Virtual Planetarium in CyberStage

Valery Burkin, Martin Göbel, Frank Hasenbrink,
Stanislav Klimenko, Igor Nikitin, Henrik Tramberend

GMD – German National Research Center for Information Technology

Abstract. We describe an educational application in virtual environment, intended for teaching and demonstration of basics of astronomy. The application includes 3D models of 30 objects in the Solar System, 3200 nearby stars, a large database, containing textual descriptions of all objects in a scene, interactive map of constellations and tools for search and navigation. The methods, needed for visualization of different scale astronomical objects in virtual environment, are described.

Modern educational process actively uses the methods of computer graphics and scientific visualization. Wide opportunities are opened by emerging technology of virtual environments, which can be used for a creation of high interactive virtual laboratories intended for teaching different disciplines. In this paper we describe an experimental course on basics of astronomy, which is delivered inside the immersive virtual environment system *CyberStage*, installed at GMD, and gives a possibility to explore interactively the Solar System and surrounding stars.

The first section presents the virtual environment system *CyberStage*. The second section outlines *Avango*, the main software component driving this system. The methods used for modeling of astronomical objects are described in the third section and summarized in conclusion.

1 CyberStage

The *CyberStage* [1] is CAVE-like [2] audio-visual projection system. It has room sizes (3m×3m×2.4m) and integrates a 4-side stereo image projection and 8-channel spatial sound projection, both controlled by the position of the user's head, followed by a tracking system (Polhemus Fastrak sensors). The sound projection is completed by vibration emitters built into the floor of the system and allowing for rendering low frequency signals perceivable through feet and legs and by the eight channel-surround-sound system which is fed by IRCAM's room acoustic software Spatilisateur [3]. Shutter glasses (Crystal Eyes) are used for stereo image perception. An SGI Onyx 2 with 4 Infinite Reality 2 graphics subsystems and 12 MIPS R10000 processors generate eight user controlled images. Each pipe generates 11 million shaded triangles per second (peak rate). The display resolution is 1024 x 768 pixels at 120 Hz for each of the four displays.

To create the illusion of presence in virtual spaces, the CyberStage system provides various interfaces and interaction metaphors to visually and acoustically respond to the user's actions. These interfaces allow for navigation in virtual spaces and manipulation of virtual objects. The software driving the CyberStage is the Avango [4] application development toolkit.

2 Avango

Avango is a programming framework for building distributed, interactive VE applications. It uses the C++ programming language to define two categories of object classes. *Nodes* provide an object-oriented *scene graph* API which allows the representation and rendering of complex geometry. *Sensors* provide Avango with it's interface to the real world and they are used to import external device data into an application.

All Avango objects are *fieldcontainers*, representing object state information as a collection of *fields*. They support a generic *streaming* interface, which allows objects and their state information to be written to a stream, and the subsequent reconstruction of the object from that stream. This interface is one of the basic building blocks used for the implementation of object distribution.

Avango uses connections between fields to build a *dataflow graph* which is conceptually orthogonal to the scene graph, and is used to specify additional relationships between nodes, which cannot be expressed in terms of the standard scene graph. This facilitates implementation of interactive behavior and the import of real world data into the scene graph.

In addition to the C++ API, Avango features a complete language binding to the interpreted language Scheme [5]. Scheme is a general purpose programming language descended from Algol and Lisp. It is a high level language, supporting operations on structured data such as strings, lists and vectors. All high level Avango objects can be created and manipulated from Scheme.

The Avango itself is based on SGI Performer to achieve the maximum possible performance for an application and addresses the special needs involved in application development for virtual environments. Along with high performance rendering this framework allows live video sources as well as prefabricated animations to be imported into virtual worlds. Advanced rendering tasks like culling, level-of-detail switching and communication with the graphics hardware are all handled by Performer. Whenever the underlying hardware allows, Performer utilizes multiple processors and multiple graphics pipelines.

3 Solar System: algorithms of visualization

Our main goal is to represent all objects in the Solar System preserving correct angular sizes from any viewpoint, as they would be visible from a traveling spacecraft. There are several reasons why this cannot be done in a simple way, for example, representing the Solar System as Open Inventor's static geometry file. Sizes and distances for objects in the Solar system are too different: maximal ratio of scales is about $r(\text{Sun-Pluto})/r(\text{Deimos}) \sim 10^9$. Usually (in Inventor, Performer and Avango) the geometric shapes in a virtual scene and position of observer in it are described by single precision (4 bytes) real numbers. This precision is insufficient to represent the objects with so different scales, as a result, trembling and other undesirable effects appear.

These problems were removed using the methods described below.

Non-linear geometrical model of the Solar System was constructed in the following three steps:

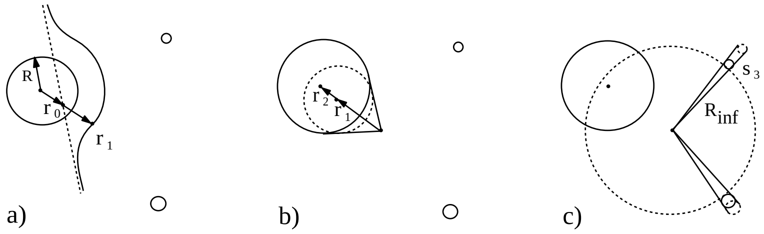


Fig.1. Non-linear geometrical model.

Initially (fig.1a) all planets have the physical sizes and located on their actual orbits. The observer travels along straight lines and can penetrate inside the planets. Then, using a convenient non-linear transformation, which maps the interior of planets to a thin outer layer (fig.2):

$$r_1(r_0) = \begin{cases} R\left(1 + (\delta - 1)\left(r_0/(R\delta)\right)^{\delta/(\delta-1)}\right), & r_0 < R\delta \\ r_0, \ r_0 \geq R\delta; & \delta = 1.5 \end{cases}$$

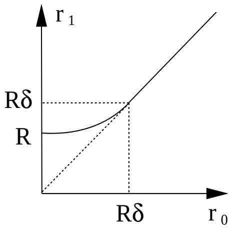


Fig.2. $r_1(r_0)$ dependency.

we push out the observer from the interior. Instead of the collision, the trajectory of observer smoothly rounds the planet and then returns to the initial straight-line course. (This method simultaneously implements no-collision algorithm and route planner.)

On the second stage (fig.1b), for all planets which are sufficiently close to the observer, we apply a uniform scaling with a center at the position of observer and coefficient, given by a function (fig.3):

$$r_2 = r_1/s_1, \ R' = R/s_1, \ s_1 = s_1((r_1 - R)/d),$$
$$s_1(x) = \begin{cases} x(2-x), \ 0 < x < 1 \\ 1, \ x \geq 1 \end{cases}$$

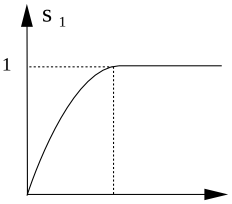


Fig.3. $s_1(x)$ dependency.

After this transformation the minimal distance from the observer to the surface of the planet is given by a formula $\lim_{r_1 \to R}(r_2 - R') = \lim_{r_1 \to R}(r_1 - R)/s_1 = d/2$. We took $d = 3 \cdot 10^4$ km (approximately 5 Earth's radii).

The described effect results in the following. After the distance d, the further approaching of the planet's surface is moderated, and simultaneously, its angular size starts to increase without a bound. This displays the effect of approaching a large celestial body.

Then all linear sizes in the system are multiplied by a global scaling factor $s_2 = 5 \cdot 10^{-8}: \ \tilde{r}_2 = r_2 s_2, \ \tilde{R}' = R' s_2$ (so that the distance of closest approach to the planet, in real units equal to $d/2 = 1.5 \cdot 10^4$ km, in *CyberStage* corresponds to 75 cm).

Finally (fig.1c), all planets, placed outside of a large sphere around the observer's position, are put to this sphere using a scaling:

$$r_3 = \tilde{r}_2 s_3, \ R'' = \tilde{R}' s_3, \ s_3 = \begin{cases} 1, \ \tilde{r}_2 - \tilde{R}' < R_{inf} \\ R_{inf}/(\tilde{r}_2 - \tilde{R}'), \ \tilde{r}_2 - \tilde{R}' \geq R_{inf} \end{cases} \qquad R_{inf} = 50 \text{ m}.$$

Note: the described transformations are applied to each planet whenever the position of observer is changed. These transformation preserve actual angular sizes of planets for any viewpoint. All calculations are performed in double precision (8-byte real numbers), the resulting positions and sizes are presented in single precision and then passed to the rendering pipeline.

Model for motion of observer. The velocity of observer should also vary in a large diapason, to make the exploration of near-Earth space and distant planets possible in one demo session. For this purpose we determine the velocity of observer by its position relative to a closest planet:

$$v(r) = Const \cdot r(1 + \log r/R),$$

r is the distance to planet's center, R is planet's radius. This expression is estimated for all planets and minimum is selected. This law of motion corresponds to the time of travel $t \sim \log(1 + \log r/R)$. Due to the double logarithmic dependence, times of flight between any pair of planets inside the Solar System have the same order of magnitude.

Images of planets – snapshots, made by NASA spacecrafts, were taken from the world-wide accessible archive [6] (courtesy Calvin J. Hamilton), and imposed onto the planetary spheres as textures, see fig.4. For the most of the planets only one-side snapshot is available, and it was imposed to the both sides in reflectionally symmetrical way. Then each planet was oriented "by face" to the Sun, and was enlightened by bright solar[1] and small ambient lights. As a result,

---

[1] Each planet was enlightened by own source of parallel light, directed from the Sun in *initial* model, displayed on fig.1a. Further transformations influence positions of planets but not the direction of enlightening.

only one of two texture copies on the surface of the planet is brightly enlightened, another one just smoothly continues the day side to the night side and then becomes almost invisible.

Far-away planets are rendered using a modified *level-of-details* mechanism, so that a planet is represented as the textured sphere only if its angular size exceeds a threshold $\alpha > \alpha_0 = 0.1°$. Otherwise it is represented as a point[2] with a size in pixels, proportional to the angular size of the planet α, a definite colour – the average one for each planet, and intensity, dependent on the direction of approach to the planet (fig.5) as $I = (1 + \cos\theta)/2$.

Remark: the ring (fig.6) and the shadow of Saturn on it were displayed in the following way. We prepared a thin transverse slice from the correspondent snapshot, and appended a shadowed copy to it. Then we imposed this image to the ring, using the following mapping onto the texture plane: $(x, y, z) \to (u, v) = ((r - r_{min})/(r_{max} - r_{min}), \epsilon)$, where ϵ was equal to 0.75 in the lightened area of the ring, and 0.25 in the shadowed area.

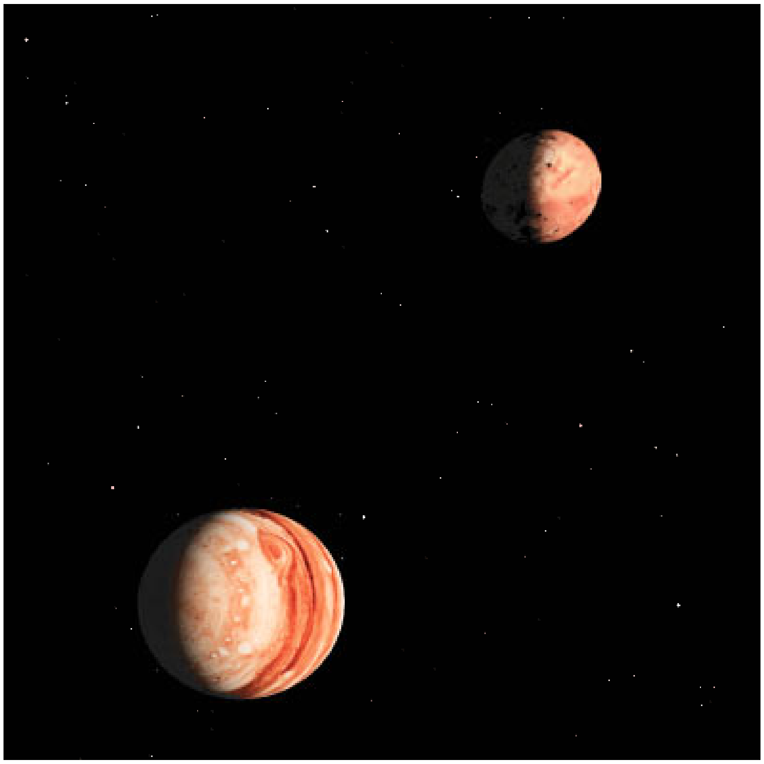


Fig.4. Jupiter and Io.

---

[2] Performer's `pfLightPoint` node.

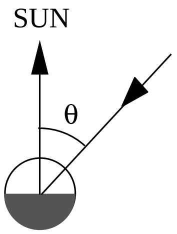


Fig.5. Angle of approach.

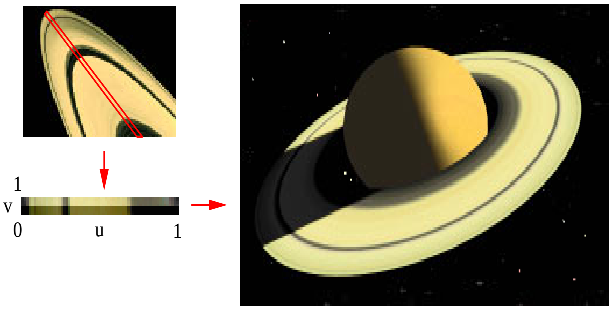
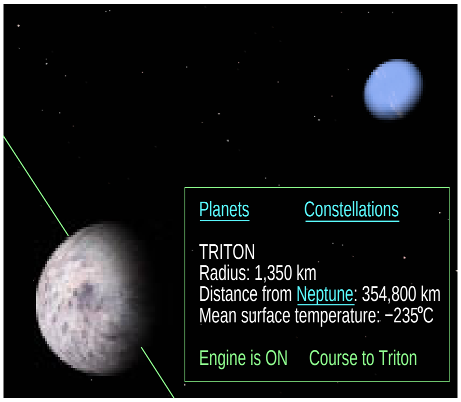


Fig.6. Preparation of a texture for Saturn's ring.

Fig.7. Orion and Taurus constellations.



TRITON
Radius: 1,350 km
Distance from Neptune: 354,800 km
Mean surface temperature: −235°C

Fig.8. Navigation panel.

Stellar sphere. Stellar data are taken from Bright Star Catalogue in Astronomical Data Center NASA [7]. Stars up to the visual magnitude 5.6 (i.e. visible by naked eye in best observation conditions) were selected, resulting to 3200 items in the list. Colours of stars were assigned by a "Blue-Violet" spectral parameter from the Catalogue and a comparative table of spectral classes (courtesy to Dr. Vladimir Pariev, Stewards Observatory, USA). Visual intensities of stars $I \sim 10^{-0.4m}$ for the range of magnitudes $m = -1.46...5.6$ differ in more than 600 times. The light points with so different intensities cannot be displayed in the *CyberStage*: if the brightest star (Sirius, $m = -1.46$) is displayed by a light spot of maximum intensity, containing 10 pixels × 10 pixels, the stars with the magnitude $m > 5$ will be invisible. By this reason we used more weak dependence: $I \sim 10^{-0.1m}$. For this dependence the brightest stars, defining the pictures of constellations, are still clearly distinguishable on the background of others.

Map of constellations. Additionally, the following artificial elements can be displayed on the real sky: a plane containing the planetary orbits (ecliptic) and a map of constellations in two forms: as simple scheme of star connections by straight lines and as high resolution image of ancient celestial map [8]. Images of constellations (fig.7) can be switched on/off separately or all together.

Navigation is performed using 3D pointing device (stylus). Position and orientation of the stylus are permanently registered by electromagnetic detectors, and its direction is marked in *CyberStage* by a long green ray. A click to the stylus' button initiates or stops the motion, with 1 sec period of acceleration. If the button is pressed and held, the rotation of stylus changes the orientation of the observer, also with a small damping.

Additionally, 0.5 Mb textual database was created, containing the lists and detailed descriptions for 30 objects in the Solar System, 88 constellations and each of 3200 stars. This information is displayed in a special panel (fig.8), which emulates HTML-browser and makes easy the exploration of the system. Selection of an item from the lists of planets/constellations initiates the motion to the planet or switches on/off the constellation. Pointing by stylus to a sky object leads to its selection and appearance of correspondent textual description on the panel.

Remarks. Required precision of pointing to the objects was set to 0.5°. To speed up the selection procedure, the stars were preliminary sorted by 100 squares on the sky, and only the stars from the square of current stylus location were considered as candidates for selection. To avoid an occasional selection in arbitrary motion of stylus' ray in the sky, the coincidence is accepted, only if the stylus was held in a fixed direction during > 0.3 sec.

Sound accompaniment was composed Martin Gerke [9]. Two musical themes were selected, representing the rest state and the fast motion. These themes were mixed dependently on the velocity and were used to emphasize the state of motion of observer.

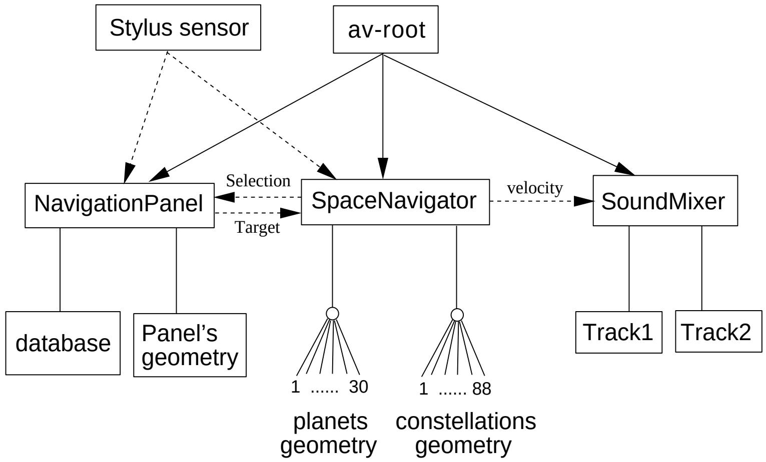


Fig.9. Avango tree.

Principal scheme of running application is shown on fig.9. Solid lines display parent-child relations in the scene graph, dashed lines – field connections. There are three main classes: `SpaceNavigator`, `NavigationPanel` and `SoundMixer`, derived from `fpDCS` basis Avango class. `SpaceNavigator` object defines the position of planets via the described non-linear transformations, switches the constellations, supports interactive selection of objects in the scene, and transmits the index of selected object to `NavigationPanel` via *Selection* field. `NavigationPanel` displays the information from the database, correspondent to the received index, detects the choice of anchor areas on the panel, and informs `SpaceNavigator` about the choice via *Target* field. `SoundMixer` mixes two sound tracks dependently on the value of velocity, received from `SpaceNavigator` via *velocity* field.

Conclusion

An educational application / interactive practicum for teaching of basic astronomy in virtual environment is developed. The application includes:

- 3D models of Solar System and brightest nearby stars created using standard astronomical databases and world-wide accessible planetary snapshots taken from the spacecrafts;
- non-linear geometrical model to represent different scale objects in frames of one virtual scene;
- model for motion of observer, overcoming analogous problem with large difference of time scales;
- implementation of automatic route planner to avoid collisions with planets;
- tools for navigation and search of objects inside and outside the Solar System;
- lightening model, representing real lightening conditions in the Solar System;
- 0.5 Mb database, containing textual descriptions of all objects;
- interactive constellation map with pictures of constellations taken from ancient stellar map;
- sound accompaniment, related with the state of motion of observer.

Using the application, 5 min demonstration video-film is recorded [10].

Further plans:

- implementation of astronomical calendar, defining the position all planets for the given time moment;
- demonstration of solar and lunar eclipses;
- modeling of local relief for simulation of landing on a planet;
- modeling of atmospheres and solar corona;
- inclusion of additional visible objects inside the Solar System (asteroid belt, comets) and outside it (nebulae, Milky Way);
- simulation of interstellar flights, demonstration of deep sky objects and other interesting systems as galaxies, pulsars, quasars and black holes.

Related works. The following projects, using virtual environments for visualization of astronomical objects, are presented in Internet:

- Trimension Technologies Group, creating immersive virtual environment systems for usage in real planetarium: `http://www.trimension-inc.com/index2.html`
- Charles Robertson's Virtual Planetarium, using head mounted displays: `http://www.engr.mun.ca/~charlesr/virplanetarium/VPbody.html`

Web's virtual planetariums:

- `http://www.geocities.com/SiliconValley/Park/8900/csphere.html`
- `http://www.astro.uni-bonn.de/~uzlender/`
- `http://space.jpl.nasa.gov/history.html`
- `http://csm.jmu.edu/orbit/`

Our contribution to this field – the development of methods for realistic representation of astronomical objects in immersive virtual environments – is described in this paper:

- `http://viswiz.gmd.de/~nikitin/stars/starsCB.html`

Acknowledgments. Authors are indebted to Astronomical Data Center / NASA, Jet Propulsion Laboratory / NASA and Calvin J. Hamilton for the given possibility to use their astronomical data and images. We acknowledge the library of University Kaiserslautern, provided us with the celestial map [8]. We thank Martin Gerke for his music, Dr. Vladimir Pariev for valuable consultations, Lialia Nikitina for scrupulous work on digitizing the constellation maps and Klaus-Günter Rautenberg for the help in video production.

References

1. Eckel G., Göbel M., Hasenbrink F., Heiden W., Lechner U., Tramberend H., Wesche G., Wind J. Benches and Caves. In: Bullinger H.J., Riedel O. (eds.) Proc. 1st Int. Immersive Projection Technology Workshop. Springer-Verlag, London, 1997.
2. Cruz-Neira C. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. Computer Graphics Proc., Annual Conference Series, 1993, pp.135-142.
3. Dechelle F., DeCecco M., The IRCAM Real-Time Platform and Applications, Proc. of the 1995 International Computer Music Conference, International Computer Music Association, San Francisco, 1995.
4. Dai P., Eckel G., Göbel M., Hasenbrink F., Lalioti V., Lechner U., Strassner J., Tramberend H., Wesche G. Virtual Spaces: VR Projection System Technologies and Applications. Tutorial Notes. Eurographics '97, Budapest, 1997, 75 pages.
5. R. Kent Dybvig. The Scheme programming language: ANSI Scheme. P T R Prentice-Hall, Englewood Cliffs, NJ 07632, USA, second edition, 1996.
6. Calvin J. Hamilton, Views of the Solar System, `http://www.hawastsoc.org/solar/`
7. Astronomical Data Center NASA, Bright Star Catalogue (5th Revised Ed.) `http://adc.gsfc.nasa.gov/adc-cgi/cat.pl?/catalogs/5/5050`
8. Hemisphaerium coeli boreale / Hemisphaerium coeli australe : in quo fixarum loca secundum eclipticae ductum ad an[n]um 1730 completum exhibentur / a Ioh. Gabriele Doppelmaiero math. prof. publ. Acad. imper. leopoldino-carolinae naturae curiosorum et Acad. scient. regiae prussicae socio ; opera Ioh. Baptistae Homanni sac. caes. maj. geogra. - Norimbergae [Nürnberg] : [Homännische Erbe], [erschienen 1742].
9. Martin Gerke, Music for Virtual Spaces, CD, recorded in AudioLab GMD 1998.
10. Igor Nikitin, Virtual Planetarium in CyberStage, video GMD 2000.

Appendix: Scholar Excursion

This article was processed using the LaTeX macro package with LLNCS style