.0,0.0,90.0> * DEG_TO_RAD;
rotation MojeRotace = llEuler2Rot(RotaceRad);
llSitTarget(<0.0,0.0,0.1>, MojeRotace);
```

Ideální je použití nástrojů pro zjištění vhodného llSitTarget - např. *Sit Target Helper*, *PrimSitter*, *Pos Rot Determinator*, *Positioner Personal*

Objekt s několika ISitTarget

Každý prim v objektu může mít svůj Sit Target. Při kliknutí (sednutí) na prim v objektu se postupuje následovně v tomto pořadí:

- 1) Pokud má daný prim Sit Target definován a není obsazen, avatar se posadí na něj
- 2) V opačném případě se kontrolují Sit Targety a obsazenost všech primů v objektu (linksetu) podle pořadí primů (1 je root prim)

```
key llAvatarOnSitTarget ();
```

VSTUP: nic

VÝSTUP: UUID avatara, který sedí na Sit Targetu,
případně NULL_KEY

```
string llGetAnimation(key Avatar) ;
```

VSTUP: avatar, u kterého chceme zjistit stav animování

VÝSTUP: typ aktuálního stavu animování, vrací se následující hodnoty:

Crouching
Hovering Up
Running
Standing
FlyingSlow
Landing
Turning Left

Hovering
Jumping
Sitting
Striding
Falling Down
Soft Landing
Turning Right

Hovering Down
PreJumping
Sitting on Ground
Flying
Standing Up
CrouchWalking
Walking

```
list llGetAnimationList(key Avatar);
```

VSTUP: avatar, u kterého chceme zjistit seznam animací

VÝSTUP: seznam UUID všech animací, které se aktuálně přehrávají u avatara

`!StartAnimation (string Animace) ;`

VSTUP: název animace, která se má začít přehrávat na avatarovi, tomuto příkazu musí předcházet získání oprávnění na animování avatara

VÝSTUP: nic

POZNÁMKA: animace se zadává názvem, nikoliv přes UUID a s výjimkou standardních animací se přehrávaná animace musí nacházet v primu

Standardní animace:

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

11StopAnimation (string Animace) ;

VSTUP: název animace, která se má přestat přehrávat na avatarovi, tomuto příkazu musí předcházet získání oprávnění na animování avatara

VÝSTUP: nic

POZNÁMKA: U tohoto příkazu se animace rovněž zadává názvem a s výjimkou standardních animací se přehrávaná animace musí nacházet v primu. Lze ale použít i UUID animace bez nutnosti mít animaci v primu.

`llRequestPermissions`

`(key Avatar, integer TypOpraveni) ;`

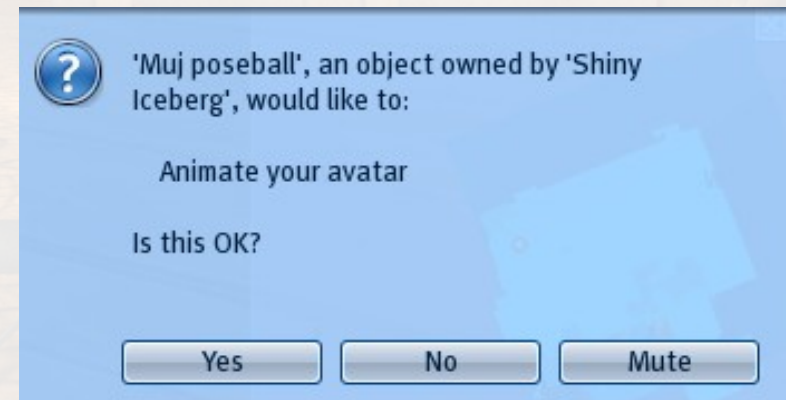
VSTUP:

- **Avatar:** avatar, u kterého chceme získat určité oprávnění
- **Avatar:** typ oprávnění, které chceme získat, pro animování potřebujeme zadat konstantu `PERMISSION_TRIGGER_ANIMATION`

VÝSTUP: nic

Poznámky k `llRequestPermissions`

- při použití funkce se u avatara objeví okno skriptu, které žádá přidělení oprávnění, v případě odpovědi Yes nebo No dojde k aktivaci eventu `run_time_permissions`
- u posazení na objekt nebo u attachmentu je ovšem povolení poskytnuto automaticky
- v daný okamžik může mít ve skriptu nějaké oprávnění pouze jediný avatar



```
key llGetPermissionsKey ();
```

VSTUP: nic

VÝSTUP: UUID avatara, který má ve skriptu nějaká práva (podle posledního volání *llRequestPermissions*)

Event *run_time_permissions*

run_time_permissions (integer *TypOpraveneni*)

- aktivuje se v okamžiku, kdy avatar odpoví kladně nebo záporně na dialog *IRequestPermissions*
- obvyklým obsahem tohoto eventu je otestování, zda práva byla udělena a sada příkazů, které se v takovém případě mají provést,
- u poseballu se oprávnění nemusí testovat a lze přímo přepnout animaci pro sezení
- do proměnné *TypOpraveneni* se uloží získaná práva

Židle s animací 1/2

```
string MojeAnimace;  
  
default  
{  
    state_entry()  
    {  
        llSetClickAction(CLICK_ACTION_SIT);  
        llSitTarget(<0.35,0.0,0.25>, ZERO_ROTATION);  
        MojeAnimace = llGetInventoryName(INVENTORY_ANIMATION, 0);  
        if(MojeAnimace == "")  
        {  
            llOwnerSay("No animation in the poseball!");  
            MojeAnimace = "sit";    // pouziji standardni animaci  
        }  
    }  
  
    run_time_permissions(integer TypOpravneni)  
    {  
        llStopAnimation("sit");  
        llStartAnimation(MojeAnimace);  
    }  
}
```

Židle s animací 2/2

```
changed(integer TypZmeny)
{
    // znak & se pouziva pro bitove porovnaní
    if(TypZmeny & CHANGED_LINK)
    {
        key Avatar = llAvatarOnSitTarget();
        if (Avatar != NULL_KEY)
        {
            llRequestPermissions(Avatar, PERMISSION_TRIGGER_ANIMATION);
        } else {
            // skript bude obvykle fungovat spravne
            // i bez vypinani animace pri ukonceni sezeni
            llStopAnimation(MojeAnimace);
        }
    }

    if(TypZmeny & CHANGED_INVENTORY)
    {
        llResetScript();
    }
}
```


Teleport pomocí llSitTarget 1/2

```
// Jake jsou souradnice cile?  
// Maximalne 300 metru daleko na libovolne ose  
vector LOKALITA = <25.0, 167.0, 400.0>;  
  
NastavitTeleport()  
{  
    vector cilSouradnice;  
    rotation cilRotace;  
  
    // prevod globalnich souradnic na lokalni  
    cilSouradnice = LOKALITA - llGetPos();  
  
    // odstraneni efektu natoceni objektu  
    cilSouradnice = LOKALITA * (ZERO_ROTATION / llGetRot());  
    cilRotace = ZERO_ROTATION / llGetRot();  
  
    llSitTarget(cilSouradnice, cilRotace);  
}
```

Teleport pomocí llSitTarget 2/2

```
default
{
    state_entry()
    {
        llSetClickAction(CLICK_ACTION_SIT);
        NastavitTeleport();
    }

    moving_end()
    {
        NastavitTeleport();
    }

    changed(integer TypZmeny)
    {
        if (TypZmeny & CHANGED_LINK)
        {
            llSleep(0.5);
            if (llAvatarOnSitTarget() != NULL_KEY)
            {
                llUnsit(llAvatarOnSitTarget());
            }
        }
    }
}
```

Otázky a diskuze

