ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: B 2341 Strojírenství
Studijní zaměření: Informační a komunikační technologie ve strojírenském podniku

Bakalářská práce

Tvorba prostředí digitální továrny pro laboratoř virtuální reality za využití nástroje Virtools

Autor: **Milan Bauernöpl**
Vedoucí práce: **Ing. Petr Hořejší, Ph.D..**

Akademický rok 2011/2012

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě bakalářskou/diplomovou práci, zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.
Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této bakalářské/diplomové práce.

V Plzni dne: ……. . . . . . .

podpis autora

ANOTAČNÍ LIST

AUTOR	**Příjmení** Bauernöpl	**Jméno** Milan	
STUDIJNÍ OBOR	B2341 „Informační a komunikační technologie ve strojírenském podniku“		
VEDOUCÍ PRÁCE	**Příjmení (včetně titulů)** Ing. Hořejší, Ph.D.	**Jméno** Petr	
PRACOVIŠTĚ	ZČU - FST - KPV		
DRUH PRÁCE	~~**DIPLOMOVÁ**~~	**BAKALÁŘSKÁ**	**Nehodící se škrtněte**
NÁZEV PRÁCE	Tvorba prostředí digitální továrny pro laboratoř virtuální reality za využití nástroje Virtools		

FAKULTA	strojní	**KATEDRA**	KPV	**ROK ODEVZD.**	2012

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	46	**TEXTOVÁ ČÁST**	39	**GRAFICKÁ ČÁST**	0

STRUČNÝ POPIS (MAX 10 ŘÁDEK) **ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY**	Práce je zaměřena na virtuální realitu a program Virtools5, který slouží pro tvorbu virtuálního prostředí. Cílem je naučit se pracovat s programem Virtools5 a vytvořit interaktivní virtuální prostředí v laboratoři virtuální reality.
KLÍČOVÁ SLOVA **ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE**	stereoskopická projekce, virtuální realita, laboratoř virtuální reality, Virtools 5, 3D, stavební bloky, skripty

SUMMARY OF BACHELOR SHEET

AUTHOR	**Surname** Bauernöpl	**Name** Milan	
FIELD OF STUDY	B2341 " Information and Communication Technology in Industrial Management"		
SUPERVISOR	**Surname (Inclusive of Degrees)** Ing. Hořejší, Ph.D..	**Name** Petr	
INSTITUTION	ZČU - FST - KPV		
TYPE OF WORK	~~**DIPLOMA**~~	**BACHELOR**	**Delete when not applicable**
TITLE OF THE WORK	Creation of enviroment of the digital faktory for laboratory of virtual reality using tool Virtools		

FACULTY	Engineering		**DEPARTMENT**	KPV		**SUBMITTED IN**	2012

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	46		**TEXT PART**	39		**GRAPHICAL PART**	0

BRIEF DESCRIPTION **TOPIC, GOAL, RESULTS AND CONTRIBUTIONS**	This thesis dissert on virtual reality and program Virtools5, which serves to create a virtual environment. The aim is to learn to work with the program Virtools5 and create interactive virtual environments in laboratory of virtual reality.
KEY WORDS	Digital factory, stereoskopic projection, virtual reality, Virtools, 3D, Building blocks, skript

Obsah

Přehled použitých zkratek a symbolů

3D – trojrozměrná dimenze
CAVE - Computer Aided Virtual Environment – počítačová podpora virtuálního prostředí
PC – Personal computer - osobní počítač
IrDA - Infrared Data Association – infračervený port
KPV - Katedra průmyslového inženýrství a managementu
3DVIA – část společnosti Dassault Systèmes zabývající se 3D

1 Úvod

Tato práce se zaměřuje na virtuální realitu a zejména na program Virtools 5 od společnosti Dassault Systèmes. Práce nejprve seznamuje s pojmem virtuální realita, který začíná být stále více rozšířený. Zde budou ukázány druhy stereoskopie, na jejímž principu je založeno zobrazení 3D v laboratoři virtuální reality, která je zde podrobně popsána. V laboratoři je dále i hardwarové zařízení pro interaktivní zasahování do VR modelů. Toto zařízení snímá naše pohyby a podle toho upravuje obraz, který se nám promítá. Pro tvorbu prostředí pro virtuální realitu je použit program Virtools 5 v němž bylo za využití knihoven vytvořen projekt výrobní haly a ke které byly dále vytvořeny i skripty pro funkce a chování jednotlivých objektů v ní. V práci je podrobně popsána práce s programem, jak se co vytváří a vše je i ukázáno na ukázkách mého projektu. Dále je v práci uveden překlad a popis vybraných stavebních bloků.

1.1 Virtuální realita

Virtuální realita je intuitivní způsob jak zobrazit virtuální objekty. Virtuální realita je médium umožňující zobrazit trojrozměrnou virtuální scénu, do které může uživatel zasáhnout a stát se tak její součásti. Hlavní benefitem je interaktivita scény, díky které lze s předměty pohybovat, otáčet a vůbec všemožně je animovat. Další výhody jsou integrace různých medií do scény, jako je třeba zvuk, video.[1] Pro zobrazení virtuální reality lze použít stereoskopi, která bude vysvětlena v příští kapitole.

Hlavní charakteristiky, které mají vliv na její využitelnost:

- Je nutné, aby VR pracovala v reálném čase, aby mohla reagovat na to, co uživatel dělá.
- Pro splnění cílů musí vytvářet co nejlepší iluzi. Umělý svět s objekty má graficky trojrozměrný charakter.
- Uživatel neprohlíží umělý svět jen zvenčí, ale vstupuje do něj a interaguje s ním. Svět není statický, ale uživatel může umělý svět přetvářet a pohybovat se v něm. [1]

Využití virtuální reality

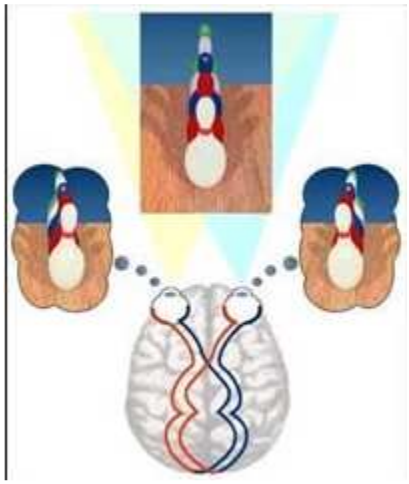
Virtuální realita se stává stále populárnější ve více odvětvích. Nyní se používá např. ve strojírenství, letectví, medicíně, sportu atd. Hodně rozšířená je zejména v automobilovém průmyslu. Virtuální realitu využívá většina předních automobilek.

1.2 Stereoskopie

Pod obecným označením 3D se neskrývá konkrétní technologie, termín pouze napovídá, že konkrétní zobrazení se řeší nikoliv v rámci dvou rozměrů (tedy s výškou a šířkou) - např. projekce u klasických filmů, které je možné vidět v běžných kinech, ale navíc i s třetím rozměrem - s hloubkou.

K tomu, aby mohl divák sledovat obraz skutečně včetně hloubky, se využívá stereoskopické vizualizace. Systém pak nepromítá pouze jeden obraz, který divák sleduje oběma očima zároveň, ale generuje dvojici oddělených obrazů (popř. vypočítán z dvou

projekcí – 3D animace). Přitom je ovšem nutné dodržet několik zásad. Zdrojový film musí být natočen dvojicí kamer, jejichž vzdálenost přibližně simuluje rozteč lidských očí. U animovaných filmů musí být obraz vypočítán dvakrát. To znamená, že dvojice obrazů je obdobná, jako kdyby se na scénu díval člověk. Dva obrazy se při projekci musejí dodat k divákovi tak, aby levý obraz vidělo pouze levé oko a pravé oko, aby sledovalo zase pouze záběr z pravé kamery. Mozek následně vyhodnotí obraz správně prostorově a tedy plně trojrozměrně (viz **Chyba! Nenalezen zdroj odkazů.**).[2]



Obr. 1.1 Prostorové vnímání člověka [2]

1.3 **Anaglyf**

Projekce typu Anaglyf je jednou z nejvíce rozšířených metod, jak lze zobrazit 3D prostorové obrázky či případně i animace a film. Známý je především proto, že je velmi snadné zajistit jeho projekci. Ve výsledku stačí pouze brýle, které jsou vybavené jednou červenou a jednou modrou nebo zelenou očnicí. Nepsaným pravidlem je, že levé oko je vždy zabarveno červeným filtrem. Pravá očnice je vybavena zeleným nebo častěji modrým filtrem. Sledovaná 3D scéna je vyrobena tak, že obsahuje smíchané oba stereoobrazy v sobě, pouze základní dvojice barev (červená a modrá nebo červená a zelená) slouží pro oddělení dvou obrazů. Pokud divák sleduje scénu s 3D brýlemi, do každého oka více či méně dostává (díky příslušným barevným filtrům) separátní obraz. Mozek ve výsledku vygeneruje z těchto obrazů 3D scénu (viz Obr. 1.2).

Bohužel, cenou za snadné a finančně nejméně náročné zobrazení anaglyfu se platí ztrátou barevných informací. Situace je o to komplikovanější, že divák vidí scénu každým okem zcela barevně jinak (jedním okem červeně a druhým modře nebo zeleně). Mozek diváka se sice tyto rysy snaží co možná eliminovat, ale vjem nikdy není tak kvalitní jako u jiných typů 3D projekcí.

Výhodou anaglyfů však je snadné šíření 3D záznamu, které lze jak tisknout např. do časopisů, knih, nebo nahrávat na běžné videokazety, přehrávat bez speciálních programů v PC nebo na běžném projektoru. [3]

Obr. 1.2 Anaglif [3]

1.4 **Pasivní stereoskopická projekce**

Pasivní 3D projekce je založena na brýlích, které mají v očnicích polarizační filtry. Jedna očnice má polarizační filtr orientovaný tak, že propouští pouze světlo kmitající v horizontální rovině. Druhá očnice obsahuje stejný o devadesát stupňů otočený filtr. Tedy takový, že propouští pouze světlo kmitající ve vertikální rovině. Dva obrazy se promítají na jednu projekční plochu, přičemž před každým projektorem je upevněn taktéž polarizační filtr. Nastavení filtrů na projektoru koresponduje s nastavením filtrů na brýlích. Dvojice obrazů (pro pravé a levé oko) se následně promítá na jednu projekční plochu, která je vyrobena ze speciálního materiálu a opatřena povrchem, který zachová polarizaci dopadajícího světla. Odražené obrazy od projekční plochy se dostávají k divákovi, nicméně do každého oka pronikne (díky polarizačním filtrům v očnicích) pouze příslušný obraz (viz Obr. 1.3).[4]

Obr. 1.3 Pasivní stereoskopická projekce [4]

1.5 **Aktivní stereoskopická projekce**

Diváci sledují obraz, který se promítá na plátno (nebo i monitor či televizor) s dvojnásobnou snímkovou frekvencí, přičemž na filmovém pásu jsou střídavě proložené obrazy pro levé a pravé oko. Elektronické brýle diváka se dálkově (většinou s pomocí IrDA paprsku, nebo kabelem) synchronizují se zdrojem vysílání a střídavě zatmívají levé nebo pravé oko. Výsledkem je, že každý lichý snímek vidí divák jedním okem a každý sudý okem druhým. Tímto systémem se sice sníží frekvence promítaných obrazů na každé oko na polovinu, nicméně každé oko návštěvníka kina dostává pouze správně předurčený obraz. Z dvojice oddělených snímků mozek následně skládá skutečnou trojrozměrnou scénu (viz Obr. 1.4). Takovýto systém využívá například i 3D kino IMAX SOLIDO, nebo domácí stereo sestavy. Výhodou je, že k této projekci není potřeba žádného speciálního projekčního plátna nebo monitor (monitor musí umět zobrazovat obraz o vyšších obrazových frekvencích).[5]

Obr. 1.4 Aktivní stereoskopie [5]

1.6 **CAVE**

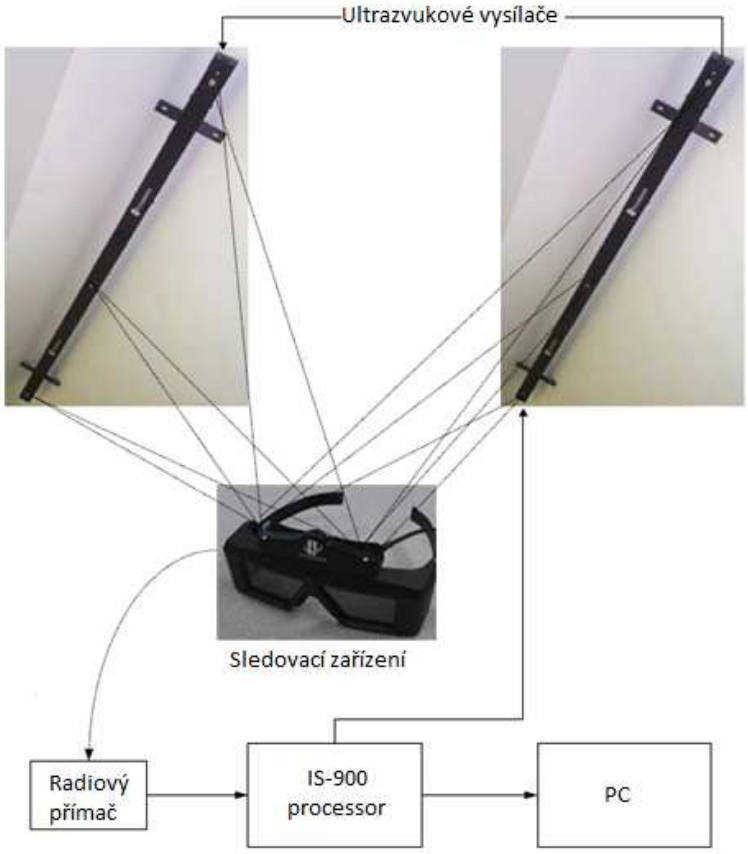
CAVE (Computer Aided Virtual Environment) neboli jeskyně je ohraničený prostor, rám na kterém jsou umístěna plátna. V tomto prostoru vzniká 3D projekce. Podle počtu pláten dostává pozorovatel větší prostorový dojem (viz Obr. 1.5).

Obr. 1.5 CAVE (zdroj www.worldviz.com)

1.7 **Trackovací zařízení**

Trackovací zařízení může být umístěno na CAVU a slouží k interaktivnímu pohybu ve virtuálním modelu. Trackovací zařízení snímá pohyb vysílače a převádí tato data do počítače. Díky tomuto zařízení obraz reaguje na naše pohyby a umožňuje nám procházet nebo hýbat s virtuálními objekty. V dále popisované práci bude využíván systém IS-900, který si níže popíšeme (viz Obr. 1.6).

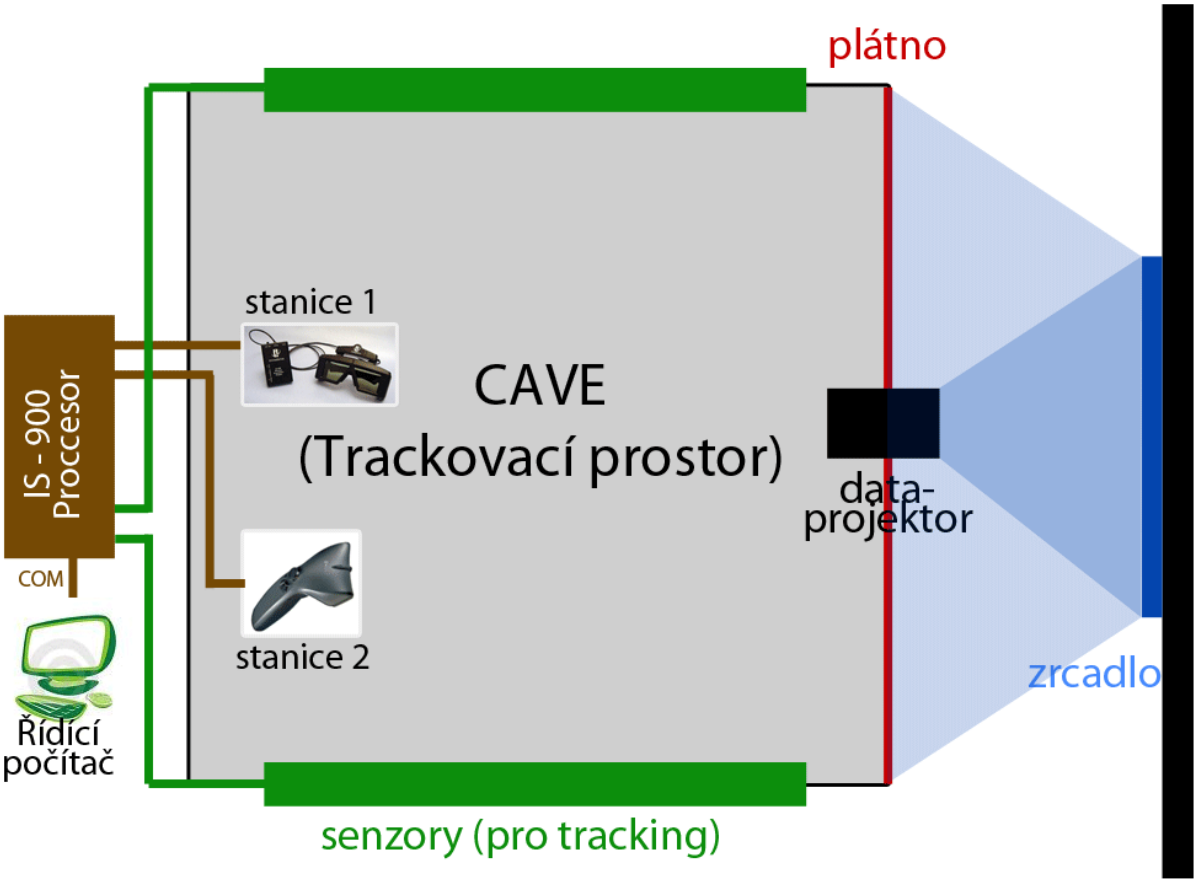
Systém IS-900 má 6 stupňů volnosti pohybu. Inertní komponenty se používají ke sledování pozic a orientací, zatímco ultrazvukový rozsah měření se používá pro korekce prostřednictvím pokročilého Kalmanova filtru. Výpočty sledovacích dat se přenáší do IS-900 procesoru. Ultrazvukové vysílače vysílají 40 kHz kmity, které jsou slyšet v mikrofonech v MiniTrax sledovacím zařízení, které je umístěno na konstrukci CAVU.[6]



Obr. 1.6 Trackovací zařízení [6]

2 Popis hardwarových možností laboratoře virtuální reality

Laboratoř virtuální reality na katedře KPV je založena na technologii 3D aktivní stereoskopické projekce. Laboratoř je jednoduchý jednoplátnový CAVE, na kterém je umístěno trackovací zařízení a projektor. Z důvodu dosažení maximálních rozměrů obrazu v limitně velké místnosti se obraz promítá na plátno přes zrcadlo umístěné na zdi, tomuto systému promítání se říká zadní projekce. Senzory trackovacího zařízení jsou umístěny nahoře po bocích rámu. Údaje z trackovacích senzorů jsou přenášeny do trackovacího procesoru, který je napojen na řídící počítač (viz Obr. 2.1).



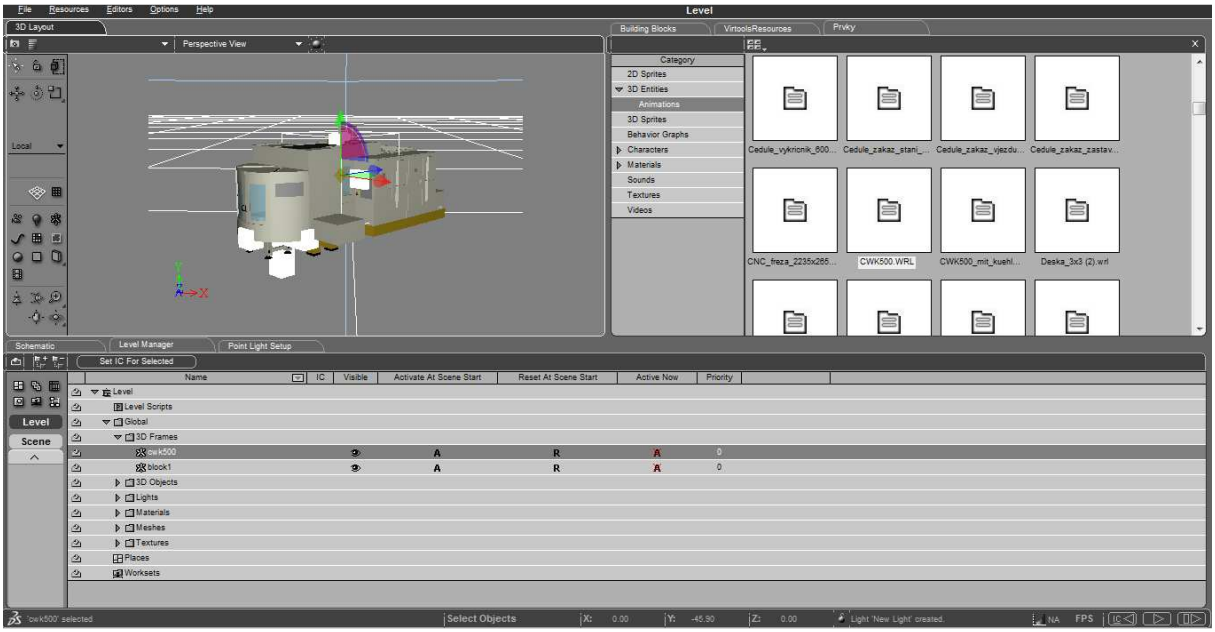
Obr. 2.1 Schéma uspořádání laboratoř virtuální reality na KPV [13]

Z programového vybavení laboratoře zmiňme alespoň systémy od společností Siemens a Dassault Systèmes.

3 Popis nástroje Virtools

V rámci digitální továrny a virtuální reality se práce zaměří na program Virtools 5 od francouzské společnosti Dassault Systèmes. Jedná se o vývojový program pro tvorbu, navrhování a simulace ve 3D . Program se používá pro tvorbu interaktivního prostředí. Do programu se dají přikládat další knihovny, díky kterým se možnosti programu zase zvýší (viz. Obr. 3.1).

3DVIA Virtools Platforma nabízí unikátní řešení pro vývoj a zavádění 3D na PC, herní konzole, intranety a na web. Architektura 3DIVA Virtools podporuje širokou škálu formátů 3D. 3D zásuvné moduly pro zachycení obsahu podporují většinu běžně používaných formátů DCC softwaru (3ds Max ®, Maya ®, XSI ®, Lightwave ®, Collada ®) pro import / export 3D souborů XML. 3DVIA Virtools je vývojový systém, používá 3D objekty jako jednotlivé komponenty nezávislé na datech spojenými s nimi. Výsledná architektura je flexibilní a umožňuje vývojářům modulární chování objektů a snadné pracování s nimi. Čas potřebný pro vývoj je dále snížen díky 3DVIA Virtools behavior library, obsahující více než 500 opakovaně použitelných stavebních bloků.[7]



Obr. 3.1 Prostředí programu Vitools 5

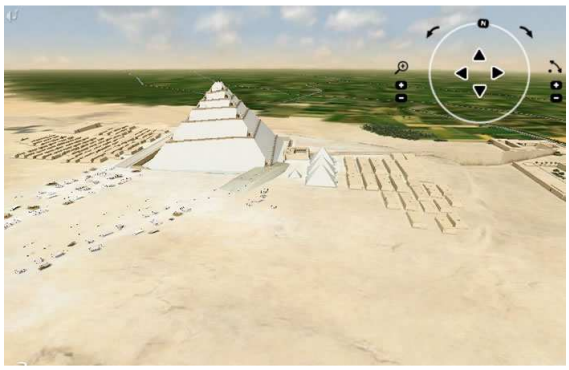
3.1 **Analýza současného stavu**

V této kapitole jsou ukázány příklady využití programu Virtools5 ve světě. Tento program se požívá v různých odvětvích např. letectví, automobilový průmysl, zdravotnictví, stavebnictví, tvorba her…

3.1.1 3D kino Géode

Francouzský archeolog Jean Pierre Houdin pomocí softwaru od Dassault Systèmes pomohl vrhnout světlo na 4500 let staré Khufuovo tajemství pyramid. Používá k tomu největší real-time 3D virtual reality hlediště ve světě, Géode v Paříži, Francie. Géode je

panoramatické kino ve středisku Cité des Sciences et le l´Industrie (Město vědy a průmyslu) ve čtvrti La Villette. Veliká koule o průměru 36 metrů stojí v bazénu, v jehož vodě se odráží a svítí do dálky svou hladkou ocelovou plochou. Plášť tvoří 6433 trojúhelníků z nerezové oceli. Promítací sál pojme 357 diváků, kteří spíše leží, než sedí v křeslech a program sledují na hemisférickém (půlkulovitém) plátně o ploše 1000 m2, jež tvoří perforovaný hliník, kudy vychází zvuk [11]. Dassault Systèmes real-time 3D řešení umožnilo architektu Jean-Pierre Houdinovi u modelu prozkoumat a spouštět simulace stavby pyramidy ve 3D. Seskupení sedmi počítačů se systémem 3DVIA Virtools obnovil stavbu Cheopsovi pyramidy přesně tak, jak to bylo o 4500 let dříve. Systém umožnil Houdinovi pohybovat virtuálním staveništěm a reagovat na dotazy publika. 3DVIA Virtools pomohl vyvinout aplikaci dynamicky, vytvářející jedinečný zážitek pro návštěvníky (viz. Obr. 3.2, Obr. 3.3). [8]



Obr. 3.2 Stavba Cheopsovy pyramidy [8]

Obr. 3.3 kino Géode [12]

3.1.2 Elektronická zábava

Umožňuje vytvářet komplexní, vysoce-kvalitní 3D hry v rekordním čase v 3DVIA Virtools. Program 3DVIA Virtools dokáže vytvářet různé žánry her: adventury, simulace, multiplayer a další. 3DVIA Virtools hladce sloučí všechny výrobní procesy pro efektivní týmové práce mezi vývojáři, návrháři hry a grafikou.[9]

3.1.3 PSA Peugeot Citroën-Design Review

3DVIA Virtools technologie jsou pro PSA Peugeot Citroën nedílnou součástí jeho nové Automotive Design centra nacházejícího se ve Velizy, Francie. 3DVIA Virtools Software Suite je používán k rozvoji a použití aplikací pro celou skupinu. PSA jedinečné prostředí se skládá z 11 genlocked a swaplocked PC (HP pracovní stanice vybavené ORAD DVG-X technologii a Nvidia grafickými kartami). Počítače jsou připojeny k Barco's I-Space 5 s aktivními stereo a AR trackovacím zařízením. 3DVIA Virtools poskytuje globální vývoj a nasazení platformy pro vytváření v širokém spektru průmyslu-specifických aplikací PSA: design , montáž / demontáž , spolupráce na projektech a další. (viz.Obr. 3.4 Obr. 3.4) [10]

Obr. 3.4 Využití Virtools v PSA

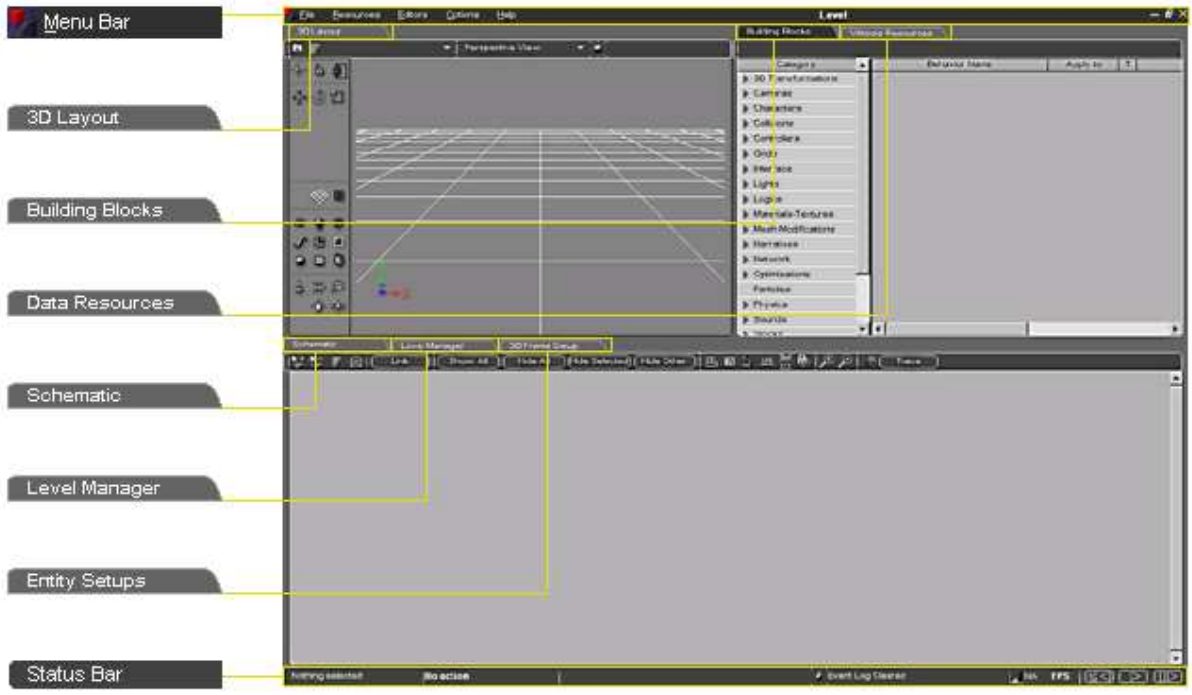
4 Popis implementace

V této části práce se seznámíme s prostředím programu Virtools a jeho členěním. Dále jsou zde vysvětleny a popsány základní operace v programu.

4.1 Popis prostředí Virtools

3DVIA Virtools 5 zahrnuje uživatelské rozhraní k rozvoji sofistikovaných aplikací pro sestavení objektů a chování, knihovny chování a objektů, Render Engine vykreslení grafiky v realném čase, Virtools skriptovací jazyk pro vytvoření specifických funkcí (viz Obr. 3.1). Scény jsou synchronizovány čtením parametrů v hlavní scéně a jejich přenosem do klientské scény. Příslušné parametry pro synchronizaci jsou určeny vývojářem aplikací, a jsou přístupné v schematickém editoru 3DVIA Virtools5. 3DVIA Virtools5 proces tvorby VR prostředí usnadňuje vytváření prototypů a komplexní vývoj.

Program Virtools, je rozdělen do částí: grafická část (3D Layout), stavební bloky (Building Blocks), knihovna dat (Data Resources) správce úkolů (Level Manager), nastavení entit (Entity setaps), stavová lišta (Status Bar) a část pro skládání stavebních bloků (Schematik), (viz. Obr. 4.1).



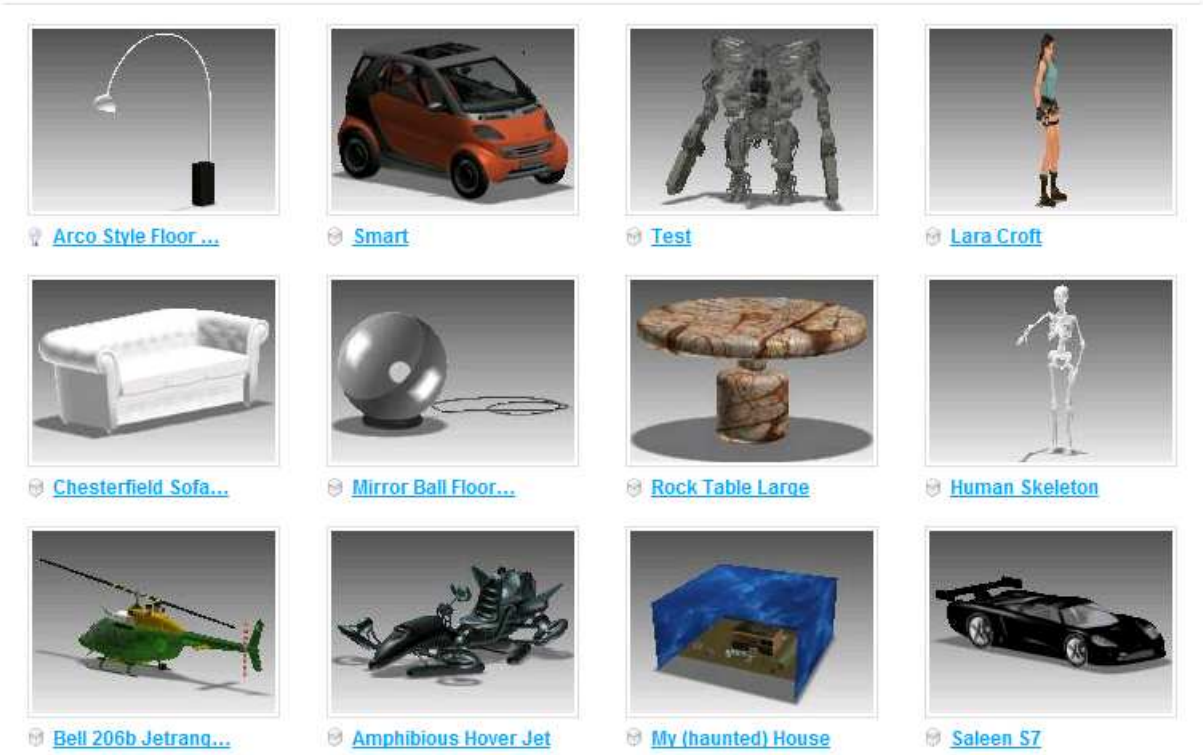
Obr. 4.1 Prostředí programu Virtools 5

- V **grafické části** je náhled na vytvořené prostředí a objekty. Dají se zde nastavit různé pohledy, přibližovat a oddalovat. Po označení objektu se zobrazí osový kříž, při uchopení jedné osy je možné s objektem pohybovat ve směru os a natáčet. Vedle grafiky je zde i pár tlačítek, díky nimž můžeme vytvářet osvětlení, křivky, 3D kříže, 2D kříže, kamery, plochy, textury, materiály a videa. V horní části se nacházejí 3 ikonky, první z nich ve tvaru fotoaparátu slouží pro tvorbu snapshotů (uloží aktuální pohled jako obrázek)

- Vedle grafické části jsou umístěny **knihovny**. V programu jsou již předinstalovány knihovny pro programování a knihovna objektů. Knihovnu si můžeme vytvořit i sami s námi zvolenými objekty. Do programu se dají vkládat objekty ve formátech:

  3D Objekty: 3D XML, 3ds Max®, Maya®, XSI®, Lightwave®, Collada®, WRL.
  Obrázky: JPG, PNG, TIFF, TGA, BMP, PCX.
  Zvuky: MP3, WMA, WAV, MIDI.

  Objekty pro vkládání do programu je možné si vytvořit v nějaké kreslícím programu, nebo si stáhnout již hotové z internetu, například zadarmo přímo ze stránek www.3dvia.com, kde si musíte vytvořit svůj účet (viz. Obr. 4.2). Poté si už jen zvolíte formát, ve kterém ho chcete uložit, ale program Virtools podporuje všechny 3 zde zobrazené formáty.



Obr. 4.2 Ukázka objektů ke stažení [15]

- **Správce úkolů** slouží jako seznam vložených a vytvořených věcí. Je rozdělen do několika kategorií, podle druhu objektů. Každý vložený prvek se zde zobrazí a po jeho zvolení se nám objeví nastavení tohoto objektu a označí se v grafické části, aby bylo vidět který objekt je to. Dále se zde dá nastavit viditelnost, jestli bude prvek aktivní po startu scény, nebo jestli bude aktivní teď a dá se zde nastavit priorita. V levé části správce jsou umístěny ikony pro vytváření scény, skript, oblastí, ploch, skupin a pracovního nastavení.

- Část pro **tvorbu skriptů**, zde se propojují jednotlivé stavební bloky, které přiřazují zvolenému objektu určité vlastnosti (posuv, rotace, zvuky, videa…).

4.2 **Základní operace v programu**

Zde jsou sepsány postupy, jak vkládat a vytvářet projekty v programu Virtools5.

4.2.1 Vkládání objektů

Objekty jsou definované prvky, které byly předem vytvořené např. stroje, nástroje, zařízení, lidi. Jednotlivé objekty lze vkládat i bez vytvoření knihovny.

- V záložce *Resources* zvolíme položku *Import File As*, zde vybereme, jaký druh objektu chceme vložit.
- Vyhledáme, kde je objekt uložený a otevřeme jej
- Nyní je prvek vložen do programu

4.2.2 Vytvoření knihovny objektů

- V záložce *Resources* vybereme položku *Create New Data Resources*
- Zvolíme umístění knihovny a její název
- V knihovně se vytvoří jednotlivá oddělení podle druhu objektu, která tam chceme vkládat
- Do zvoleného oddělení vložíme prvky
- Knihovna se objeví jako nová záložka, vedle knihovny programu

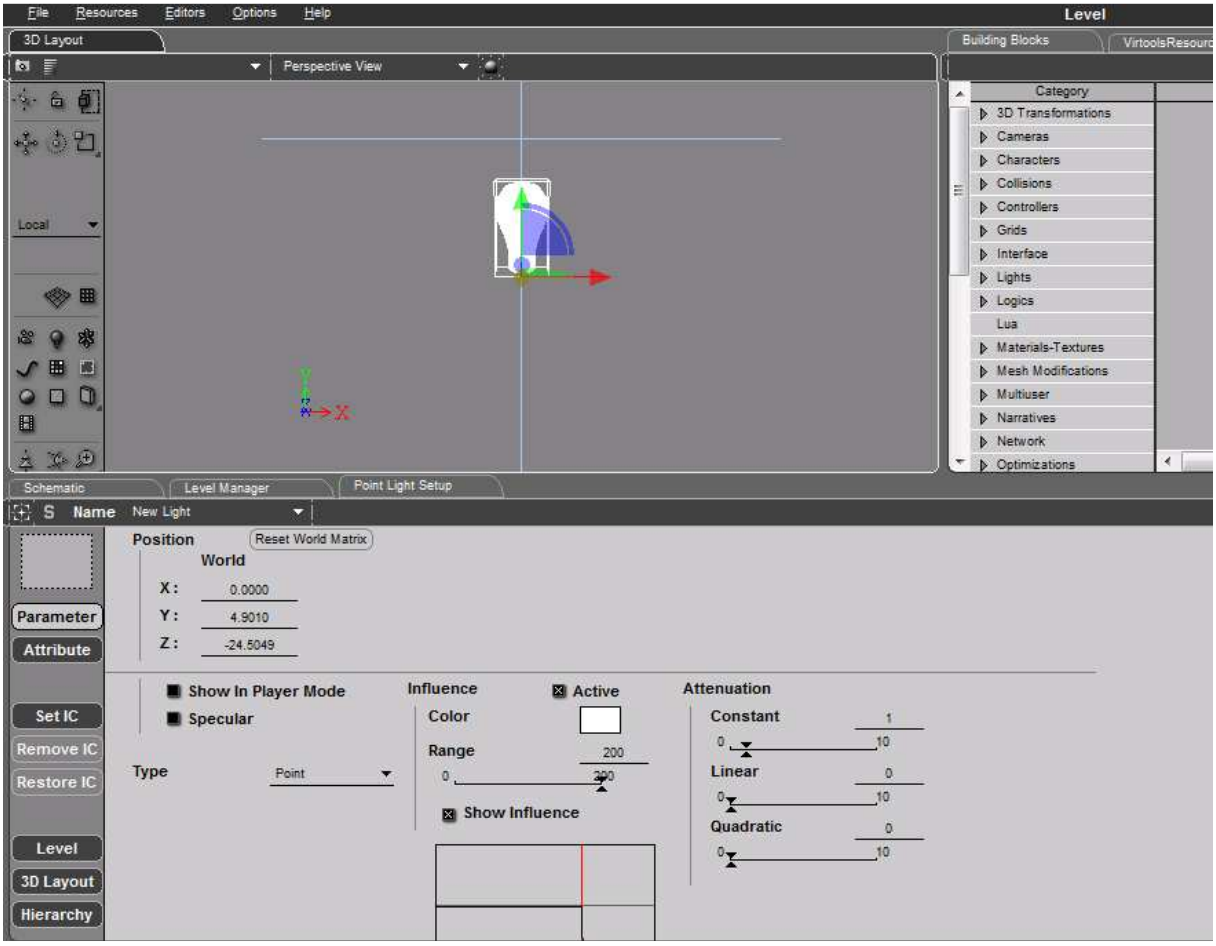
4.2.3 Umisťování objektů

Když máme objekt vložený do programu, můžeme s ním pohybovat, buď v grafické části, nebo pomocí souřadnic.

- Pohybování v grafické části se provádí pomocí souřadnicového kříže. Podle toho jakým směrem chceme s objektem pohybovat, uchopíme danou souřadnicovou osu.
- Pomocí souřadnic se s objektem pohybuje tak, že otevřeme v dolní části záložku *Level Manager* v té si rozevřem položku *Global* a zde zvolíme *3D Frames.* Podle názvu objektu nalezneme správný 3D kříž objektu, ten dvojklikem otevřeme a zobrazí se panel *3D Frame Setup*. Zde jsou tři možnosti, pozice, orientace a měřítko. V položce pozice upravujeme umístění objektu, v orientaci otáčíme objektem a v měřítku zvětšujeme, nebo zmenšujeme objekt.

4.2.4 Osvícení plochy

Program je ze začátku nastaven bez světla, které je tam nutno vložit, jinak je vše hodně tmavé. U světla lze nastavit jeho polohu a velikost osvícené plochy. Světlo se vkládá malou ikonkou žárovky po levé straně grafické části. Se světlem lze hýbat jako s jinými objekty. (viz. Obr. 4.3)

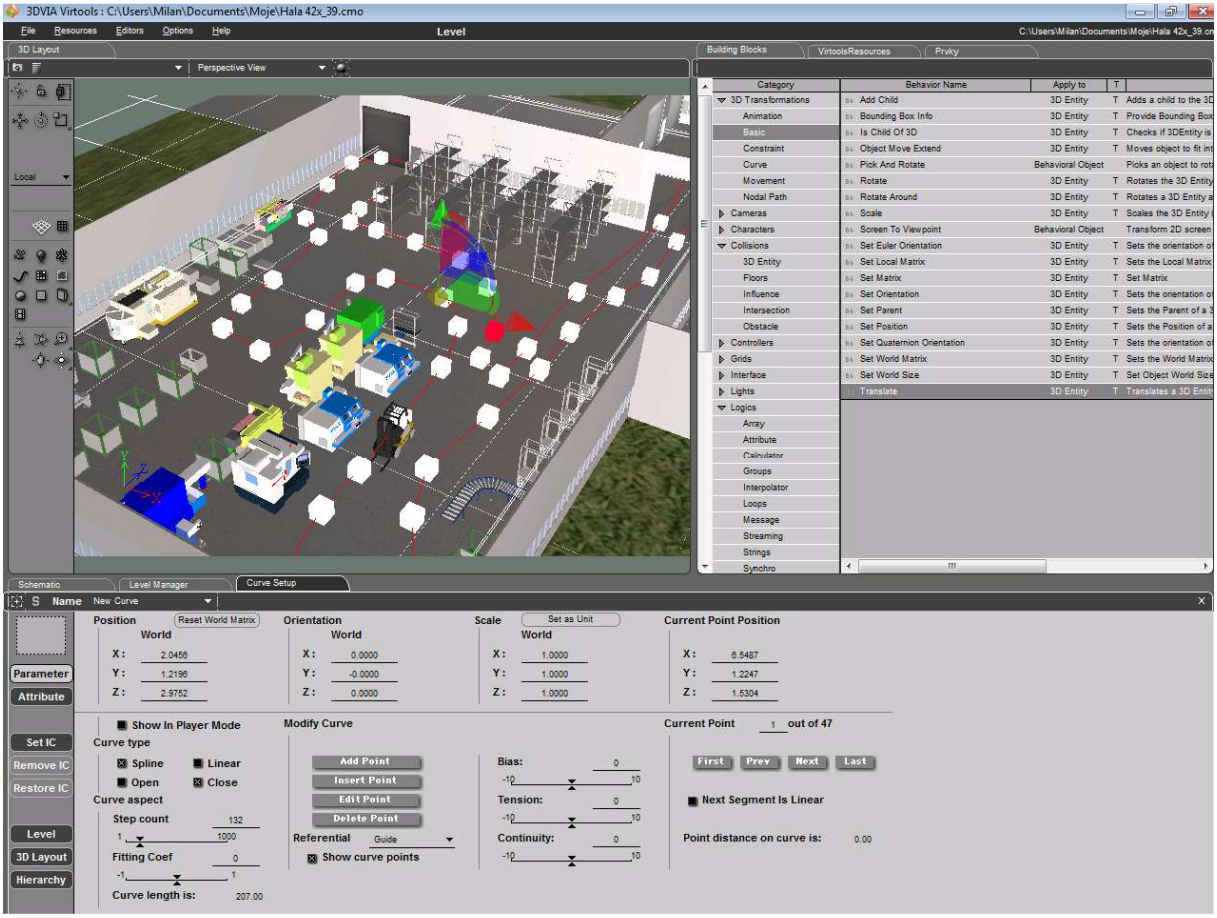


Obr. 4.3 Osvícení plochy

4.2.5 Tvorba křivky

Křivka slouží pro další práci v programu Virtools, nejčastěji pro určení dráhy pohybu objektů. Skládá se z jednotlivých bodů, které jsou zvoleny uživatelem. S body se dá pohybovat pomocí souřadnicových os, nebo pomocí zadání souřadnic jednotlivých bodů. Křivka se dále dá nastavit, jestli bude uzavřená (close) nebo otevřená (open). Nastavuje se propojení jednotlivých bodů, jestli bude zaoblená (spline), nebo spojena lineárně, tedy pomocí úseček. Po vytvoření křivky se s ní dá pohybovat s celou jako jedním prvkem (viz Obr. 4.4).

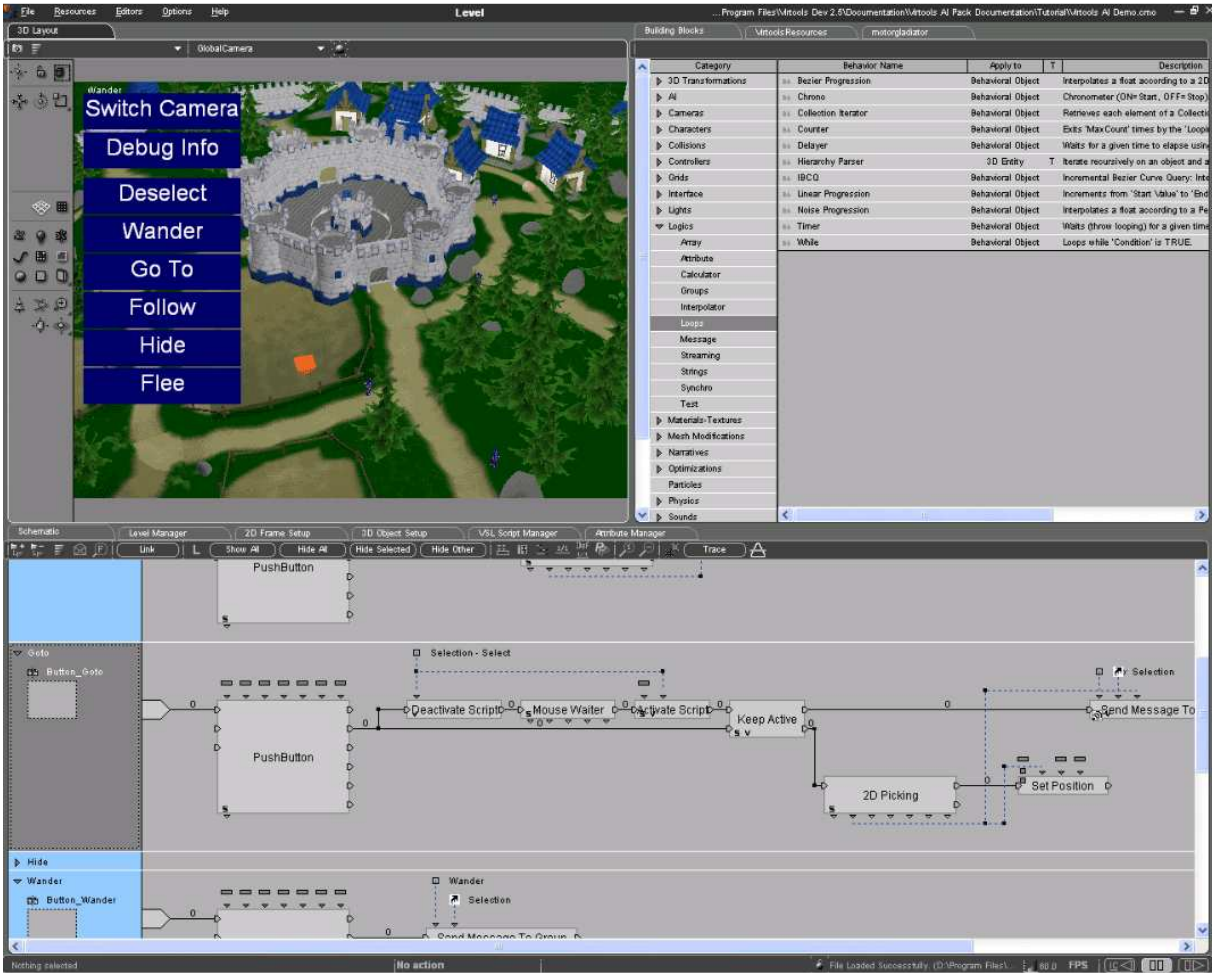
- Nejprve se upraví pohled, aby byla vidět plocha, na které bude křivka umístěna.
- Vlevo vedle grafické části je ikonka pro křivky  . Po jejím zmáčknutí se objeví první výchozí bod křivky umístěný do souřadnice [0;0;0] a dole se otevře pole nastavení křivky.
- V nastavení křivky v oddělení modify Curve si nastavíme, jestli chceme přidávat další bod, vybrat bod, upravit bod nebo smazat bod.
- Po vytvoření křivky, se může s ní pohybovat a upravovat její tvar.



Obr. 4.4 Vytvořená křivka

4.3 **Tvorba skriptů v programu Virtools**

V programu Virtools je dále možné i vytvářet sestavy stavebních bloků pro chování jednotlivých objektů. Pro skládání slouží část s názvem Schematic, zde se vytváří daná skripta spojováním jednotlivých stavebních bloků, které přidávají objektu určité vlastnosti. Program Virtools obsahuje již základní knihovnu těchto stavebních bloků, z kterých se dá sestavit program. Každý stavební blok má vstup a výstup, ale může jich mít i více pro specifické funkce bloku. Jednotlivé Stavební bloky se spojují pomocí vazeb, uchopí se výstupní bod a propojí se vstupním bodem (viz. Obr. 4.5).

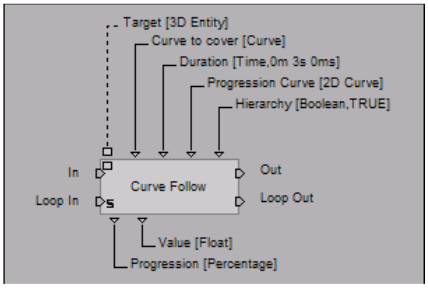


Obr. 4.5 Ukázka programování v programu Virtools [14]

Skript je vizuální reprezentace chování, použitá na prvku, který je zastoupen ve schématu. Skript se skládá ze dvou částí - hlavičky a těla. V záhlaví se zobrazí název a tvůrce tohoto skriptu, a volitelně malý snímek. Tělo skriptu je složeno ze začátku (výchozí bod) a jednoho nebo více BBs, BGs, paramOps, Parameters, bLinks, pLinks, comments, etc.

4.4 **Stavebních bloky – Building Blocks**

Curve Follow - následování křivky (viz. Obr. 4.6).

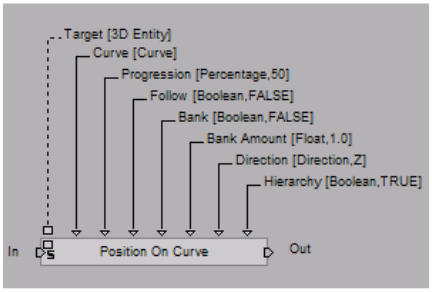


Obr. 4.6 Curve Follow

Přesouvá 3D entity podél cesty.

In: spouští proces.
Loop In: spouští další krok v procesu.
Out: aktivuje se, když je proces ukončen.
Loop Out: aktivuje se, když proces potřebuje smyčku.
Curve to cover: následování 3D křivky.
Duration: jak dlouho by měl celý proces trvat.
Progression curve: 2D křivka představující progresi 3D entit podél jeho pohyb.
Hierarchy: pokud je TRUE, pak se tento stavební blok bude vztahovat i na podskupiny 3D entitní jednotky.
Progression: procento mezi 0% a 100%, který definuje průběh procesu. Start = 0%, střední doba = 50%, konec = 100%. Užitečné, pokud potřebujete interpolovat ve stejnou dobu (ve stejné smyčce), barevná hodnota, vektor, orientace nebo něco takového.
Value: aktuální Bézierovo-interpolované hodnoty.
Time Based: Je-li zaškrtnuto, bude tento stavební blok závislý na čase a ne na snímku. Vytvořením tohoto stavebního bloku, má časově závislost výhodu, že složení bude probíhat se stejnou sazbou pro všechny konfigurace počítače.

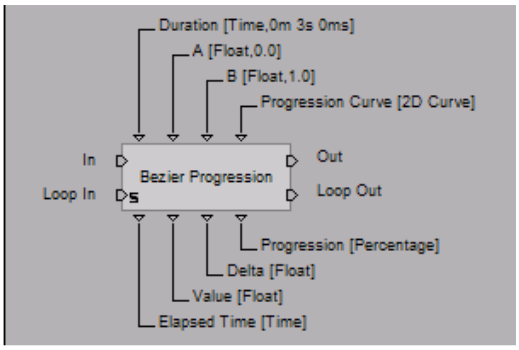
Position On Curve - umisťuje 3D entity na specifikovanou pozici na křivce (viz. Obr. 4.7).



Obr. 4.7 Position On Curve

In: spouští proces
Out: je aktivován, když je proces ukončen.
Curve: 3D křivky, na kterých by měli být 3D entity umístěny.
Progression: progrese na křivce v procentech (0% = start / 100% = konec).
Follow: pokud je TRUE, bude orientace 3D entita bude bod ve směru tangenty křivky.
Bank: pokud TRUE, 3D entita bude naklánět na křivce.
Bank Amount: výše naklápěcího účinku (1 je standardní naklonění, 2 a vyšší hodnoty pro zvýšení účinku).
Direction: Určuje, které základní vektor má být tímto směrem (v případě, že ' Follow ' parametr je nastaven na hodnotu TRUE).
Hierarchy: pokud je TRUE, pak se tento stavební blok bude vztahovat i na podskupiny 3D entitní jednotky.
Roll: určuje počet otáček a úhel pro 3D entitu při výkonu.

Bezier Progression - Interpoluje oběh podle 2D Bézierovy křivky v [Min, Max] rozsahu, za daný počet milisekund (viz. Obr. 4.8).



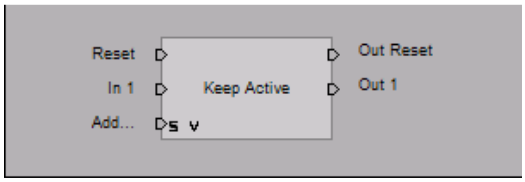
Obr. 4.8 Bezier Progression

In: spouští proces.
Loop in: spouští další krok v procesu.
Out: je aktivován, když je proces ukončen. POZNÁMKA: Výstup ven by měl být spojen s uzavřenými stavebními bloky od posledního kroku progrese který je vlastně výstup přes Out, ne smyčka ven.
Loop out: aktivuje se, pokud proces potřebuje mít smyčku.
Duration: jak dlouho by celý proces měl trvat, nebo počet snímků: počet kroků, na které se vztahuje celá řada.
A: dolní konec rozsahu, který se vztahuje k Bézierově-interpolované hodnotě.
B: horní konec rozsahu, který se vztahuje k Bézierově-interpolované hodnotě.
Progression Curve: 2D křivka představující průběh výstupní hodnoty.
*Elapsed Time:*čas, který uplynul od začátku progrese, nebo aktuální snímek: aktuální krok k dalšímu rozvoji.

Hodnota: aktuální Bézierovi-interpolované hodnoty.
Value: odčítání mezi současnou Bézierovou-interpolovanou hodnotou a tou předchozí.
Progression: procento mezi 0% a 100%, který definuje průběh procesu. Start = 0%, střední doba = 50%, konec = 100%. Užitečné, pokud potřebujete interpolovat ve stejnou dobu (ve stejné smyčce), barevná hodnota, vektor, orientace nebo něco takového.
Time Based: Je-li zaškrtnuto, bude tento stavební blok závislý na čase a ne na snímku. Vytvořením tohoto stavebního bloku, má časově závislost výhodu, že složení bude probíhat se stejnou sazbou pro všechny konfigurace počítače.
Tip: Vzhledem k tomu že 'Progression' hodnota je uvedena jako výstupní parametr, můžete jej použít ve smyčce o při interpolaci orientace objektu ve stejnou dobu (s 'Interpolator Orientace' stavebními bloky například).

Tento stavební blok může být použit k transformaci lineární progrese Bézierovy křivky do progrese v [A, B] rozsahu.

Keep Active - udržuje proces aktivní, i když vstup byl aktivován pouze jednou (viz. Obr. 4.9).

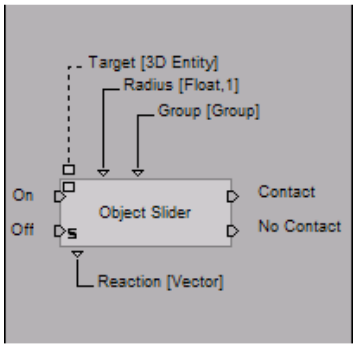


Obr. 4.9 Keep Active

Reset: obnoví stavební bloky do původního stavu (deaktivuje všechny vstupy).
In 1: vstup signálu do paměti.
Out Reset: je aktivován, když je stavební blok obnoven (resetován).
Out 1: Konstantní výstupní signál odpovídající vstupnímu signálu.
One at a time: určuje, zda některé výstupy mohou být aktivovány současně, nebo ne. Pokud může být aktivován pouze jeden výstup, pak je to ten který odpovídá poslednímu aktivnímu vstupu.

Tento stavební blok si zapamatuje vstupní údaje a aktivuje výstupy u každého snímku, i když je vstupní signál zastaven.

Object Slider - zabraňuje pronikání 3D objektům (viz. Obr. 4.10).

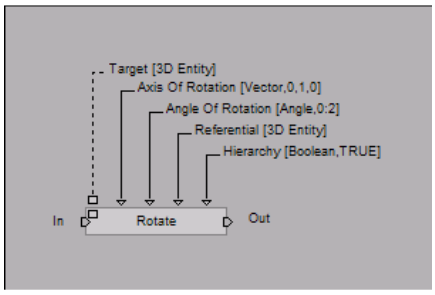


Obr. 4.10 Object Slider

On: aktivuje proces.
Off: vypne proces.
Contact: aktivuje se pokaždé, když je kontakt.
No Contact: aktivovaný pokaždé když není žádný kontakt.
Radius: poloměr bránicí pronikání objektu (vyjádřen v absolutní hodnotě).
Group: skupina, v níž jsou obsaženy i jiné předměty (objekt [ball] nepronikne do těchto objektů).
Reaction Vector: 3d vektor reakce (nenormalizovaný).
Accuracy: maximální přesnost pro kolize objektu.
Touched Objects Group: skupina všech objektů, kterých se dotkl při kontaktu.
*Place Optimization:*je-li TRUE, bude tento proces optimalizován pro úrovně navrženou v místě. Namísto analýzy všech objektů ve skupině, a pak provedení testu pro každý z nich. Namísto toho se provede analýza všech míst a pro všechna místa se provede hierarchický test. Pokud je test TRUE, potom se analyzuje místo hierarchie rekurzivním způsobem, a pro každou

podskupinu v místě provádíme opět hierarchický ohraničení kolize, test… a tak dále.

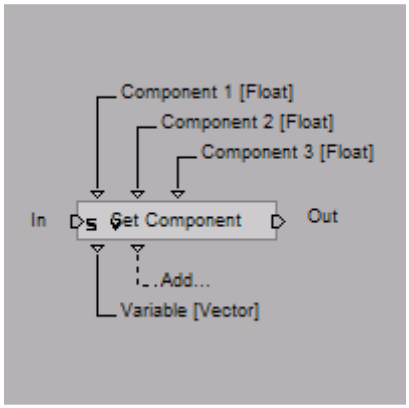
Rotate 3D entit. Tento stavební blok vytváří otáčení 3D entit kolem své osy (viz. Obr. 4.11).



Obr. 4.11 Rotate

In: spouští proces
Out: je aktivován, když je proces ukončen.
*Axis Of Rotation:*vektor popisující orientaci os.
*Angle Of Rotation:*úhel ve stupních a počet závitů.
Referential: slouží k definování směru osy otáčení.
Hierarchy: pokud je TRUE, pak se tento stavební blok bude vztahovat i na podskupiny 3D entitní jednotky.

Set komponent – přiřazuje parametry do procesu (viz. Obr. 4.12).



Obr. 4.12 Set Component

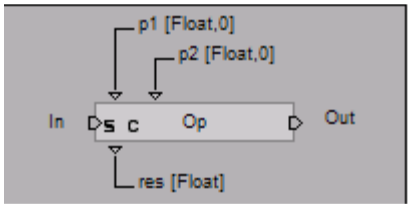
Skládá COLOR, VECTOR nebo 2D VECTOR, pro určením jednotlivých komponent.

In: spouští proces.
Out: je aktivován, když je proces ukončen.
Component 1: první složka (float).
Component 2: druhé složky (float).
Component 3: třetí složky (float).
Variable: složený z COLOR, VECTOR, EULER ANGLES, VECTOR 2D, RECTANGLE, BOUNDING BOX nebo STRUCTURE.

Rectangle/Box Mode: v případě, že výstupní typ parametru je RECTANGLE nebo BOUNDING, pak toto nastavení definuje, jak by měly být tyto informace uvedeny (centrum / polovina-velikost ... nahoru, vlevo (min) koutek / dole / vpravo (max) rohu ... nebo nahoře, vlevo (min) roh / velikost).

Tento stavební blok se skládá buď z COLOR, VECTOR, EULER ANGLES, VECTOR 2D, RECTANGLE, BOUNDING BOX nebo STRUCTURE jejich součásti .
např.: Řekněme že 'Proměnná (Variable)' je vektor a
'Component 1' = 8.
'Component 2' = 3.
'Component 3' = 5.
Pak ‘Proměnná' = [8,3,5].
Pokud změníte typ 'Proměnná', pak se (v případě potřeby) počet vstupních parametrů změní automaticky.

Op -blok slouží k operacím s různými typy vstupů (viz. Obr. 4.13).

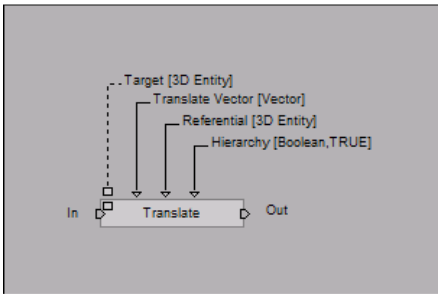


Obr. 4.13 Op

Proces pro jakékoliv parametry Operace.

In: aktivuje proces.
Out: se aktivuje, když je proces dokončen.
p1: první argument operace.
p2: druhý argument operace.
res: výsledek operace.

Translate- blok, který udává posuvný pohyb ve směru vektoru rychlosti (viz. Obr. 4.14).



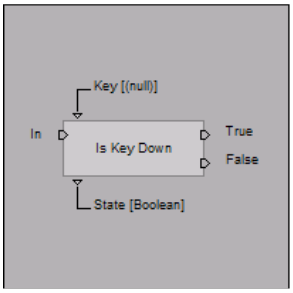
Obr. 4.14 Translate

Převádí 3D entity

In: spouští proces
Out: je aktivován, když je proces ukončen.

*Translate Vector:*3D Vector.
Referential: souřadnicový systém, v němž je vyjádřena Translate Vector.
Hierarchy: pokud je TRUE, pak se tento stavební blok bude vztahovat i na podskupiny 3D entitní jednotky.

Is Key Down- slouží k jednoduchému ovládání programu pomocí klávesnice (viz. Obr. 4.15).

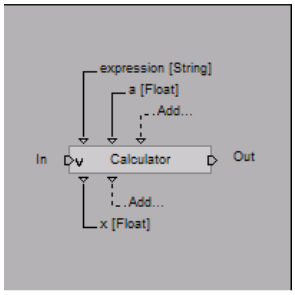


Obr. 4.15 Is Key Down

Zjišťuje, zda je klávesa stisknuta nebo ne.

In: spouští proces.
True: je aktivován, když je vybraná klávesa stisknuta.
False: je aktivován, když není vybraná klávesa stisknuta.
Key: Přiřazení klávesy.
State: PRAVDA, pokud je vybraná klávesa stisknuta, FALSE není vybraná klávesa stisknuta.

Calculator- je blok do, kterého si zapíšeme pomoci matematických výrazů rovnici (viz. Obr. 4.16).

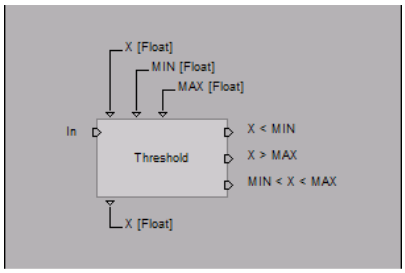


Obr. 4.16 Calculator

Vypočítá hodnotu aritmetického výrazu, 'a' pro první parametr, 'b' pro druhý, atd. ...

In: spouští proces.
Out: je aktivován, když je proces ukončen.
Expresion: aritmetický výraz.
A: první hodnota.
B: druhá hodnota.
C: třetí hodnota.
D: ... atd.
X: výsledek.

Threshold- blok, který nastavuje hranice pro vstupní parametr (viz. Obr. 4.17).

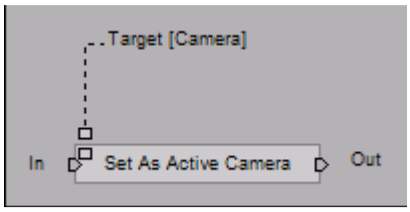


Obr. 4.17 Threshold

Udržuje proměnné mezi dvěma limity (Min a Max).

In: spouští proces.
X <Min: se aktivuje, pokud x <min.
X> Max: je aktivován, pokud x> max.
Min <X <Max: je aktivován, pokud x ≥ min a x ≤ max.
X: hodnotu vstupu.
Min: minimální hodnota.
Max: maximální hodnota.

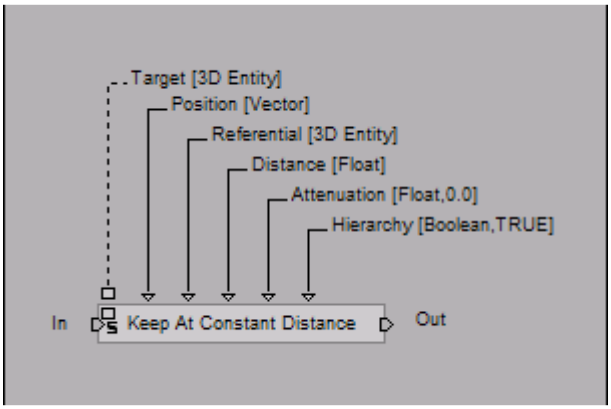
Set As Active Camera – nastavení aktivní kamery (viz. Obr. 4.18).



Obr. 4.18 Set As Active Camera

In: spouští proces.
Out: je aktivován, když je proces ukončen.
Target (Camera): námi zvolená kamera

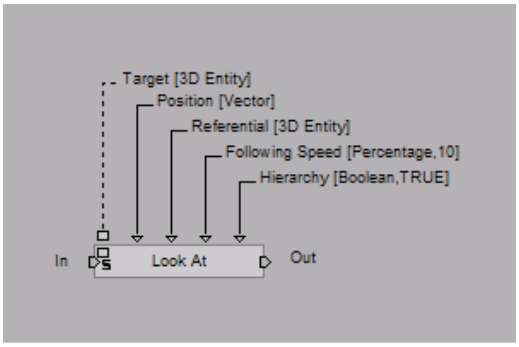
Keep At Constant Distance – vybraný objekt bude následovat daný objekt a bude udržovat konstantní vzdálenost mezi nimi (viz. Obr. 4.19).



Obr. 4.19 Keep At Constant Distance

In: spouští proces
Out: je aktivován, když je proces ukončen.
Position: vektor vyjádřený v 'Referential'.
Referential: následovaný 3D objekt.
Distance: udržovaná vzdálenost mezi 2 objekty.
Attenuation: prodleva mezi následováním prvního objektu druhým. Hodnota 0 (nula) by znamenala okamžitou odezvu.
Hierarchy: pokud je TRUE, pak se tento stavební blok bude vztahovat i na podskupiny 3D entitní jednotky.
Time Based: Pokud je zvoleno, bude toto chování závislé na čase, a ne na snímku. Díky tomuto stavebnímu bloku, časově závislá má tu výhodu, že složení bude probíhat se stejnou sazbu pro všechny konfigurace počítače.

Look At – natavení pozice bodu na Z ose (viz. Obr. 4.20).

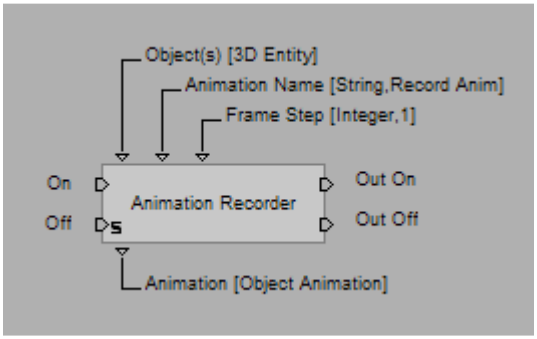


Obr. 4.20 Look At

In: spouští proces.
Out: je aktivován, když je proces ukončen.
Position: vektor pozice pohledu.
Referential: referenční 3D entity, od nichž je vyjádřena pozice.
Following Speed: jak rychle by se měla 3D entita podívat se na danou pozici. Čím vyšší procento, tím rychleji 3D entita se podívá.
Hierarchy: pokud je TRUE, pak se tento stavební blok bude vztahovat i na podskupiny 3D entitní jednotky.
Time Based: Pokud je zvoleno, bude toto chování závislé na čase, nikoli na snímku. Díky tomuto stavebnímu bloku, Time Based má tu výhodu, že složení bude probíhat se stejnou hodnotou pro všechny konfigurace počítače.

Direction: Určuje, který základní vektor je třeba na hledání směru.
Roll: určuje počet otáček a úhel nastavení pohledu pro 3D entity.

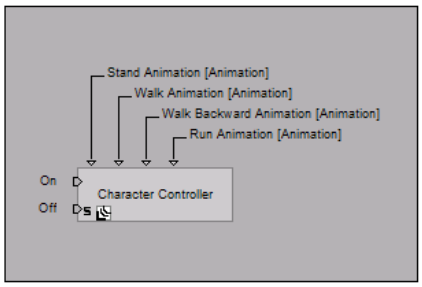
Animation rekord - záznam animace objektu, nebo skupiny objektů s uložením (pozice, orientace, měřítka), (viz. Obr. 4.21).



Obr. 4.21 Animation Recorder

On: začátek záznamu.
Off: konec záznamu.
Out on: aktivuje, když má začít nahrávat.
Out off: aktivuje, když má nahrávání skončit.
*Object(s):*objekt, který se zaznamenává.
Animation name: název vytvářené animace.
Frame step: záznam klíčových animací každý n obraz.
Animation: vytvářená/nahrávaná animace.

Character controller - jednoduché ovládání animací postavy prostřednictvím joysticku nebo klávesnice. (viz. Obr. 4.22)

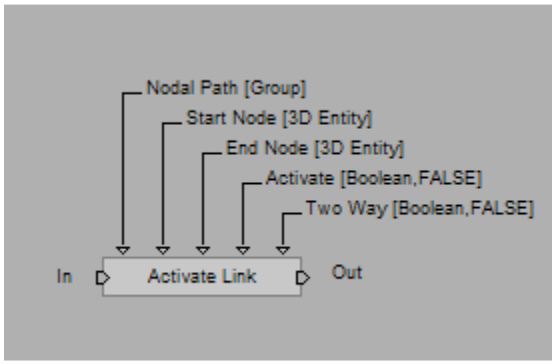


Obr. 4.22 Character controller

On: aktivuje proces.
Off: vypne proces.
Stand Animation: výchozí animace, běží, když žádná jiná animace není spuštěna.
Walk Animation: Animace chůze, když je stisknuto "Joy_Up '.
Walk Backward Animation: chůze pozpátku animace běží, když se drží "Joy_Down '.
Run animation: animace je spuštěna, když se drží "Joy_Button1 '.

Tento stavební prvek ovládá postavu pomocí animace, v reakci na zprávách z joysticku nebo klávesnice.

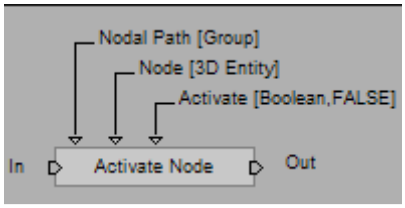
Activate link - aktivuje nebo deaktivuje spojení mezi dvěma uzly (viz. Obr. 4.23).



Obr. 4.23 Activate Link

In: spouští proces.
Out: aktivuje se, když je proces u konce.
Nodal path: uzlová cesta ke které spojení patří.
Start node: spojení začíná bodem.
End node: spojení končí bodem.
Activate: spojení bude aktivováno, když bude TRUE, jinak bude spojení deaktivováno.
TwoWay: stavební bloky budou použity v obou směrech, když bude TRUE a dvojcestné spojení.

Activate node - aktivuje nebo deaktivuje uzly (viz. Obr. 4.24).



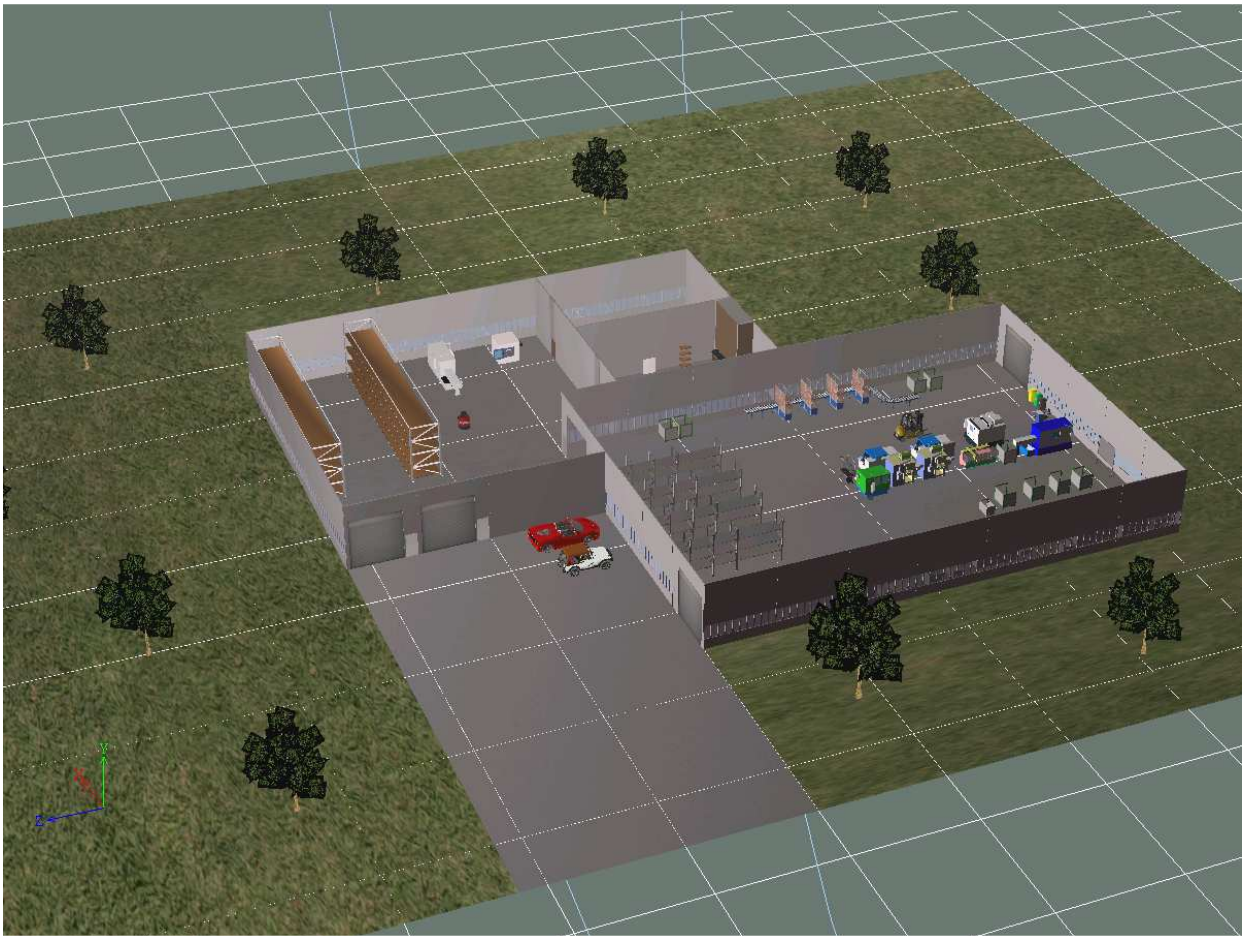
Obr. 4.24 Activate Node

In: spouští proces.
Out: se aktivuje, když proces skončí.
Nodal path: uzlová cesta kam uzel patří.
Node: uzel reprezentující objekt je aktivován nebo deaktivován.
Activate: uzel který bude aktivován když TRUE, jinak bude uzel deaktivován.

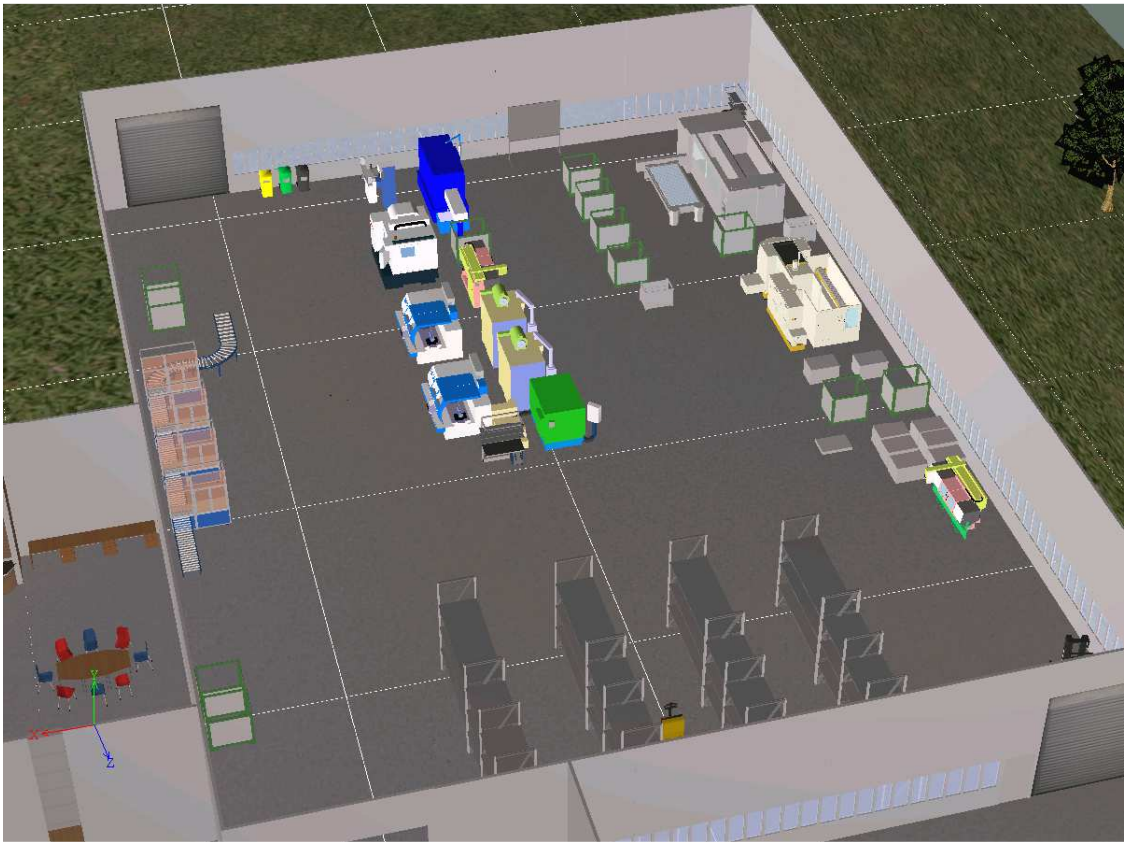
5 Tvorba výrobní haly v programu Virtools

V programu Virtools 5 jsem si nejprve vytvořil knihovnu objektů, ve které mám uloženy stroje, zdi, pracoviště, nábytek atd. Za použití této knihovny jsem začal vytvářet vlastní výrobní halu, která byla dále doplněna o objekty stažené z oficiálních stránek. [15]

Celkový pohled na halu, zde je vidět struktura haly. Hala je rozdělena na dvě hlavní části, na výrobní část, v které jsou umístěny výrobní stoje a sklad hotových výrobků. V druhé části je sklad materiálu a součástí pro výrobu, kancelář a šatny (viz. Obr. 5.1).

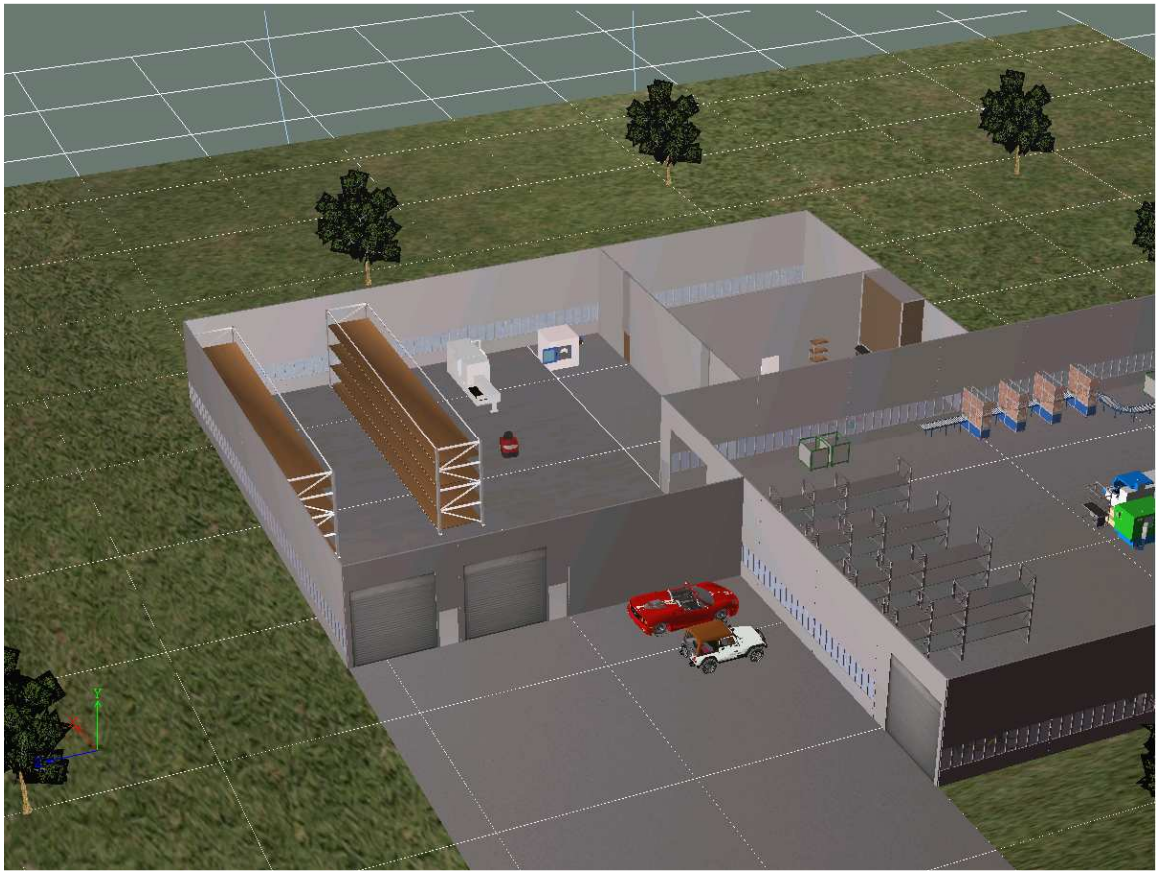


Obr. 5.1 Hala celkový pohled

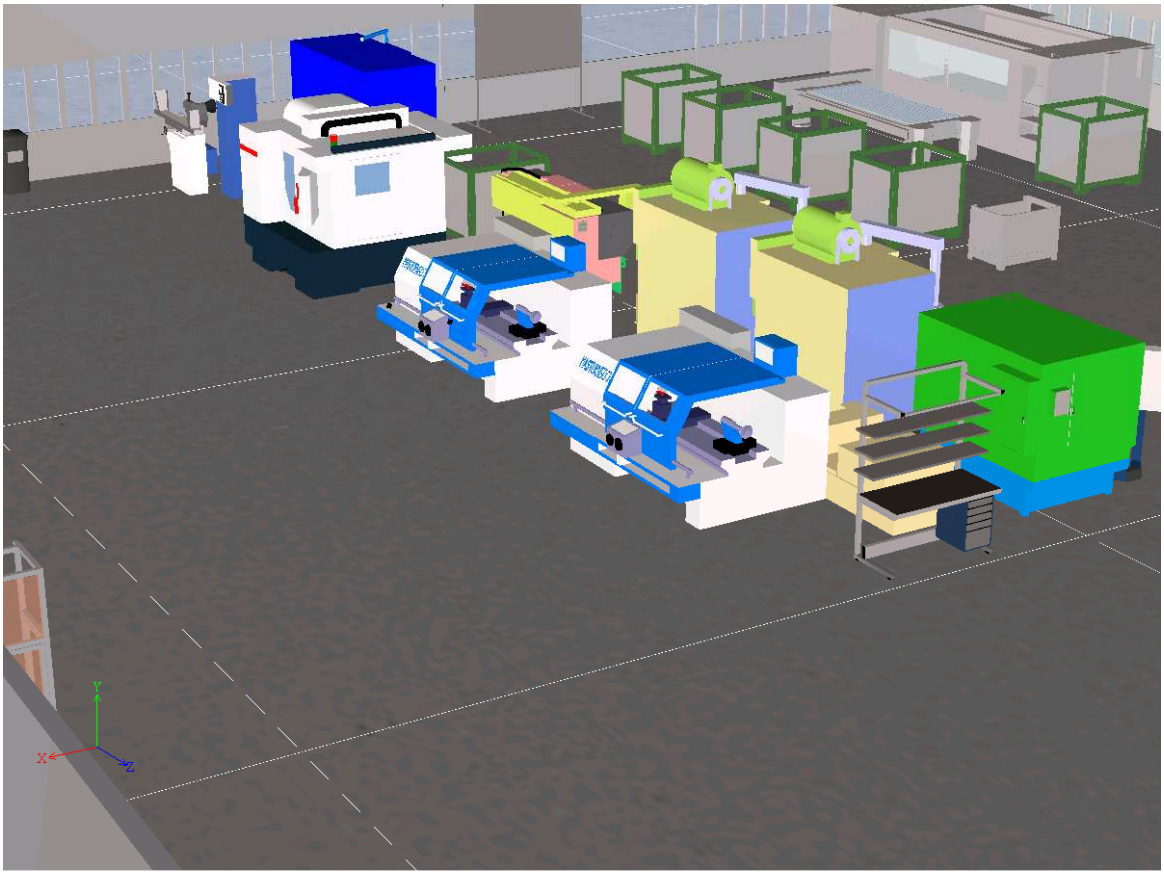


Obr. 5.2 Hala - výrobní část

Uspořádání výrobních strojů a montovací linky ve výrobní hale (viz. Obr. 5.2) a pohled na skladovou část pro ještě nezpracovaný materiál a sklad součástí (viz. Obr. 5.3).



Obr. 5.3 Hala - skladové prostory



Obr. 5.4 Hala - bližší pohled

Bližší pohled na výrobní část, kde jsou vidět obráběcí stroje a manipulační technika.

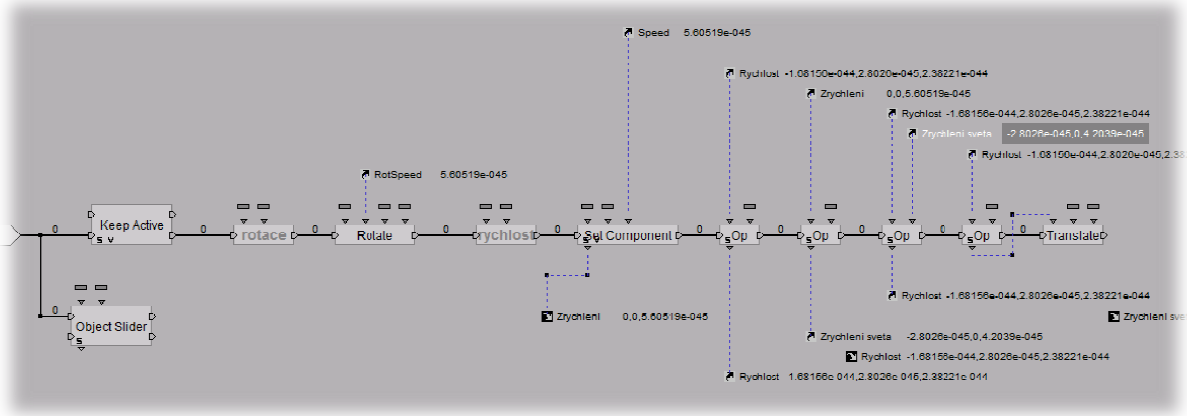
V této kapitole je popsána tvorba skriptů pro pohyb po křivce, ovládání objektu pomocí klávesnice a nastavení kamery. K tvorbě skriptů byly použity stavební bloky, které jsou přesně popsány výše (viz. Obr. 5.4).

5.1 **Výběr objektu, pro který se budou vytvářet skripta**

Skripta se vytváří pro jednotlivé objekty, např. stroje, roviny, křivky, světla, kamery atd… V poli Level manager najdeme objekt, který je chceme vytvořit, označíme jej a v levé části zmáčkneme tlačítko  . Ve správci úloh se nám objeví u tohoto objektu nový odkaz a v části schematik se zobrazí pole na stavbu skriptů pro tento objekt.

5.2 **Ovládání objektu pomocí šipek**

Tento skript byl vytvořen za pomoci pana Josefa Kralovce. Slouží pro pohyb vysokozdvižného vozík po hale za pomocí klávesnice a dále zde byly doplněny i hranice, aby nedocházelo k projíždění vozíku skrz zdi a stoje (viz. Obr. 5.5).



Obr. 5.5 Skript pro ovládání objektu pomocí klávesnice

Set Component- slouží k přiřazení hodnoty Speed do vektoru "zrychlení". Tento blok má 3 vstupy, pro každou osu jeden. Náš parametr je připojen k ose Z. *Op* -blok slouží k operacím s různými typy vstupů. První *Op* blok slouží jako vliv tření prostoru na objekt. Vstupem je vektor rychlosti a číslo udávající tření, kterým vektor násobíme (například 0,9). Druhý *Op*

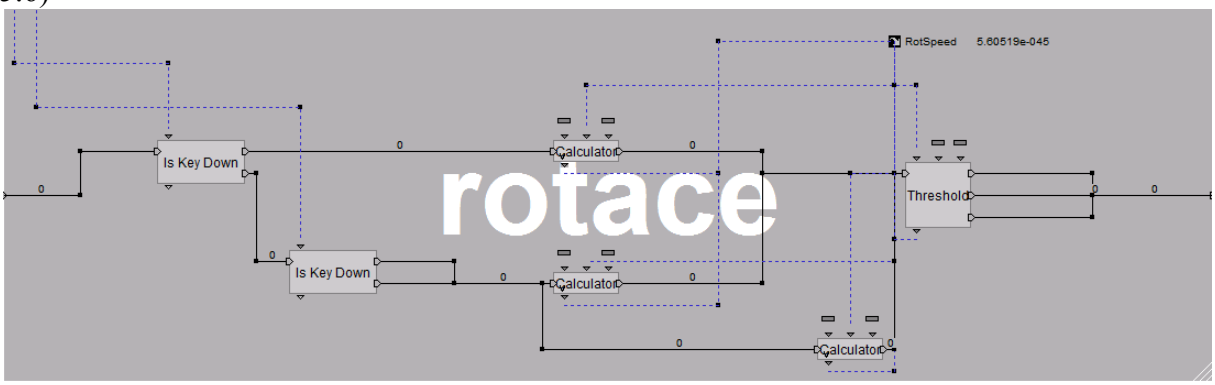
blok slouží k vyjádření zrychlení ve globalních koordinatech (souřadnicích). Vstupní vektor zrychleni je v lokálním prostoru vozíku. Druhým vstupem je 3D Entita vozíku. Provádí se transformace vektoru. Výstupem je vektor zrychlení udávaný v globálních souřadnicích. Třetí *Op* blok slouží k výpočtu rychlosti. Je to jednoduchý princip sčítání vektoru při, kterým se hodnota rychlosti zvyšuje o zrychlení (zrychlení světa) v závislosti na čase. *Object Slider*-blok, který pro náš objekt nastavuje kolize. První parametr je rozsah kolizní zóny okolo nastavených objektů. Druhý parametr je skupina objektů, které budou s objektem kolidovat při pokusu projet skrze ně.

Postup:

- Vybere objekt, pro který chceme vytvořit skript.
- V *Building Blocks* otevřeme *Logic – Streaming*, zde označíme *Keep Active* a přetáhneme jej do části schematik. *In 1* tohoto bloku spojíme se začátkem.
- Dále se *in* bloku *Keep* aktive spojí s podskriptem *Rotace* (viz 5.2.1).
- Následně výstup z podprogramu spojíme s blokem *Rotate* který se nachází v *Building Blocks* po otevření *Logic – Streaming*. Dvojklikem na tento blok se nám otevře tabulka, kde nastavíme nastvíme constantu pro rotační rychlost (RotSpeed).
- Zadání konstanty, ve skriptovacím poli máčkneme pravé tlačítko myši, zde Add *Local Parameter* a objeví se tabulka pro nastavení konstanty. V první řádce nastavíme název konstanty, v druhé necháme typ *Floot* (číslo), v třetím řádku nastavíme hodnotu konstanty. Proto, aby se zobrazil název a velikost konstanty, označíme konstantu, zmáčkneme pravé tlačítko myši a vybereme *Change Parametr Display* a zde vybereme *Name and Value*. Nyní je konstanta v požadovaném a můžeme ji propojovat s ostatními stavebními bloky.
- Propjíme *Out Rotate* s podskriptem *Rychlost* (viz 5.2.2).
- V *Building Blocks* otevřeme *Logic – Calculator*, zde označíme *Set Component* a přetáhneme jej do části schematik. Dvojklikem na tento blok se nám objeví tabulky, do které vyplníme rychlost (speed). *In* tohoto bloku spojíme s podskriptem *Rychlost*.
- Out bloku Set Component spojíme s blokem Op, který najdeme v *Building Blocks* po otevření *Logic – Calculator*. Ty rovnou vložíme 4 za sebe. Ke každému *Op* bloku přiřadíme námi zvolené konstanty. Jednotlivé bloky spojíme z *In* do *Out* následujícího bloku.
- Nakonec skriptu vložíme blok *Translate* (*Building Blocks-3D Transformation- Basic*). *In* tohoto bloku spojíme s *Out* bloku *Op* a *Res* bloku *Op* propojíme s *Target* bloku *Translate*.
- Nakonec ještě k začátku připojíme blok *Object Slider*, který naleznem v *Building Blocks-Colision- 3D Entity.* V kterém po dvojkliku nastavíme objekty skrz které se nelze pohybovat.

5.2.1 Podskript rotace

Slouží pro přiřazení tlačítek pro pohyb do leva a doprava. Nastavuje se zde i rychlost pohybu pomocí Kalkulátorů, který pracují se vzorci a námi zvolenými konstantami (viz. Obr. 5.6)



Obr. 5.6 Skript rotace

Rotate- tento blok slouží k otáčení okolo určeného vektoru. Pro nás je tento vektor (0,1,0), aby k otáčení docházelo jen okolo osy Y. Druhým parametrem je hodnota rychlosti otáčení *RotSpeed*.

Is Key Down- slouží k jednoduchému ovládání programu pomocí klávesnice. Parametrem je klávesa, výstupem zda je či není stlačená. Pro Rotaci je to klávesa doleva a doprava, pro zrychlení je to klávesa dopředu a dozadu. *Calculator*- je blok do, kterého si zapíšeme pomocí matematických výrazů rovnici. V našem případě je v podskriptu použita rovnice A+B a A*B. Pro výpočet jsme si přidali parametr *RotSpeed*. Ten používáme (u rovnice A+B) jako vstup pro písmeno A a pro písmeno B je nastavena pevná hodnota přímo v bloku. Výstupy jsou dva X, které posíláme zpět do *RotSpeed* a tím zajišťujeme, že dokud držíme tlačítko, se zvyšuje rychlost. Druhý výstup je *Out*, který posílá okamžitou hodnotu dále do programu. Druhý typ výpočtu, který se aktivuje v případě, že ani jedna klávesa není stlačena je A*B. Vstup A je opět *RotSpeed* a hodnota B je menší než 1. Tím zajišťujeme plynulé snižování hodnoty. *Threshold* - blok, který nastavuje hranice pro vstupní parametr. Vstupní parametr X je *RotSpeed*. Vnitřní nastavitelné parametry jsou *MIN* a *MAX*. Spodní výstup je připojeny k *RotSpeed*, aby tuto hodnotu omezoval. Výstupy jsou všechny propojeny, aby byl program za tímto blokem aktivní za jakéhokoliv stavu *RotSpeed*.

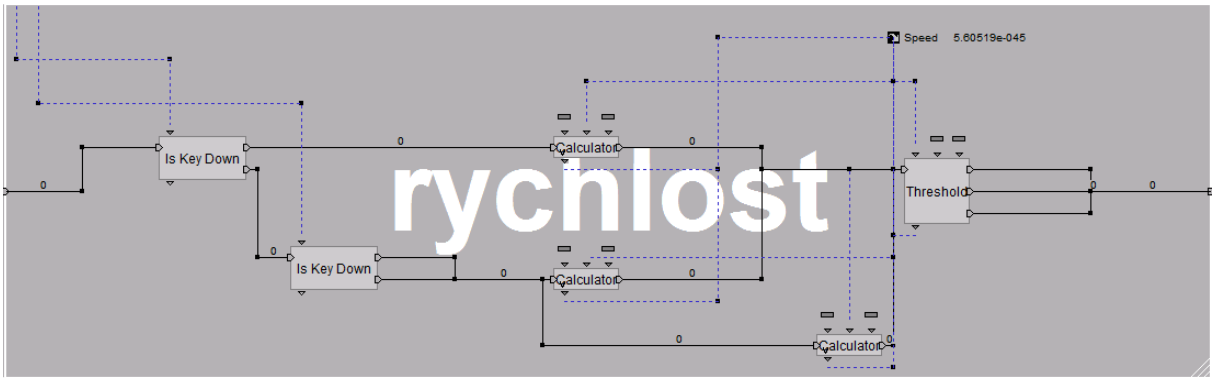
Postup:

- V *Building Blocks* otevřeme *Controllers – Keyboard*, zde označíme *Is Key Down* a přetáhneme jej do části schematik, toto uděláme rovnou dvakrát. *In 1* tohoto bloku spojíme s *Out* bloku *Keep Active*. Dvojklikem na blok se objeví tabulka v, které nastavíme klávesu pro daný pohyb.
- Poté vložíme 3 bloky *Calculator* (*Building Blocks-Logic-Calculator*) a blok *Thereshold* (*Building Blocks-Logic-Calculator*).
- *False* bloku *Is Key Down (1)* spojíme s *In* dalšího bloku *Is Key Down (2)*. True obou bloků *Is Key Down* spojíme s bloky *Calculator (1),(2)* a *Out* bloků *Calculator* spojíme s *In* bloku *Threshold.*
- *False* bloku *Is Key Down (2)* spojíme s *In* bloku *Calculator (3)*. Do *Calculator (3)* nastavíme vzorec a*b, konstantu b=0.9 .
- Do horního *Calculátoru (1)* nastavíme vzorec a+b, konstantu b=0.01 a u dolního *Calculatoru (2)* nastavíme vzorec a-b, konstantu b=0.01. Konstanta a je také předem definovaná (*RotSpeed*)

- Zadání konstanty, ve skriptovacím poli máčkneme pravé tlačítko myši, zde Add *Local Parameter* a objeví se tabulka pro nastavení konstanty. V první řádce nastavíme název konstanty, v druhé necháme typ *Floot* (číslo), v třetím řádku nastavíme hodnotu konstanty. Proto, aby se zobrazil název a velikost konstanty, označíme konstantu, zmáčkneme pravé tlačítko myši a vybereme *Change Parametr Display* a zde vybereme *Name and Value*. Nyní je konstanta v požadovaném a můžeme ji propojovat s ostatními stavebními bloky.
- Konstantu *RodSpeed* spojíme i s vstupy *X (float)* na bloku *Threshold.*
- Všechny 3 výstupy z *Threshold* spojíme v jeden a propojíme s *In* bloku *Set Component*.

5.2.2 Podskript rychlosti

Slouží pro přiřazení tlačítek pro pohyb dopředu a dozadu. Nastavení rychlosti pohybu je obdobné jako u rotace. Podskript "rychlost"- má stejné bloky jako rotace, ale parametr, který se používá při výpočtech je označen jako speed (viz. Obr. 5.7).



Obr. 5.7 Skript rychlosti

Podprogram "rychlost"- má stejné bloky jako rotace, ale parametr, který se používá při výpočtech je označení jako speed.

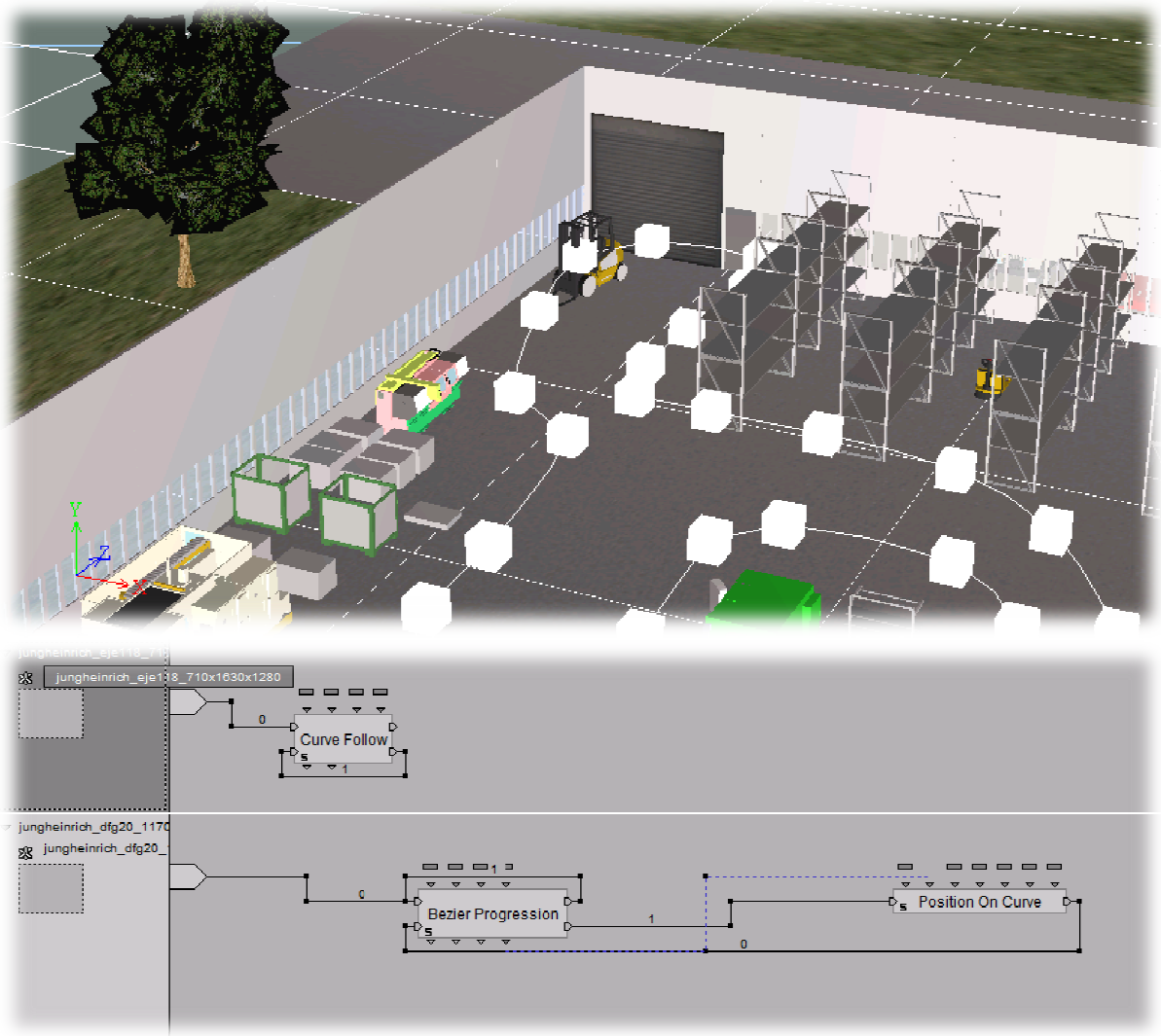
Postup:

- V *Building Blocks* otevřeme *Controllers – Keyboard*, zde označíme *Is Key Down* a přetáhneme jej do části schematik, toto uděláme rovnou dvakrát. *In 1* tohoto bloku spojíme s *Out* bloku *Keep Active*. Dvojklikem na blok se objeví tabulka v, které nastavíme klávesu pro daný pohyb.
- Poté vložíme 3 bloky *Calculator* (*Building Blocks-Logic-Calculator*) a blok *Thereshold* (*Building Blocks-Logic-Calculator*).
- *False* bloku *Is Key Down (1)* spojíme s *In* dalšího bloku *Is Key Down (2)*. True obou bloků *Is Key Down* spojíme s bloky *Calculator (1),(2)* a *Out* bloků *Calculator* spojíme s *In* bloku *Threshold.*
- *False* bloku *Is Key Down (2)* spojíme s *In* bloku *Calculator (3)*. Do *Calculator (3)* nastavíme vzorec a*b, konstantu b=0.9.
- Do horního *Calculátoru (1)* nastavíme vzorec a+b, konstantu b=0.01 a u dolního *Calculatoru (2)* nastavíme vzorec a-b, konstantu b=0.01. Konstanta a je také předem definovaná (*Speed*)

- Konstantu *Speed* spojíme i s vstupy *X (float)* na bloku *Threshold*. Otevřeme threshold dvojklikem a nastavíme zde *Maximální* a *Minimální* hodnotu.
- Všechny 3 výstupy z *Threshold* spojíme v jeden a propojíme s *In* bloku *Set Component*.

5.3 **Pohyb objektu po křivce**

Po vytvoření křivky je možné sestavit stavební bloky v části Schematik pro pohybování objektu po dané křivce. Toto se dá vyřešit více způsoby, volba způsobu stavby záleží na tom, jak chceme, aby se objekt po křivce pohyboval. Varianty jsou ukázány na pohybu vysokozdvižného vozíku po křivce. (viz. Obr. 5.8)



Obr. 5.8 Stript pohybu po křivce

5.3.1 **Pohyb po křivce bez natáčení**

Objekt se bude pohybovat po křivce, ale nebude se natáčet ve směru pohybu. K tomuto slouží blok **Curve Follow** (následování křivky).

- Zvolí se objekt, pro který chceme vytvářet skripta.

- V části schematik se nám vytvoří pole pro tvorbu skript.
- V *Building Blocks* otevřeme *3D Transformations – Curve*, zde označíme *Curve Follow* a přetáhneme jej do skriptovací části.
- Spojíme tažením počátek s *In* bloku *Curve Follow*.
- Propojíme *Loop In* s *Loop Out* bloku *Curve Follow*, čímž se vytvoří smyčka.

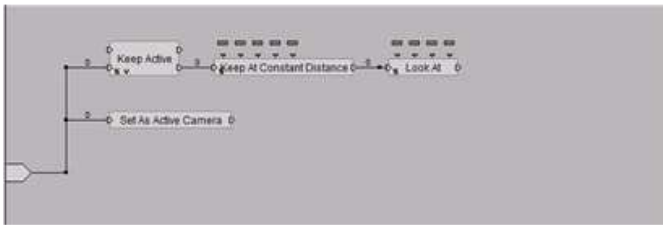
5.3.2 Pohyb po křivce s natáčením

Objekt se bude pohybovat po křivce a bude se natáčet ve směru pohybu. Zde se použijí bloky **Bezier Progression** a **Position On Curve**.

- Vybere objekt, pro který chceme vytvořit skript.
- V *Building Blocks* otevřeme *Logic – Loops*, zde označíme *Bezier Progression* a přetáhneme jej do části schematik.
- Poté opět v *Building Blocks* otevřeme *3D Transformations* – Curve, zde označíme *Position On Curve* a přetáhneme jej do části schematik.
- Spojíme začátek s *In* bloku *Bezier Progresion* a propojíme *In* s *Out*, čímž nám vznikne smička.
- Poté spojíme *Loop Out* bloku *Bezier Progresion* s *In* bloku *Position On Curve*.
- Nakonec propojíme *Loop* In bloku *Bezier Progresion* s *Out* bloku *Position On Curve.* A spojíme *Loop* In bloku *Bezier Progresion* s *Curve* bloku *Position On Curve.*

5.4 Nastavení kamery

Tímto skriptem nastavíme, aby se kamera pohybovala v určité vzdálenosti za námi vybraným objektem.**Keep Active**- popsáno výše.**Keep at constant distance**- tento blok zajišťuje konstantní vzdálenost od objektu. V našem případě je referenční objekt vozík. A vzdálenost je zadána pomocí vektoru a vzdálenosti od vozíku. **Look At**- blok, který má za úkol sledovat referenční objekt. V našem případě sleduje vozík. Rychlost sledování se dá nastavit v % rychlosti pohybu objektu. **Set As Active Camera**- je blok, který při zapnutí programu nastaví tuto kameru jako aktivní (viz. Obr. 5.9).



Obr. 5.9 Skript nastavení kamery

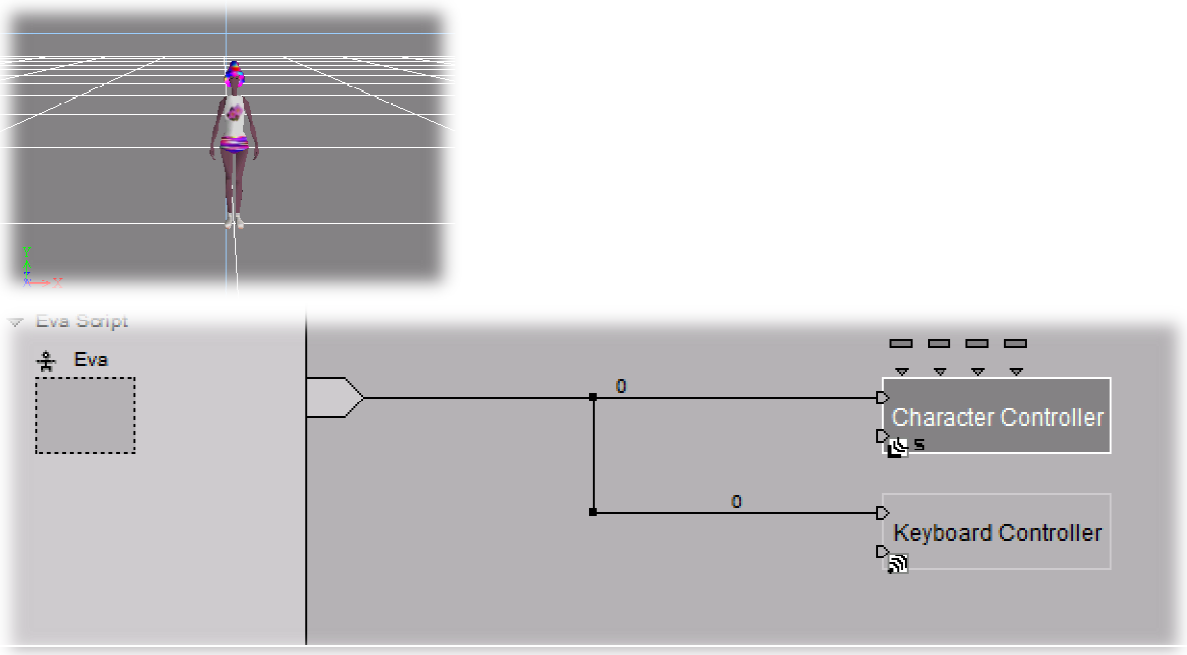
Postup:

- Vybereme kameru, pro kterou chceme skript vytvářet.
- V *Building Blocks* otevřeme *Logic – Streaming*, zde označíme *Keep Active* a přetáhneme jej do části schematik.
- Dále pokračujeme stejně, tedy *3D Transformations – Constaint*, zde označíme *Keep At Constant Distance* a přetáhneme jej do části schematik. Stejně tak i *Look At.*
- V *Building Blocks* otevřeme *Cameras – Montage*, zde označíme *Set As Active Camera* a přetáhneme jej do části schematik.

- Spojíme začátek s *In 1* bloku *Keep Active* a *Out 1* spojíme s In bloku *Keep At Constant*. *Out* tohoto bloku spojíme s *In* bloku *Look At*.
- Nakonec ještě přidáme propojení mezi začátkem a *In* bloku *Set As Active Camera.*

5.5 **Základní ovládání charakteru objektů**

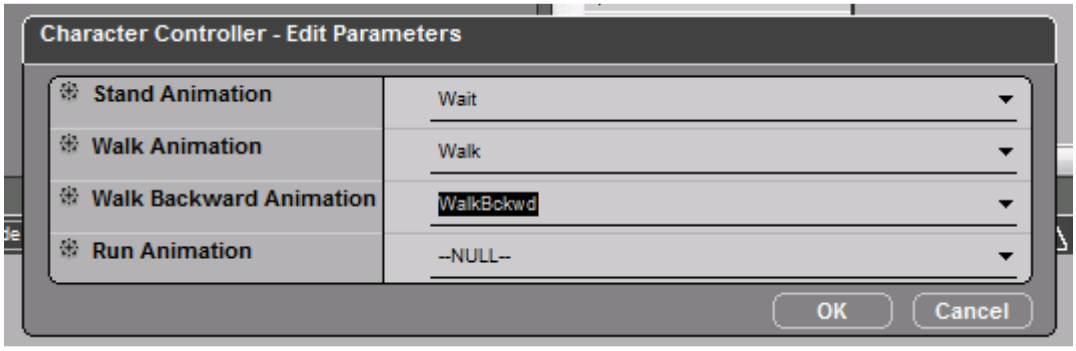
Tento skript slouží pro ukázku základního ovládání vlastností u animace objektů. Použijeme postavu ženy z knihovny programu, které byly přidány vlastnosti chůze a čekat (viz. Obr. 5.10). [16]



Obr. 5.10 Základní ovládání charakteru objektu

Postup:

- V knihovně *VirtoolsResources* najdeme kategorii *Characters*, *Animations* a v ní *Skin Characters*.
- Zde si zvolíme objekt, pro který budeme dávat vlatnosti (zvoleno Eva) a přetáhneme do grafické části.
- Objektu přidáme vlastnosti z knihovny *Skin Characters Animations*. Zde zvolíme soubory wait, walk a walkbckwd. Označíme je a přetáhneme do vybraného objektu umístěného v grafické části, při této akci se objekt označí žlutě.
- O tom, že jsou vlastnosti doopravdy vloženy, se můžeme přesvědčit v části *Level Manager.* Po otevření *Level – Global – Characters – Eva –Animations,* zde vidíme vlastnosti, které jsme přidali.
- V *Level Manager* stiskneme *Set IC For Selected.*
- Nyní vytvoříme skript, otevřeme knihovnu *Building Blocks – Charecters – Movement* a vybereme *Charakter Controller*, který přetáhneme ho na Evu. Tím se nám vytvořil skript v záložce *Schematik.*
- V tabulce *charakter Controller – Edit Parameters* nastavíme dle obrázku. (viz. Obr. 5.11)



Obr. 5.11 Tabulka nastavení parametrů

- Dále přidáme stavební blok *Controllers- Keyboard – Kayboard Controller.*
- Nyní můžeme skript spustit a ovládat Evu pomocí klávesnice.

6 Závěr

Virtuální realita je medium, které reprezentuje tří rozměrné prostředí, které dává pozorovateli vněm skutečnosti. Virtuální realita se může zobrazovat pomocí stereoskopie a například pomocí trackovacího zařízení do ní můžeme interaktivně zasahovat. Jedním z programů pro tvorbu virtuální reality je program Virtools5. Program Virtools5 je vývojový program pro tvorbu, navrhování a simulace ve 3D s podporou nástrojů VR. Využitím zdroje objektů byla vytvořena knihovna objektů v programu Virtools 5. Tyto objekty byly postupně rozmístěny a uspořádány tak aby z nich vznikla výrobní hala. Rozmisťování objektů bylo prováděno pomocí souřadnicového systému. Problém zde byl s označováním objektů.

Objekt se skládá z více částí, které bylo složité označit všechny nejednou, aby bylo možné s ním pohybovat. Po správném uchopení objektu už bylo umisťování jednoduché. Nejdříve byly vytvořeny zdi, aby byl prostor ohraničen, poté byly vkládány stroje a zařízení haly. Hala je rozdělena na výrobní a skladovou část a vše bylo popsáno. Dále byl popsán program Virtools 5, popis jeho částí a tvorby skriptů. Tvorba skriptů slouží pro přiřazování jednotlivých vlastností zvoleným objektům. Skripty se tvoří skládáním jednotlivých stavebních bloků, každý stavební blok má své vlastnosti, vstupy a výstupy. Vybrané stavební bloky byly přeloženy a popsány. Propojováním jednotlivých stavebních bloků se upravují potřebné vlastnosti a chování objektu. V práci je postupně krok za krokem popsána stavba těchto skriptů na 3 příkladech.

7 Seznam literatury

[1] http://www.cis.cz/portfolio-sluzeb/3d-virtualni-realita/ (11.12.2010)
[2] http://www.gali-3d.com/cz/techno-co-je-3d-stereo/techno-co-je-3d-stereo.php (15.11.2010)
[3] http://www.gali-3d.com/cz/techno-anaglyph/techno-anaglyph.php (12.12.2010)
[4] http://www.gali-3d.com/cz/techno-passive-stereo/techno-passive-stereo.php (12.12.2010)
[5] http://www.gali-3d.com/cz/techno-active-stereo/techno-active-stereo.php (15.11.2010)
[6] http://www.intersense.com/uploadedFiles/Products/White_Papers/Comparison%20of%20InterSense%20IS-900%20System%20and%20Optical%20Systems.pdf (15.11.2010)
[7] http://www.3ds.com/fileadmin/PRODUCTS/3DVIA/3DVIAVirtools/Resources/datasheets/ds_virtools5_3DVIA_eng_LR.pdf (12.12.2010)
[8] http://www.3ds.com/products/3dvia/3dvia-virtools/showcase/web/ (12.12.2010)
[9] http://www.3ds.com/products/3dvia/3dvia-virtools/showcase/electronic-entertainment/ (12.12.2010)
[10] http://www.3ds.com/products/3dvia/3dvia-virtools/showcase/virtual-reality/ (12.12.2010)
[11] http://marrtinkyblog.blog.cz/0704/francie 12.12.2010 (12.12.2010)
[12] http://www.freemages.fr/browse/photo-922-geode.html (15.12.2010)
[13] Ing. Petr Hořejší Ph.D., prezentace PIS-01, 2010
[14] http://games.greggman.com/games/Virtools-Review/figure1.gif 31.3.2011
[15] http://www.3dvia.com/search/?search[current_page]=2&search[results_per_page]=12&search[sort_order]=Rank&search[file_types]=1 8.4.2011
[16] http://www.youtube.com/watch?v=MBYWvuBRLKw 3.4.2012