Serious Gaming for Virtual Archaeoastronomy

GEORG ZOTTI, LBI for Archaeological Prospection and Virtual Archaeology, Austria
BERNARD D. FRISCHER, Indiana University, IN, USA
JOHN FILLWALK, Ball State University, IN, USA

Many cultures worldwide have left traces of sacred architecture and monuments which often show correlation to astronomical events like solstitial sunrises. Virtual archaeology can be used to explore such orientation patterns using digital reconstructions and positions of celestial objects computed from modern astronomical models. Most 3D editing systems used to build virtual reconstructions of such monuments however fail to provide astronomically accurate solar illumination models which can recreate the slightly different solar positions of antiquity or even prehistory, and even worse, any usable representation of the night sky. In recent years, two systems created independently by the authors of this study have been utilized for investigations into the orientation of architecture with respect to celestial processes. Both had their advantages and shortcomings compared to each other. One extended a dedicated open-source desktop astronomy program with a 3D rendering engine where such monuments can be investigated in the first-person perspective with an interactive walkthrough. The other system uses a game engine and external online resources which provides only solar or planetary positions, but no star data. This study presents ways of connecting both systems in an attempt to take advantage of the best of both approaches.

Key words:
Archaeoastronomy, Astronomical Heritage, Virtual Environments, Stellarium, Unity.

SDH Reference:
Georg Zotti et al. 2020. Serious Gaming for Virtual Archaeoastronomy, SDH, 4, 1, 51-74.

https://doi.org/10.14434/sdh.v4i1.31041

1. INTRODUCTION

For millennia, the sky has fascinated and inspired humankind. Evidence from cave art indicates that "sky watching", a systematic and in parts ritualized activity, must have been performed as early as Paleolithic times [Rappenglück 1996], and most civilizations since antiquity have built temples which in some form show connections to celestial processes. The best-known example of European

The Ludwig Boltzmann Institute for Archaeological Prospection and Virtual Archaeology (archpro.lbg.ac.at) is based on an international cooperation of the Ludwig Boltzmann Gesellschaft (A), Amt der Niederösterreichischen Landesregierung (A), University of Vienna (A), TU Wien (A), ZAMG–Central Institute for Meteorology and Geodynamics (A), 7reasons (A), ArcTron 3D (D), LWL–Federal state archaeology of Westphalia-Lippe (D), NIKU–Norwegian Institute for Cultural Heritage (N) and Vestfold fylkeskommune–Kulturarv (N).
Authors' addresses: Georg Zotti, Ludwig Boltzmann Institute for Archaeological Prospection and Virtual Archaeology, Hohe Warte 38, 1190 Vienna, Austria; email: Georg.Zotti@univie.ac.at; Bernard D. Frischer, Dept. of Informatics, Indiana University, 700 N Woodlawn Ave, Bloomington, 47408, IN, USA; email: bernard.d.frischer@gmail.com; John Fillwalk, IDIA Lab, Ball State University, 3100 W Godman Avenue, Muncie, 43304, IN, USA; email: jfillwalk@gmail.com

prehistory is, undoubtedly, Stonehenge with a main axis oriented toward the summer sunrise and winter sunset [Ruggles 2015]. Elsewhere, prehistoric tombs show a preference for sunrise orientations, as, for example, is the case with the 7-Stone Antas of Portugal [Hoskin 2001]; and temples in Malta show solar orientation patterns [Hoskin 2001; Ventura and Hoskin 2015]. Egyptian pyramids show near-perfect cardinal orientation probably derived from stellar observation [Magli 2016], whereas temples show axial orientations which relate both to solstitial and equinoctial sunrise directions as well as the direction of the nearby Nile river [Belmonte et al. 2009]. In eastern Asia, the temples of Angkor Wat show a noticeable correlation with the zenithal passage of the Sun [Barnhard and Powell 2003]. Likewise, the civilizations of the New World built temples with calendrical alignments related to the Sun or to the planet Venus [Šprajc 1996, 2015].

Unfortunately, the separation and specialization of (natural) sciences and the humanities causes a loss in interdisciplinary knowledge and understanding. Many experts in cultural topics are unfamiliar with natural processes observable in the sky, and therefore many potential insights into the role of astronomy or of celestial processes in the everyday life of the cultures they are studying may easily elude them. That said, the generally accepted results already obtained have sometimes made their way via the mass media to the general public, which has responded with fascination and enthusiasm.

In the computer age some important tools for planning astronomical observations have been developed. Of these the most important is a sky simulation, or desktop planetarium program. In principle, it can reproduce the view of the sky as seen from any spot in the world at any time in human history. Increasing levels of sophistication of the programs have brought a corresponding increase in realism, and since the early 2000s some of the planetarium programs have made it possible to put a panorama photograph (hereafter referred to as "pano") into the scene to enhance the observer's sense of immersion. Taken at and correctly aligned to an interesting viewpoint within a historical site possibly built with some astronomical event in mind such as a solstice sunrise, the pano can be used for the purpose of testing archaeoastronomical hypotheses by simulating the sky as seen from the relevant position on the site. The simulation can also show slight difference of solstitial sunrise directions as experienced today in comparison with those occurring when the monument was originally constructed. These differences result from changes in the obliquity of earth's axis to its orbital plane around the Sun. Likewise, it can also demonstrate the slow shift of all stars (almost) parallel to the solar annual path (ecliptic) caused by the precession of Earth's axis, which causes the stars to change their rising and setting points as well as their appearance in relation to the seasons.

In general, we may state that geometrical investigations of ancient monuments involving celestial phenomena seen overhead can only be performed with proper simulation of the relevant ancient skies, which can be easily and reliably provided by these programs. However, the programs do have two important limitations. First, until recently, the user has been restricted to a single predefined viewpoint. This means that explorative investigations which require making observations at arbitrarily changing viewpoints along axes, in corners or other architectural features were not possible. Likewise, other possibly important aspects of the astronomical orientation, such as the interplay of light and shadow in a corridor or the seasonally changing motion of a patch of sunlight, and even interaction with scene objects like historical observing instruments, cannot be simulated. The purpose of this paper is to present a way we have developed to overcome these limitations.

2. SERIOUS GAMING FOR ARCHAEOASTRONOMICAL RESEARCH

The term "serious gaming" describes the use of 3D computer game technology for non-recreational purposes like real-world process simulations or emergency and rescue planning [Vaz de Carvalho et al. 2013]. In a context of cultural heritage, recreations of the past inside a game-like virtual environment have gained high popularity, thanks to the appealing view of scenic reconstructions of past architecture surrounded by landscapes which can be interactively explored in a way already familiar to most users from recreational computer games. The computer game industry also has produced several "game engines", i.e., programming frameworks which support basic requirements such as movement (walking, driving, etc.) in a virtual landscape, importation of 3D architectural models from modeling programs, use of real-world terrain from geodata services, and interaction with scene objects in different ways through the use of physics engines (simulating the effects of gravity, momentum, inertia, etc.). In most cases, scientific investigation and outreach applications in archaeology focus on the reconstructed architecture, which is most often displayed under a blue daylight sky or, at best, a sky decorated with clouds. This is generally implemented through a simple skybox, i.e., a translation invariant scene background consisting of just six (or even five, omitting the bottom) faces of a cube surrounding the user's viewpoint. Virtual characters controlled by algorithms, or even avatars of other users, can be seen as they are moving in the scene, and in educational applications users can interact with characters (e.g., a guide or teacher) to learn about the environment as they are exploring it. Game-like interaction with scene objects is often possible when it can help to bring life into the scenery to be explored. The game engine framework invites the addition of further "eye candy" like vegetation, moving clouds, volumetric shadow effects, dramatic weather, or directional sounds from scene objects. Scientifically useful reconstructions should, however, abstain from such catchy elements and adhere to archaeologically sound data, observing a clear distinction between evidence and hypotheses. Such guidelines can be found in the London Charter [Denard 2009] and Seville Principles [Carrillo Gea et al. 2013].

When it comes to simulating the interaction of past architecture with celestial views, these simple skybox models necessarily fail. While it may be possible to set the position of the Sun, also as most important light source, in the available sky models, computing the solar position for a certain date is usually not available, or, if available, is often limited to dates around the present day. When it comes to prehistoric dates, the slight change in obliquity of earth's axis, which influences the exact rising and setting points of the Sun on the horizon, is not found in commonly available sky or sunlight modules. And while computing the solar position is still moderately easy, when it comes to the simulation of lunar or even planetary alignments of architecture, adding these astronomical simulation data poses a considerable challenge for the application developer.

Zotti and Neubauer [2012a, 2012b] have presented Scenery3D, an experimental 3D visualization module for the open-source desktop planetarium program Stellarium [Zotti et al. 2020]. This program provides many features for sky visualization and observation planning geared mostly toward amateur astronomers and astronomical outreach. Over the past several years, the astronomical simulation models have undergone many improvements especially geared towards applications in historical research. The Scenery3D module has been developed to allow a three-dimensional investigation of a human-made monument placed into a limited enclosing clip of a digital terrain model, thereby facilitating a search of possible astronomical orientation patterns encoded in the architecture [Zotti 2019]. In its current state it can represent large triangulated landscapes extending tens of kilometers and several millions of triangles, however, not modeling the effects of earth curvature may introduce errors. Therefore it is recommended to enclose a 3D model like a single monument with potentially relevant foreground landscape extending only a few kilometers by a landscape panorama that can be created taking effects of earth curvature into account and models mountains in tens of km distance, which do not shift perceivably when the observer walks inside the model by a few steps [Zotti et al. 2020]. In very critical cases like notches formed in the landscape by the intersection of slopes in greatly different distances, this may still only lead to an approximate or preliminary simulation result which should be verified by observation in situ.

At the same time Frischer and Fillwalk [2012] used a bespoke module developed for the Unity game engine to show the insolation (incidence of sunshine) in a particular architectural structure in Hadrian's Villa in Tivoli. In their approach, an online lookup to NASA's "Horizon" service retrieved astronomically accurate solar positions.

Zotti [2014] has presented another Unity-based simulation environment for the archaeoastronomical visualization of a Neolithic circular ditch system in Lower Austria that included a combination of a "live" analytical blue-sky model with a simple cloud module and a transparent skybox-like layer of superimposable diagrams that enhanced the sky with lines helping to visualize the archaeo-astronomical topics under investigation. Diurnal tracks for bright stars, the solstitial Sun paths, similar extreme paths of the Moon, and an altitude grid could be superimposed. In addition, a "particle system" was used to enclose the observer with a stellar sphere accurately computed from a catalog of about 9,000 bright stars [Hoffleit and Warren 1991] which could be rotated according to time of day and the seasons, however, given that stars did not play a critical role for the scene, important visual effects like atmospheric extinction or refraction were not applied to the stars. The Sun was modeled as sphere in some (finite) distance which could be moved along the ecliptic path in the sky, relative to the stars. It also defined the position of the directional light source which casts shadows in the scene and showed a game-like lens flare effect. Along the horizon, the sunlight was reddened, and the glare dimmed to simulate atmospheric extinction, so that the actual sunrise and sunset could also be observed in a zoomed (magnified) view. The Moon and the planets had been found not to have been relevant for the simulation of astronomical details for this site and were thus not included in the scene. Since this whole sky module was rendered as the first element in each frame, it appeared to be at infinite distance. The simulation was also presented as an outreach feature on the project website, but, unfortunately, since it was created, Unity web browser plugins have been abandoned in favor of a unified WebGL-based solution.

Frischer et al. [2016] have presented a study in which the so-called "Antinoeion" in the Villa of Roman emperor Hadrian in Tivoli was placed in these two different virtual environments, the Unity-based environment developed earlier [Frischer and Fillwalk 2012], and the meanwhile published Scenery3D rendering module in Stellarium [Zotti 2015]. One motivation in doing this was for purposes of validation: since the altazimuthal data was calculated differently in each solution, it was interesting to investigate whether the results would be—as hoped—the same. They were, and with both approaches giving practically the same solution with respect to solar positions and shadow effects, the temple could clearly be identified as being oriented towards summer solstice sunrise, corroborating the Egyptianizing interpretation of this temple as being dedicated to Antinous, Hadrian's deceased favorite. However, while Stellarium's simulation of the night sky was found to provide potentially new aspects for architectural investigations with respect to stars and constellations, of course the plants visible in the Unity environment provided the more natural looking views. The temple model was displayed without vegetation and texture-free in Stellarium to make it clear that this visualization was created only for investigation of the geometry in connection to astronomical orientation, not for colorful illustrations (Fig.1). Later, Frischer et al. [2017] used both systems again in an investigation of the Solarium Augusti.

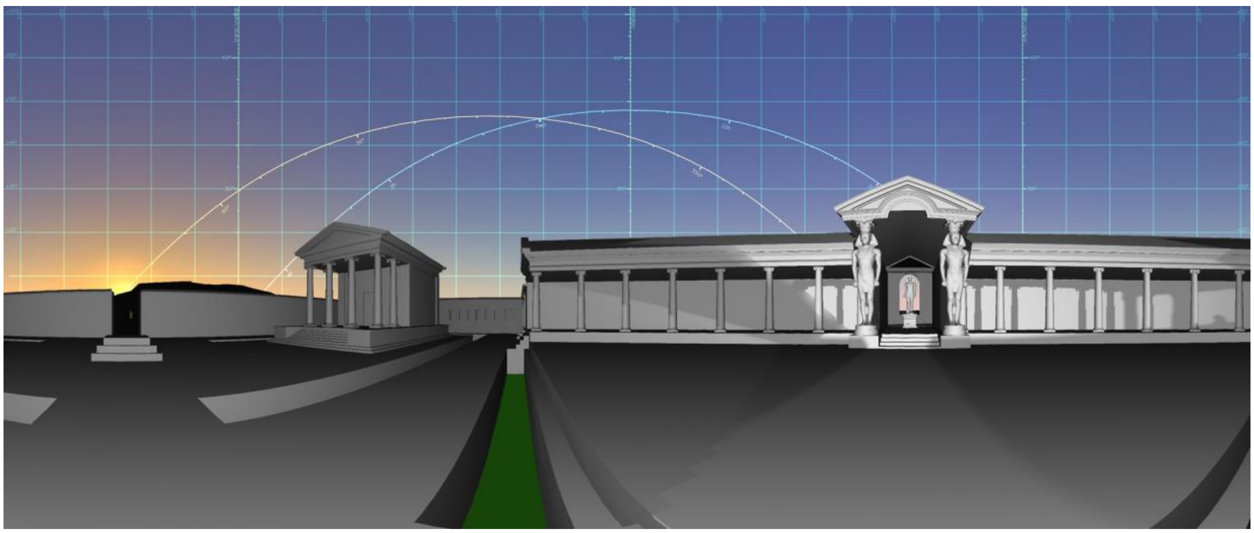


Figure 1. Summer solstice sunrise behind the Colle Ripoli east of Hadrian's Villa in Tivoli is shown to be in line with the main axis of the so-called Antinoeion. The observer is located on the axial path connecting the north-eastern entrance (seen left) to the main statue (seen at central right). Stellarium can produce views in several projections, and a cylindrical projection was used here for a very wide panorama view spanning more than 280° in azimuth. The vertical grid lines are 10° apart and enhanced vertical lines in the angular grid indicate due East, South and West. The blue curve represents the celestial equator, the orange curve is the solar annual path (ecliptic). Screenshot from Stellarium V0.20.0. Model from Frischer et al. [2016] courtesy Matthew R. Brennan and Bernard Frischer (Indiana University).

The Scenery3D module of Stellarium is however not intended to work as complete game engine. Its aim is the investigation of architecture and landscape, i.e., only static objects. In the context of a long-time simulation, parts of the model can meanwhile be made transparent to hide them from view when the time set in the astronomy simulator does not fit the period where the architecture part is assumed to have been visible [Zotti et al. 2018]. It has already been used for accurate visualization of multi-million face models derived from laser scans. It is however not designed to allow motion of or interaction with scene objects other than walking around them.

3. CONNECTING UNITY AND STELLARIUM

In some situations it may be desirable to have a complete rendition of an astronomically accurate and complete sky in combination with the game-like interactivity available when using a complete game engine like Unity. Some simulations aimed at a general public may simply need to look more natural by, e.g., including moving plants or other elements of everyday life easily accommodated in the Unity game engine, while in a historical context we may be interested in simulating the functionality of historical astronomical measuring devices described in ancient manuscripts. A few more developments were required to allow interaction of Stellarium and Unity. One was a way of controlling Stellarium from external programs. For this, the RemoteControl plugin has been developed for use in an exhibition on Stonehenge [Zotti et al. 2017]. Stellarium can now be controlled from a menu presented in a web browser, or by other programs using an HTTP based API. External programs can activate switches to influence the sky displayed in Stellarium, or retrieve data about celestial objects.

The other necessary component for an immediate interaction between the programs has been provided by other developers. Spout, a real-time video sharing framework for Windows, is a free programming library which makes use of NVidia's NV_DX_interop extension to allow using the graphic output of one program that has just been created in the graphics card's framebuffer as texture in another program [Jarvis 2018]. The library is already in widespread use in applications mostly for media and video artists. Developers of Unity-based applications can make use of the free Spout4Unity add-on [Schlupek 2015] which allows the use of Unity both as Spout source (Unity display output to be used elsewhere) and Spout receiver (use of other programs' output as texture in Unity).

Stellarium's Windows version has therefore also been equipped with an optional Spout output since version 0.15.1 and can now be used in several ways to provide an astronomically rich and accurate sky background for applications like Unity that can act as Spout receiver. (Similar functionality exists for Apple Macintosh under the name Syphon, but support for this has not been implemented yet in Stellarium.) In this study, we present three scenarios.

3.1 Simple Skybox Mode

This is the technically simplest way of interaction (Fig. 2): We have added a script to Stellarium which writes the tiles for the faces of a "sky cube," a simple method to produce a sky background that encloses the whole scene. A whole collection of sky scenes can be prepared, which not only contains the six (or five; the bottom face is usually covered by our simulation landscape foreground) textures, but also a simple JSON text file which provides data about solar, lunar and Venus positions and brightness values which can be used by the Unity-based application to configure the scene-relevant illumination and shadows, and optional lens flare effects. Stellarium does not have to run while the Unity-based application is running. While dynamic changes in the sky cannot be simulated in this mode, the whole set of game-like interaction is available under a sky that is astronomically correct.

Such prepared skyboxes can also be used for web-based Unity applications which run in an HTML5 and WebGL enabled web browser, where online retrieval of new skybox data may also happen long after the application itself has been created. It may also be possible to trigger creation of new skyboxes by a remote server running Stellarium and feeding such snapshots as required by the application, for example creating a pseudo "live mode" (with a permissible one or five-minute delay) for the reconstruction of some interesting site, this has however not been tested.

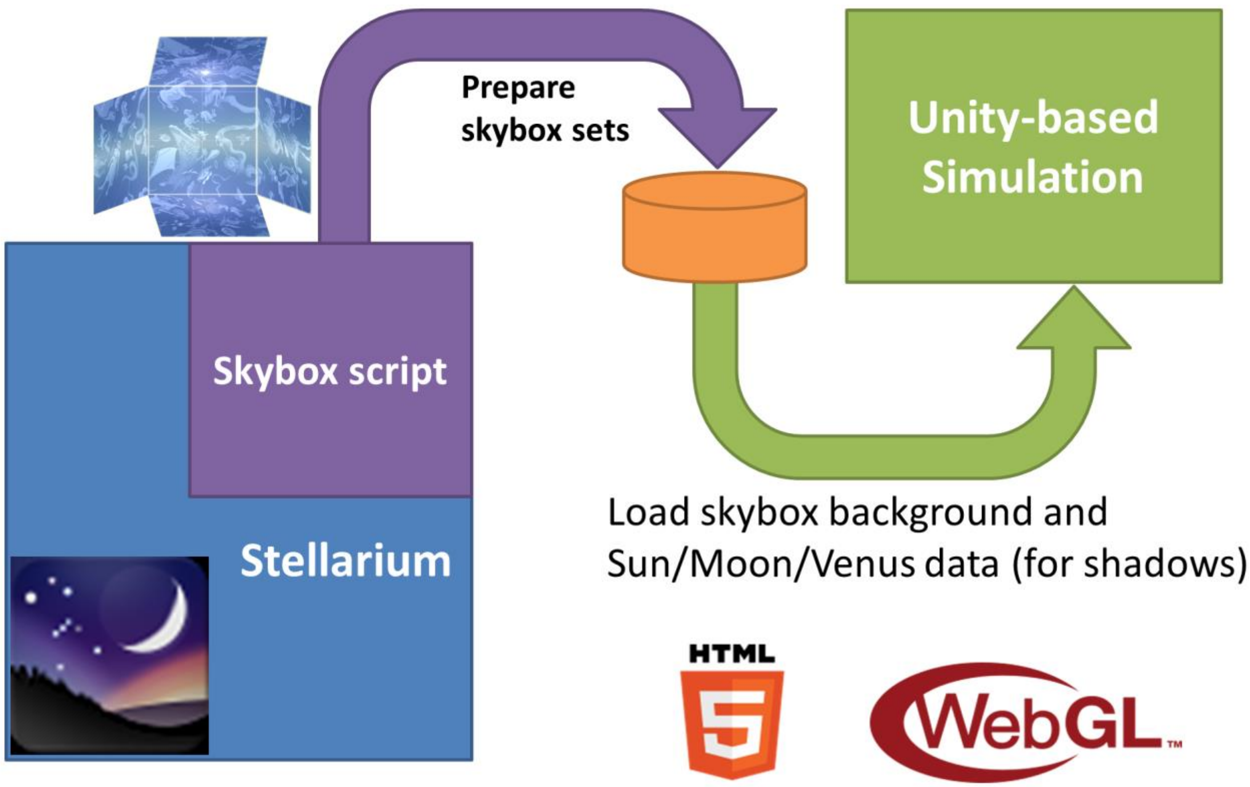


Figure 2. Stellarium is used to prepare skybox textures and data files providing information about the brightest celestial light sources. Unity-based applications, also web versions using WebGL, can use these files to represent the sky background as a skybox and configure the dominant light source: Sun, Moon, or the planet Venus.

3.2 Live Skybox Mode

Stellarium is running in the background, and can deliver astronomical data to the Unity-based application by use of its RemoteControl HTTP plugin API (Fig. 3). Whenever required, the Unity-based application can trigger creation of skybox tiles together with the light data file, and a "file sensor" detects availability of new skybox tiles and updates the skybox and shadow-casting light source from the dominant celestial light source (Sun, Moon, or Venus).

This mode allows setting particular dates and times, or selecting various detail settings in Stellarium's sky like constellation patterns, artwork, or coordinate lines to be displayed in the sky, before triggering the next skybox generation. In all other respects, the Unity-based simulation behaves just like any other simulation that uses a skybox background.

When in the application the user wants to have a close-up view of a celestial object, the resolution of the skybox texture may become a problem. Celestial objects of interest, especially the Sun and Moon, should therefore be added as simple spheres into the Unity scene which should be linked to the observer position to avoid parallax shift. Their positions in the sky will be delivered by Stellarium, and a combination of azimuth and altitude rotations will place them into their correct position in the sky. The "Sun" should then be a self-luminous body, while the Moon should take solar position into account for the simulation of its phase. These will then cover the areas which are not sufficiently detailed in the sky background.

These two modes can also be combined, by providing a few preconfigured skyboxes, plus a "skybox on demand."

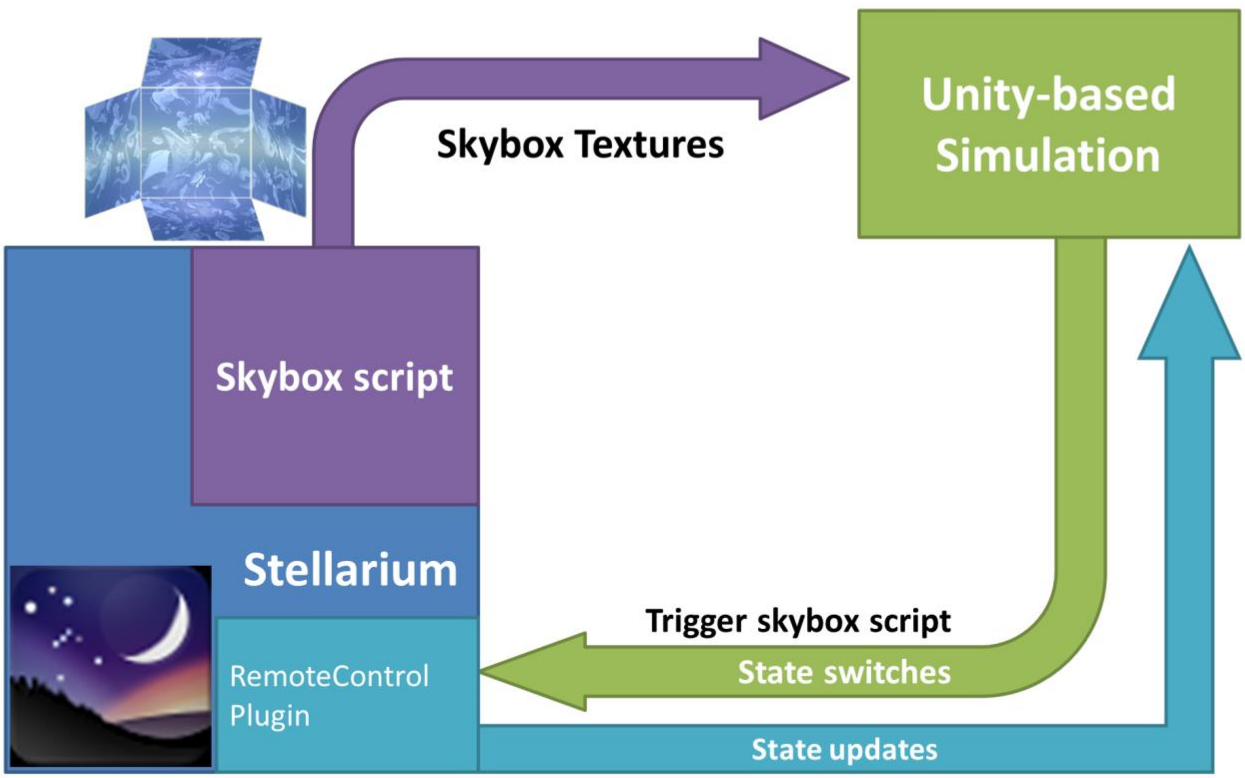


Figure 3. Stellarium is running in the background and interacts with the Unity-based Simulation. An astronomically accurate skybox can be created on demand and will be immediately used.

3.3 Spout Mode

In this scenario, which is limited to the Windows platform, Stellarium is running in the background in "Spout mode" (Fig. 4).

If we want to utilize Stellarium's sky rendition directly in a typical Unity-based application, we have to set perspective projection in Stellarium and then take the usual central perspective camera model and display aspect ratio into account to create a rectangular "canvas" in a position permanently fixed relative to the player's eye or "camera" which is completely filled by the Stellarium image of the same size, and which completely and exactly fills the camera's "view frustum," or field of view. This rectangle is then rendered with a dedicated shader ahead of all terrestrial scene content and without depth information. The sky in Unity is configured to remain blank. Like in the "Live Skybox" mode, other data like date, time, geographical location, or display settings for Stellarium, are exchanged with Stellarium's RemoteControl plugin. The most important switches concerning the sky settings can be controlled directly from within the Unity-based application's menu or hotkeys which forward those commands to Stellarium. When required, the expert user can change over to Stellarium and switch particular settings not foreseen during application development, or apply Stellarium's web interface on a second screen.

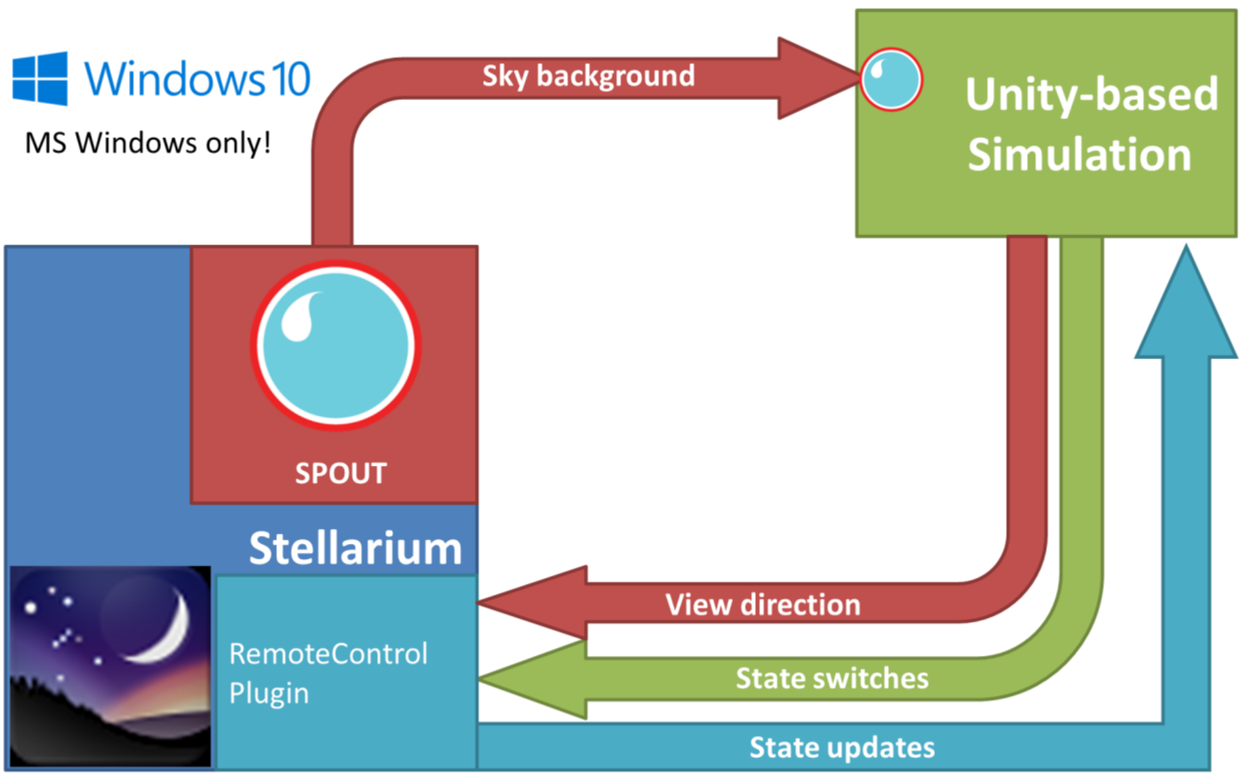


Figure 4. Stellarium is running in the background and interacts closely with the Unity-based application. View direction changes are transferred to Stellarium and almost instantly cause a new background to be visible. Features in Stellarium can be switched with its RemoteControl API from the Unity-based application.

Certainly, the most critical part to keep synchronized is the camera orientation. As the user is moving through the virtual landscape, any camera rotation ("head movement") has to be transmitted to Stellarium, which changes its view direction accordingly. Although Stellarium usually has fast frame updates (depending on CPU/GPU and scene contents, 18-150fps can be observed on an average laptop PC; for regular use, fps can be deliberately reduced to a configurable minimum like 15 or 20fps to conserve energy), there is necessarily a small delay between the immediately visible camera movement with respect to the content of the foreground scene and the sky which updates in the background a few frames later. This may be a problem in applications which target a pure "gaming" audience, and we can especially estimate (without having explicitly tested such scenario) that using this mode in connection with a head-mounted VR display may become uncomfortable. On the other hand, this is clearly the recommended mode of use when it comes to simulation of dynamic changes in the sky, e.g., following the movement of celestial objects or also of shadows on the ground. Zooming the camera field of view will also yield more details in the sky, in contrast to just zooming into the static fixed-size texture maps of the sky box in the other modes. Stellarium delivers again positions for the three luminaries which can cast shadows (Sun, Moon, and Venus) from which the Unity-based application can configure the dominant light source as directional light source and shadow caster, and optionally show a lens flare. As long as the camera does not change orientation, the illusion of an extensive sky module which does everything that is known from Stellarium works perfectly.

4. A SAMPLE APPLICATION

Development of an application providing game-like interaction and an astronomically accurate sky rendition as complete as Stellarium provides is very demanding, especially when it comes to applications that are developed purely for illustrations or for a limited audience. The "Virtual Park of Astronomical Instruments" [Zotti, forthcoming] has been developed as one author's experimental testbed of a Unity based serious gaming application in the domain of historical astronomy. It connects to Stellarium via Spout and the RemoteControl plugin. The idea was the creation of a virtual park in which historical, pre-telescopic astronomical observation instruments can be studied and actually operated under a sky simulation which can give a rendition of the sky for the time when the instruments were in use. This may help to better understand observational records preserved from the historical period. For a deeper "game-natural" experience a few other aspects of creating a nature-like virtual environment were also tried. In this section, we concentrate on the game creation process. We hope it is in the interest of aspiring modelers to add a few practical observations and technical details regarding the handling of models imported from other programs into Unity. We have also picked up several tips from various Unity developers in its excellent developer forum.

Unity provides a means for developing a graphical user interface (GUI) consisting of typical elements like buttons or sliders. Apart from informative text panels and a time display, these were not used in this development, instead we made use of hotkeys (keyboard shortcuts). Where possible, several of the keyboard shortcuts known from Stellarium have been re-implemented and simply trigger a forwarding of the requested action to Stellarium. For example, toggling various coordinate grids, or the progress of time (only in Spout mode), or creation of a new skybox, can thus easily be handled in the same way in this application as it is known to a user of Stellarium. If some setting in Stellarium is not available from the Unity user interface, the user may just switch over to the instance of Stellarium running in the background and activate the required setting there. Alternatively, Stellarium can be controlled using its web browser interface on a second screen or even second computer: it allows connection of several controlling applications which are properly synchronized.

4.1 Terrain

For our virtual park, which is placed in the latitude of Vienna, Austria, we started with a simple flat piece of terrain. We sculpted a few hills near the border just for decoration. Unity also allows import of a real-world digital elevation model (DEM) which must be provided in a 16-bit grayscale raster format. The resulting vertical resolution of 64k values should be enough for most applications. The vertical step width depends on the user's detail settings for the terrain properties. The terrain, if representing some real-world environment, should have its raster parallel to a known Cartesian survey coordinate system (e.g., UTM), and not use geographic coordinates like SRTM data, to avoid distortions. Geographical Information System (GIS) programs can be applied to convert between required formats or also resample to different resolutions. The local offset to geography-ical/astronomical north caused by meridian convergence can be compensated by a rotation of the sky model.

There are limits to the texture size for elevation data described in the Unity documentation. If the scene has to include more terrain, several terrain tiles can be combined, but in an earlier work [Zotti 2014] this raster-based or cell-based approach appeared to not scale well, making it impossible to use high resolution data, or to add lots of elevation detail, over large areas. High resolution may be required to see the effects of solitary rocks or similar features along the landscape horizon. For example, a feature of 1m width would appear 1 arcminute wide (the angular resolution of the human eye) in over 3.4 km distance [Zotti 2020]. It may be more efficient to use a lower-resolution DEM and model the important small details as separate 3D objects. Third-party commercial packages have meanwhile become available from the "Unity Asset Store" which promise to allow larger, triangulated real-world terrain import, which were however not tested and which were not in the focus of this sample application. Another issue for the simulation of large landscapes in Unity (and certainly most other similar environments) is the "camera far clipping plane", a technical requirement which simply cuts off any scene elements farther from the viewer than a certain distance. Setting a too large "far distance" may cause graphical problems in the nearest vicinity.

In any case, Unity-based scenes also suffer from the same principal shortcoming as Stellarium's Scenery3D module: The terrain is modeled in Cartesian coordinates on a "flat earth," so that mountains tens of kilometers in the distance appear slightly too high; a mountain peak 1,000 m higher than the observer should sink behind the curved earth's mathematical horizon about 113 km in the distance. In a flat-earth simulation, the peak still appears about 0.51°, or about a solar diameter, high [Zotti et al. 2020]. A small piece of terrain containing the monument of interest could therefore be enclosed by a static panorama or horizon polygon on a transparent skybox as used by Zotti [2014, Fig.3] which represents far mountains visible in the background as long as a little motion in the scene does not cause a perceivable change of the far horizon.

Invisible collider walls can be placed at the limits of a central area of interest or at the outermost edge of the terrain, preventing the user from leaving or even “falling off” the game terrain.

For the scene user or visitor who explores the park in the first-person perspective, development started with Unity’s standard FPSController.prefab, to which a few additional components were added as described in section 3.3.

4.1.1 Seasonal Vegetation

The ground in Unity applications is usually textured by splat maps which give a mix of actual ground-like textures in sets of four per splat map. A mix of four real textures, like grass, dirt, sand and gravel, seemed sufficient.

It is possible to change the terrain textures on the fly, by programmatically exchanging the textures used to render the splat maps. This can be exploited for seasonally adaptive textures, like using fresh green grass in spring, yellow-brownish summer grass, darker green again in autumn and frosted white winter landscapes (Fig. 5 and Fig. 8). Higher snow cover is of course not possible with this approach, but a mix of frosted-over grass and some snow piles visible only in winter appear useful in communicating the “winter” message.

There are also commercial add-on packages available in the Unity Asset Store which deliver seasonally changing terrain and trees. Plant growth or seasonal differences in tree lines along the horizon can influence the visibility or even apparent directions of rising or setting of celestial objects, and the game engine allows experimenting with such ideas, however, these were not tried in this explorative demonstrator application.

4.1.2 Water surface

To further experiment with a livelier environment, a small pond was placed close to the Sterngarten (Fig. 5). Unity’s default WaterProDaytime prefab works sufficiently well to see curly waves in motion which reflect scene objects and also the sky (either skybox or the “Stellarium Spout canvas”) for a pretty good game-realistic experience, with one lamentable exception: the bright lights (Sun or Moon) do not reflect properly in the water to show a “glitter path.” This will, ideally, follow when real-time raytracing becomes widely available.

In the Water script settings, the clip plane offset was set to 1.6m (approximate adult male eye height) for more accurate reflection of, e.g., the Milky Way. Even better, this clip plane offset can be controlled dynamically to be the difference of camera eye height and water surface height.

4.1.3 Soundscape

The border of the terrain has been decorated with a forest, into which a few audio sources playing birds’ voices were placed. The audio sources were set to “real 3D” with a logarithmic distance falloff to around 50-70 m. All sound clips play permanently but will only be audible when approached by the visitor. Alternatively, a trigger like a SphereCollider could be configured, and the audio source played only when the visitor is close enough.

The first-person controller prefab emits a stomping sound when the user is walking around. Some tweaking or even exchanging of sound files may improve the realism of this sound. In closed buildings like the Observatory (see section 4.3.4) a "Reverb Zone" can be added for interesting echo effects. Try "Stoneroom" or even "Psychotic" settings.

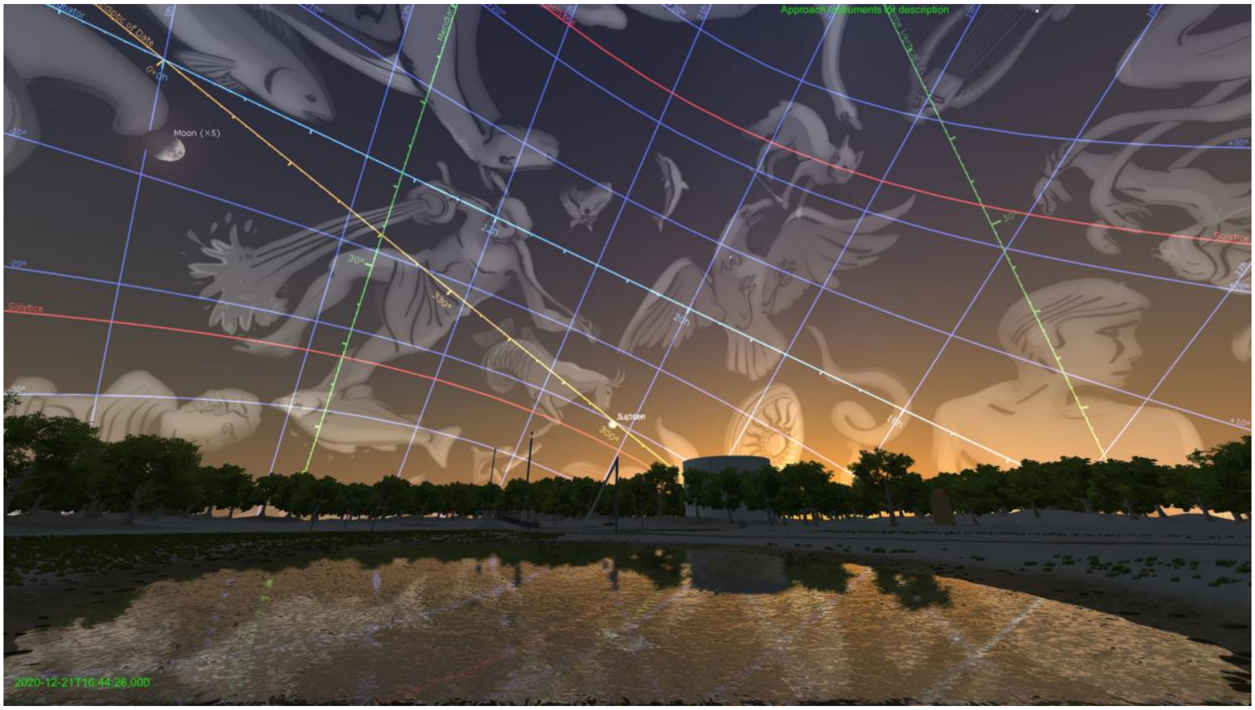


Figure 5. A reflecting water surface can add a very natural element to the simulation. It nicely reflects both the game objects and the sky background provided by Stellarium. This scene shows the conjunction of Jupiter and Saturn after sunset around winter solstice, 2020. The Moon has been enlarged in Stellarium for clarity. A celestial grid and several important coordinate lines have been switched on in Stellarium for explanations of celestial processes. Also note the "frosted-over" appearance of the seasonally changing terrain texture.

4.2 The Stellarium Sky Module

In this demonstrator, all three playing modes described above are available:

- Live Spout Mode
- Live Skybox Mode: The Skybox temporary folder can be the Unity application's folder StreamingAssets/SkyBoxes/live, and must definitely be configurable in Stellarium. For this, the Stellarium scripting core object received a function to read environment variables, and the ability to write to arbitrary directories. Note that the StreamingAssets folder in the final build of the application will be located elsewhere, so after deployment and while running the final program, the environment variable must be reconfigured. Also, the folder must be configured to be user-writable.

- Preconfigured Skybox Mode: A series of prefabricated skyboxes can be loaded as needed, the other two modes are disabled. This is also the version of the application that can be built for WebGL.

After the addition of the NativeSpout module in the Unity Integrated Development Environment (IDE), you should exclude WebGL from settings for Assets/Plugins/*/NativeSpout, and also limit "regular" OS settings to "Windows-only." Spout does not exist on other platforms.

To allow loading skybox tiles which can be made on the fly or pre-configured for WebGL, Unity requires textures to be stored in a special directory, StreamingAssets. A script in Stellarium writes screenshots and the light data file to a directory configured from the environment variable STEL_UNITY_SKYBOXDIR, e.g. D:\InstrumentPark\Assets\StreamingAssets\SkyBoxes\live. From there, the preconfigured skyboxes and data file should be moved to other subdirectories inside the StreamingAssets\SkyBoxes\ directory. Skyboxes can be prepared before compilation or even after. Depending on the application, it may appear useful to prepare skyboxes for each hour on winter solstice, equinox and summer solstice. The vast possibilities of content delivered from Stellarium, including grids, constellation lines, artwork, also the many constellations from other cultures, number of labels, etc. should also be taken into consideration.

A list of skybox names could be offered from a game GUI. For this prototype a skybox selection by hotkey seemed to be enough. The WebGL version loads new skybox tiles over the network only as requested. In the standalone version, other hotkeys trigger "live skybox" mode and switching to Spout mode.

Important hint: The Skybox shader will not be found in the build unless it is referenced in another game object or if you include it manually in the graphics details of the project settings.

4.2.1 Light Optimizations

4.2.1.1 Lanterns and other "evil tweaks"

A pointlight attached to the FirstPersonCharacter can obviously help seeing in the dark or reading the instrument scales in daylight. Also, for better visibility, we can allow ourselves to modify the brightness of the directional light, in case we want to more clearly discern the shadows cast by Venus.

4.2.1.2 A Sun Impostor for Skybox Mode

Looking through the instruments in Skybox Mode suffers from the low-resolution cubemap textures in which the Sun is just a large bright patch. It is not possible to discern the outlines of the solar disk.

To remedy this lack of detail we can add a child node, a little sphere, to the scene, and control its position in relation to the FPScontroller's camera, and its orientation and visible diameter to match the solar position and size in the sky.

4.3 The Observational Instruments

From previous work several models of astronomical interest had been developed in Google/Trimble SketchUp. Unity (version 2017.3 was used for the development described here) has a pretty good importer for models created with SketchUp. It fully preserves scene hierarchy, which means we can animate the instruments which usually have a joint where an altitude bar (alidade) can be lifted to measure the altitude of a celestial object. Some instruments can also be rotated in azimuth (compass direction), i.e., around the vertical axis. To allow this, we must define the component axes and ensure correct hierarchy of static outer frames, joints, and moving parts, in SketchUp. Then the model can be imported to the scene, and C# (or JavaScript) scripts can be developed to control the movement (simple angular rotation) of the joints. Occasionally it was useful to invert a joint axis in SketchUp and re-import the model when the sense of rotation contradicted the astronomical conventions.

Put the SKP files to the Assets/Models folder. Use the Unity IDE to "Extract Materials and Textures" to subfolders of these names, respectively. Materials will have the model names prepended.

It is wise to modify all textures from SketchUp: select all textures and set wrap mode to "Repeat", and create MipMaps to avoid aliasing (moire) effects. Likewise, you can modify the materials to achieve more realistic effects than can be achieved in SketchUp (e.g., Metallic reflection).

All parts of the model which the visitor may interact with, be that floor levels, interactive instrument elements or just barriers that should impede walking, have to be equipped with a MeshCollider.

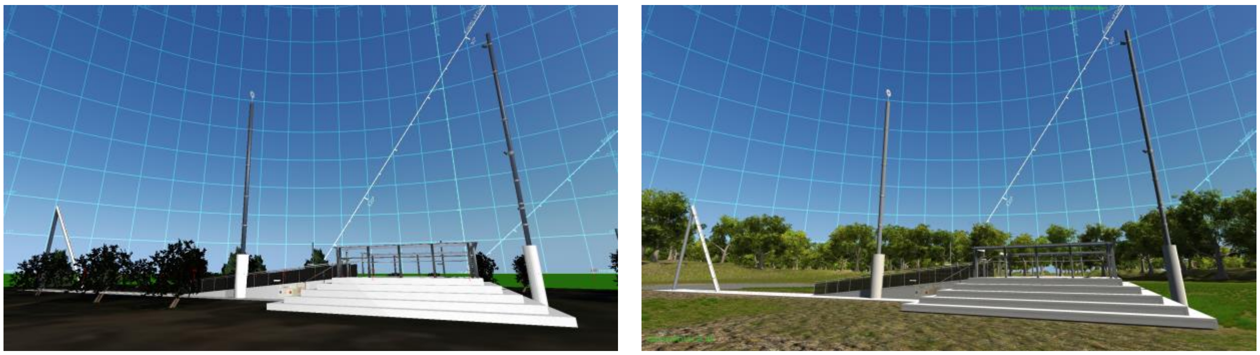
With other modeling programs, the process should be similar. Note that e.g. OBJ models will import as one mesh, therefore adding movable joints is not possible.

To allow interaction only in the vicinity of the instruments, a SphereCollider placed over each interesting part of the scene can be helpful. When the player approaches the instrument and enters the SphereCollider's radius, a help panel (text overlay) is displayed which explains the instrument, and for instruments that allow direct observation, interaction can be activated. A radius of about 6m seems adequate in most cases.

Note that these trigger SphereColliders by default also make the lens flare vanish when the Sun is covered, unless you set all instrument nodes with trigger spheres, but not their child nodes, to the Layer "Ignore raycast".

4.3.1 Vienna "Sterngarten"

The "Sterngarten" (Star Garden) is a public star observing platform built 1997-2001 by the Austrian Astronomical Society on the outskirts of Vienna, Austria, after plans of astronomy popularizer Prof. Hermann Mucke (1935-2019). It consists of a raised platform surrounded by a railing that defines the mathematical horizon for an observer standing on a defined spot in the platform center with an eye height of 1.5m above the platform. The architecture has been devised for explanations of fundamental concepts of astronomical phenomenology [Mucke 2002]. Two high masts in the north and south mark the meridian. They bear altitude marks, and while the northern mast carries a disk that indicates the celestial North Pole, the southern mast carries altitude marks for the Sun when it crosses the local meridian at solstices and equinoxes. Six outlying poles indicate where on the mathematical horizon the Sun rises or sets on the equinoxes and solstices. Short arms indicate the azimuth deviation of the sunrise and sunset points caused by atmospheric refraction. A meridian line extending to the north indicates the calendar date by the end of the shadow of the north mast when it crosses the meridian. A third inclined mast, parallel to Earth's axis, acts as gnomon for a large sundial (Fig.6).



(a) (b)

Figure 6. Vienna "Sterngarten" as presented (a) in Stellarium and (b) in the game-like Virtual Park of Astronomical Instruments. While the astronomical-geometrical features of the architecture can be completely explained and reproduced in Stellarium, the visual appeal of the presentation in the game engine is expectedly higher. Stellarium provides the sky background, azimuthal grid and important lines painted in the sky for explanations.

The "Sterngarten" includes in its architecture an immensely rich collection of astronomical details and was therefore perfectly suited as test environment. The model had been created in 2010 during early refinements of Stellarium's atmospherical refraction (tested with a static landscape panorama exported from the model), development of Stellarium's Scenery3D plugin [Zotti and Neubauer 2012b], and an OBJ export has been included as one of the default 3D sceneries in Stellarium. It had also been used in Google Earth's 3D building layer. Its geometry only works in the geographical latitude of Vienna, requiring the placement of the Virtual Park in this latitude. However, the "Sterngarten" does deliberately not have any moving parts, therefore, apart from the more natural aesthetics and explanatory inserts it does not require the game engine but can be explored, demonstrated and explained completely in Stellarium.

4.3.2 Ghāzān Khān's instruments

Another previous study [Mozaffari and Zotti 2012] had resulted in SketchUp models for most of the 12 instruments presented by Ghāzān Khān in the second phase of Marāgha (around 1300). These consist of mostly straight rules with regular graduations which are connected with simple joints and axes (Fig. 7). For the user interaction, each instrument is controlled by a dedicated controller script. In the respective Update() method we can handle mouse input via a Raycast: depending on the point where the mouse click hits the instrument, its axes can move. A sound effect can be linked to the instrument's motion.

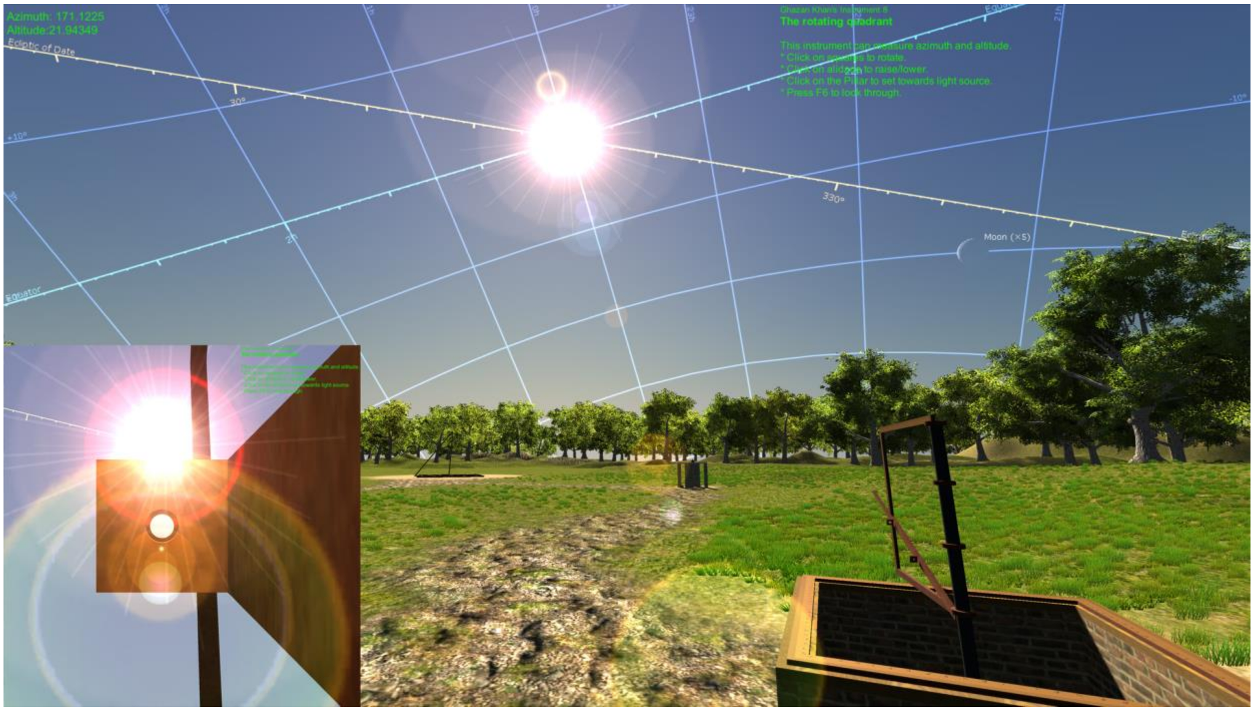


Figure 7. Virtual reconstruction of one of the instruments of Ghāzān Khān used in the early 14th century in Marāgha, Iran, to measure altitude and azimuth. A moving arm (alidade) with sights (peepholes) can be lifted inside a rotating square frame. The azimuth (bearing) of this vertical square can be read on the outer, horizontal square. The instrument can be moved with the mouse or keyboard commands, or the eye linked to the instrument, and so the measurement process (including the discomforting solar glare; see inset) can be replayed by the visitor. An almost identical instrument was built in the late 16th century by Danish astronomer Tycho Brahe. The scene shows the morning of the equinox, where the Sun, moving on its annual path, the ecliptic (orange line), crosses the celestial equator (blue line). Two more instruments are visible in the scene background. Note the distortion of the square frames and apparent tilt of the vertical central pole of the instrument typical for a wide-angle perspective.

4.3.2.1 Attaching the viewer's camera to the instrument

We want to reconstruct the process of "measurement" with the instruments. For this we need a "users' eyepoint" prepared in SketchUp. Typically, this is a point in line with the sights on the alidade or similar device. Orient the green (y) component axis into the designated view direction. Then the respective InstrumentController script should include a key-triggered switch of camera position between the FirstPersonController and that observer position. While the visitor is observing with (looking through) the instrument, the cursor keys can move the instrument as needed.

4.3.3 The Fakhrī sextant

In 994 AD, Abū Maḥmūd Ḥāmid b. al-Khiḍr al-Khujandī (*ca.* 945–1000) built a gigantic instrument near Ray (near modern Tehran), the so-called *Fakhrī* sextant, exclusively for the determination of the ecliptic obliquity. It consisted of two walls aligned along the meridian between which a cylindrical trench was excavated [Repsold 1918]. During solar noon, a spot of light cast by an opening in the axis of the trench can be observed when it crosses the meridian marked along the meridian arc formed by the trench. In the latitude of Ray (35.6°), the Sun stays always above 30° altitude when crossing the meridian, and therefore only a sixth of a circle (a sextant), which was carved into the ground for stability, was needed for all measurements (Fig. 8).

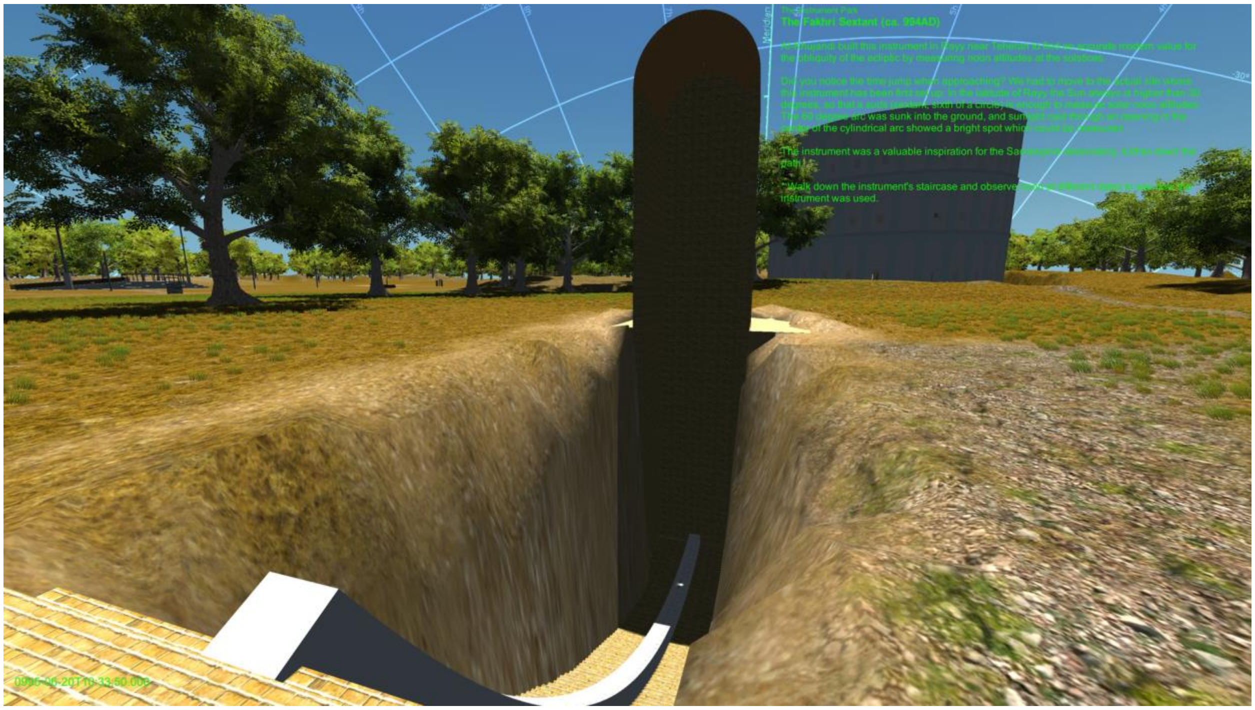


Figure 8. The Fakhrī Sextant. The scene shows Solar noon at summer solstice, when a patch of sunlight falling through a hole at the center point of the arc's curvature (near the top of the back wall) crosses the meridian close to the bottom of the trench (bright spot in the shadow). In winter, the light patch would cross the arch close to its near end (just below the surface). The angular distance represents twice the ecliptic obliquity determined with this instrument. Also note the automatic vegetation change to dry grass indicating the summer season (see Section 4.1.1). The outline of the reconstructed observatory of Samarqand is visible in the background.

Placing this model, created for yet another study [Zotti and Mozaffari 2020], into the virtual park faces an interesting problem. The park has been designed around the Sterngarten at the geographical latitude of Vienna. For a demonstration of the Fakhrī Sextant we have to allow location travel in our exploration. This is pretty simple to achieve, though, by the vicinity trigger script attached to the model's SphereCollider which usually just triggers the display of a text description. In this case, we also move the location to the original place where the Sun is considerably higher and can experience the seasonal variations of solar noon passage with this interesting instrument. Of course, in the skybox modes, we would have to prepare skyboxes particularly for the latitude of this instrument and the instants of solar noon at the solstices.

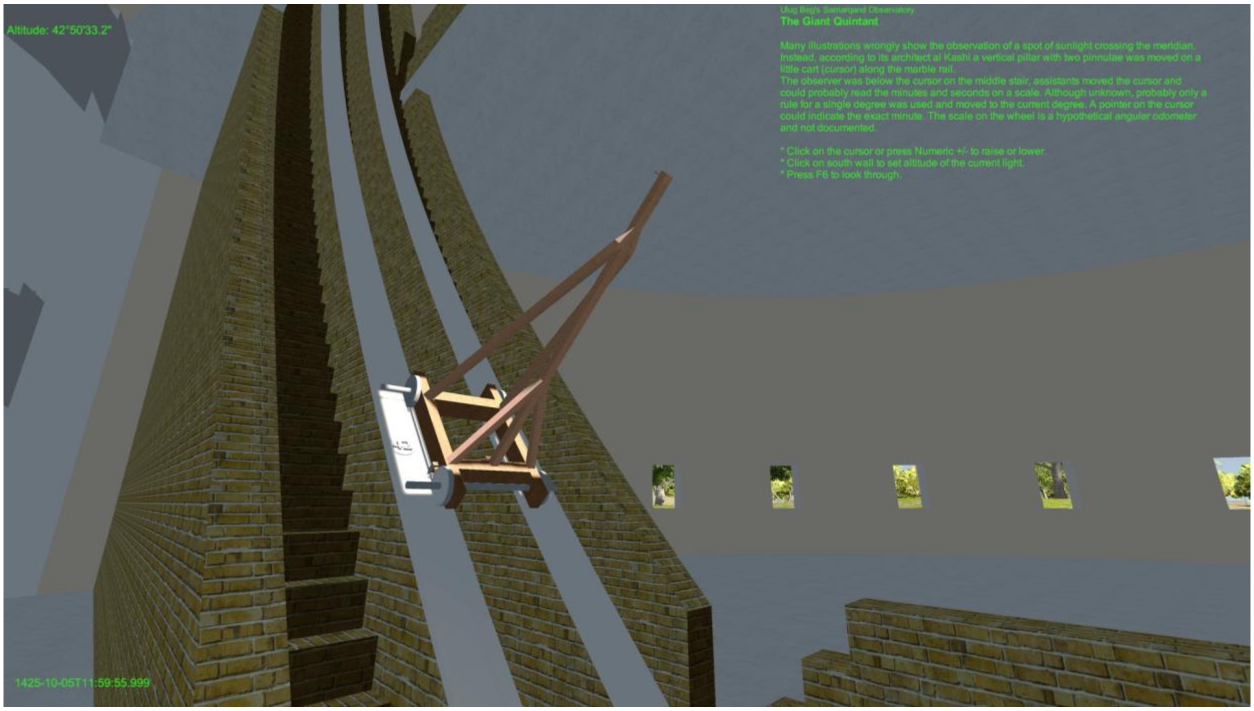


Figure 9. The observing cart in the virtual reconstruction of the central meridian instrument of Ulugh Beg's Samarqand observatory [Zotti and Mozaffari 2020] can be moved by the visitor along the curved rails cut into the marble surface of the 40 m arc. Looking through the sights onto the sky provided by Stellarium allows the recreation of the historical observations from its heydays (around 1430 AD). In this scene, a patch of sunlight crosses the meridian arc at local noon. The cart, for which proof has been found in a manuscript, has been positioned to measure the transit altitude. The observer would have been placed on the central staircase, while the cart would have been operated by two assistants in the side staircases and probably secured by bolts. The model also allowed developing the authors' conviction that this instrument was not a full quadrant, given that the cart would not be usable in the vertical uppermost parts of the instrument elsewhere hypothesized as complete quadrant. The image has been brightened up for reproduction.

4.3.4 Samarqand observatory

The largest instrument of its kind ever created was a meridian quadrant-like instrument that has been built by Ulugh Beg (1394-1449) in Samarqand. His astronomical tables of stellar positions measured with this and other instruments were considered valued data and were, therefore, published in 17th century Europe. However, after Ulugh Beg's assassination, the magnificent observatory was soon destroyed and plundered for bricks until well into the 20th century. During the 20th century, its remains were excavated, and so a partial reconstruction is possible based on very incomplete contemporary descriptions of its lost splendor. The visitor today can visit the subterranean trench and the preserved remains of the meridian arc of about 40 m radius (now vaulted over for protection) which was based on the Fakhrī Sextant but shows important differences. The major part of the arc, which extended high over ground to allow observations of objects also lower in the sky than the Sun, was built inside a large, richly decorated cylindrical three-store observatory building. It has, however, not survived the centuries, and the exact shape and mode of operation can therefore only be speculated upon. Recently, a clear proof in form of a manuscript has been found, showing a movable cart with sights running up and down the rails carved in marble plates which still are partially visible in the ruins. We have presented our results elsewhere [Zotti and Mozaffari 2020] but saw immediately that it can also be presented and demonstrated in the Virtual Park of Astronomical Instruments (Fig. 9).

Given the fragmentary descriptions of the building, it was only reconstructed approximately as a large cylindrical building. Again, a SphereCollider was added around the whole building as trigger sphere for GUI explanation text and activation of the interior machinery. As was the case for the Fakhrī Sextant, in this case it also triggers a location change to its original setting, Samarqand, to be able to recreate observations that may be required to better understand its operation. This helped us to understand that with great probability the instrument was limited to measure meridian altitudes from 80° to 19° or 18° (these altitudes proven by sculpted altitude plaques found in-situ along the marble rails, or in the excavated rubble, respectively), as it would have been merely impossible to keep the cart on the rail in a hypothetical full quadrant's highest, near vertical part.

In this simulation, the user can explore the building, climb a staircase to the roof, or operate the meridian instrument by pushing the cart up and down, and then looking through the instrument, fine-tuning the altitude while the observed object moves through the field of view of the sights in natural speed.

4.4 Observations for a WebGL version

A WebGL version of a Unity-based application which should work in web browsers provides attractive opportunities for outreach but also new challenges for the development. For obvious reasons, this cannot have Spout interaction, but the instruments can be operated, and a few skyboxes should be loadable on demand.

Some precautions have to be taken for building such a variant. Most important, the whole program should be limited in size to reduce download times, and should have limited scene complexity, especially for non-essential elements such as a large number of detailed trees in the terrain. The application of various scene optimizations provided by the Unity IDE, for example occlusion culling and static batching for the non-moving model parts, also appears to be more important in this version than in the desktop application. It is generally recommended to load data only as required to allow a fast start. In our application, only the skybox textures can be easily loaded on demand, which however even allows creation of skybox sets after the application itself has been deployed. Also, texture size can generally be reduced, or more aggressive texture compression can be applied. It seems advisable to limit skybox tiles to 512x512 pixels in the WebGL version for fast data transfer.

The Spout scripting object and its target, the Stellarium screen background, must be disabled. In code, this can be done in their Awake() methods with #if UNITY_WEBGL … #endif preprocessor macros. Other C# functionality not available in the WebGL version, like screenshot creation, can be excluded from building in the same way.

In the build settings, WebGL memory has to be increased to 500MB for reasonably smooth operation.

4.4.1 Website Integration

The default template for WebGL builds is a half-sized window in a full-screen page without surrounding text. The default CSS tags seem to leave much room for improvement. Some modifications in the CSS and index.html allowed to provide a textual introduction with the game window on the right side. The demonstrator application can be visited on https://homepage.univie.ac.at/Georg.Zotti/InstrumentPark/index.html.

5. CONCLUSION

Serious gaming for applications in archaeoastronomy and historical astronomy has become available and is more accessible than ever. The Scenery3D plugin of Stellarium can be used to explore static models and large landscapes in OBJ format and can also make use of Stellarium's many camera projections which allow wide-angle views that appear much more natural than the standard OpenGL perspective camera model can deliver. More lively applications, which allow manifold interaction possibilities and that we like to term "Advanced Virtual Archaeoastronomy," can be created by combining a game engine like Unity with the astronomically solid sky background created by the Stellarium desktop planetarium.

It should also be possible to retrofit an existing Unity-based simulation with this module for astronomical investigation, which may offer significant new insights for interactive investigation and replay of possible astronomical orientation schemes if the scene has been built with accurately georeferenced models. Of course, for that to happen it will be important to create a web-based repository of openly available accurate data and 3D models of heritage sites to be investigated or demonstrated. At the moment, no such repository exists.

The basic ingredients of a Unity package, C# code and a "Stellarium controller prefab" for Unity, are available from the Stellarium project site at github, https://github.com/Stellarium/stellarium-unity.

6. ACKNOWLEDGEMENTS

Parts of the development were implemented by Neil Zehr and David Rodriguez from IDIA Lab.

7. DISCLAIMER

These materials are not sponsored by or affiliated with Unity Technologies or its affiliates. "Unity" is a trademark or registered trademark of Unity Technologies or its affiliates in the U.S. and elsewhere.

Product names in this paper are used for identification and reference only and are the sole property of the respective copyright holders.

8. REFERENCES

Edwin Barnhart and Christopher Powell. 2013. The importance of zenith passage at Angkor, Cambodia. (2013). Retrieved March 20, 2020 from http://mayaexploration.com/pdf/angkorzenithpassage.pdf

Juan Antonio Belmonte, Mosalam Shaltout and Magdi Fekri. 2009. Astronomy, Landscape and Symbolism: A Study of the Orientation of Ancient Egyptian Temples. In: Juan Antonio Belmonte and Mosalam Shaltout, eds. *In Search of Cosmic Order: Selected Essays on Egyptian Archaeoastronomy*. Supreme Council of Antiquities Press, Cairo ISBN: 9789774794836, p. 213.

Juan M. Carrillo Gea, Ambrosio Toval, José L. Fernández Alemán, Joaquín Nicolás y Mariano Flores. 2013. The London Charter and the Seville Principles as sources of requirements for e-archaeology systems development purposes. *Virtual Archaeology Review*, 4,9, (2013), 205–211. https://doi.org/10.4995/var.2013.4275

Denard, H. 2009. The London Charter for the computer-based visualisation of cultural heritage. Retrieved from http://www.londoncharter.org/fileadmin/templates/main/docs/london_charter_2_1_en.pdf

Bernard Frischer and John Fillwalk. 2012. The Digital Hadrian's Villa Project: Using Virtual Worlds to Control Suspected Solar Alignments, in Gabriele Guidi and Alonzo C. Addison, eds. In *VSMM2012 (Virtual Systems in the Information Society) – Proceedings*. IEEE, 49-55.

Bernard Frischer, Georg Zotti, Zaccaria Mari and Giuseppina Capriotti Vittozzi. 2016. Archaeoastronomical experiments supported by virtual simulation environments: Celestial alignments in the Antinoeion at Hadrian's Villa (Tivoli, Italy). *Digital Applications in Archaeology and Cultural Heritage*, 3,3 (2016), 55–79. https://doi.org/10.1016/j.daach.2016.06.001

Bernard Frischer, John Fillwalk, Paolo Albéri-Auber, David S. P. Dearborn, Mika Kajava, and Stefano Floris. 2017. Edmund Buchner's Solarium Augusti: New observations and simpirical studies. In *Rendiconti (Atti della Pontificia Accademia Romana di Archeologia, Serie III)* Vol. LXXXIX. Tipografia Vaticana, 3–90.

Ellen Dorrit Hoffleit and Wayne H. Warren, Jr. 1991. The Yale Bright Star Catalogue, 5th Revised Ed. (Preliminary Version). Retrieved September 6, 2020 from http://tdc-www.harvard.edu/catalogs/bsc5.html

Michael Hoskin. 2001. Tombs, Temples and their Orientations: A New Perspective on Mediterranean Prehistory, Bognor Regis, Ocarina Books.

Lynn Jarvis. 2018. Spout. Realtime video sharing framework for Windows. Retrieved March 26, 2020 from https://spout.zeal.co

Norman Lockyer. 1906. Stonehenge and other British Stone Monuments Astronomically Considered. London, Macmillan and Co.

Magli, Giulio. 2016. "Archaeoastronomy : Introduction to the Science of Stars and Stones." Springer International Publishing. https://doi.org/10.1007/978-3-319-22882-2

S. Mohammad Mozaffari and Georg Zotti. 2012. Ghāzān Khān's Astronomical Innovations at Marāgha Observatory. *Journal of the American Oriental Society*, 132, 3 (July-September 2012), 395–425.

Hermann Mucke. 2002. *Himmelskunde im Freiluftplanetarium Wien*. Österreichischer Astronomischer Verein, Vienna.

Michael Rappenglück. 1996. The Pleiades in the "Salle des Taureaux," Grotte de Lascaux (France). Does a Rock Picture in the Cave of Lascaux Show the Open Star Cluster of the Pleiades at the Magdalénien Era, ca. 15,300 B.C.? In Carlos Jaschek & F. Atrio Barandela, eds. *Actas del IV Congreso de la SEAC – Proceedings of the IVth SEAC Meeting, Astronomy and Culture*. Universidad de Salamanca, 217–225.

Johann A. Repsold. 1908. Zur Geschichte der astronomischen Meßwerkzeuge, *Astronomische Nachrichten* 206, Cols. 125-138.

Clive L. N. Ruggles. 2015. Stonehenge and its Landscape. In C.L.N. Ruggles, ed. *Handbook of Archaeoastronomy and Ethnoastronomy*. Springer Science+Business Media New York, 1223-38. https://doi.org/10.1007/978-1-4614-6141-8_118

Stefan Schlupek. 2015. Spout4Unity. Retrieved April 7, 2020 from https://github.com/sloopidoopi/Spout4Unity

Iván Šprajc. 1996. *Venus, lluvia y maíz: simbolismo y astronomía en la cosmovisión mesoamericana*. Córdoba: Instituto Nacional de Antropología e Historia.

Iván Šprajc. 2015. Astronomical Correlates of Architecture and Landscape in Mesoamerica. In C.L.N. Ruggles, ed. *Handbook of Archaeoastronomy and Ethnoastronomy*. New York: Springer Science+Business Media, 715–728. https://doi.org/10.1007/978-1-4614-6141-8_56

Frank Ventura and Michael Hoskin. 2015. Temples of Malta. In C.L.N. Ruggles, ed. *Handbook of Archaeoastronomy and Ethnoastronomy*. New York: Springer Science+Business, 1421-1430. https://doi.org/10.1007/978-1-4614-6141-8_133

Carlos Vaz de Carvalho, Pedro M. Latorre Andrés and Francisco J. Serón Arbeloa. 2013. Serious Games Network. *Virtual Archaeology Review*. 4,9 (2013), 174–180. https://doi.org/10.4995/var.2013.4271

Georg Zotti and Wolfgang Neubauer. 2012a. A virtual reconstruction approach for archaeoastronomical research. *Proc. 2012 18th Int. Conf. Virtual Syst. Multimedia, VSMM 2012 Virtual Syst. Inf. Soc.* (2012), 33–40. https://doi.org/10.1109/VSMM.2012.6365904

Georg Zotti and Wolfgang Neubauer. 2012b. Virtual Reconstructions in a Desktop Planetarium for Demonstrations in Cultural Astronomy. In M. Ioannides, D. Fritsch, J. Leissner, R. Davies, F. Remondino, and R. Caffo, eds. *Progress in Cultural Heritage Preservation (Proc. 4th International Conference EuroMed 2012, Limassol, Cyprus, Oct. 29–Nov. 3, 2012)*, volume 7616 of Lecture Notes in Computer Science. Springer: Heidelberg, 170–180. https://doi.org/10.1007/978-3-642-34234-9_17

Georg Zotti. 2014. Towards serious gaming for archaeoastronomical simulation. *Mediterranean Archaeology and Archaeometry*, 14,3 (2014), 271–281. Retrieved from http://maajournal.com/Issues/2014/Vol14-3/Full25.pdf

Georg Zotti. 2016. Open-Source Virtual Archaeoastronomy. *Mediterranean Archaeology and Archaeometry*, 16,4 (2016), 17–24. https://doi.org/10.5281/zenodo.207260

Georg Zotti, Florian Schaukowitsch and Michael Wimmer. 2017. The Skyscape Planetarium. In Liz Henty, Bernadette Brady, Darrelyn Gunzburg, Frank Prendergast, and Fabio Silva, eds. *The Marriage of Astronomy and Culture: Theory and Method in the Study of Cultural Astronomy (Papers from the 2016 SEAC Conference)*, volume 21 of *Culture and Cosmos*, Lampeter, Ceredigion, Wales, Spring/Summer and Autumn/Winter 2017, Culture and Cosmos & Sophia Centre Press, 269–281. http://www.cultureandcosmos.org/pdfs/21/CCv21_17Zotti.pdf

Georg Zotti, Florian Schaukowitsch and Michael Wimmer. 2018. Beyond 3D Models: Simulation of Phased Models in Stellarium. *Mediterranean Archaeology and Archaeometry*. 18, 4, (2018), 501–506. Accessed on Oct 2020 from this link.

Georg Zotti. 2019. Visualising skyscapes: GIS-based 3-D modelling and astronomical simulation. In Liz Henty and Daniel Brown, editors, *Visualising Skyscapes: Material Forms of Cultural*

Engagement with the Heavens. London and New York: Routledge Studies in Archaeology, 35–54.
Georg Zotti. 2020. GIS, Landschaft, 3D-Modelle und Himmelssimulation. In Gudrun Wolfschmidt, ed. *Maß und Mythos, Zahl und Zauber: Die Vermessung von Himmel und Erde* (Tagung der Gesellschaft für Archäoastronomie in Dortmund 2018), Nuncius Hamburgensis 48: Beiträge zur Geschichte der Naturwissenschaften, tredition Hamburg, 296-305.
Georg Zotti and Alexander Wolf. 2020. *Stellarium User Guide*, version 0.20.0. Retrieved March 2020 from https://github.com/Stellarium/stellarium/releases/download/v0.20.0/stellarium_user_guide-0.20.0-1.pdf
Georg Zotti and S. Mohammad Mozaffari. 2020. New Light on the Main Instrument of the Samarqand Observatory. *Journal for the History of Astronomy* 51(3), 255–271.
Georg Zotti, Susanne M. Hoffmann, Alexander Wolf, Fabien Chéreau and Guillaume Chéreau. 2020. The Simulated Sky: Stellarium for Cultural Astronomy Research. *Journal for Skyscape Archaeology* 6, 2 (in press).
Georg Zotti. Forthcoming. A Virtual Park of Astronomical Instruments. *Proc. SEAC2018*, Graz.

Received June 2020; revised September 2020; accepted September 2020.