Home Planet Development Log

by John Walker

Release 2 Development

16 September 1994

Integrated fixes to SKY.C from the Sky Screen Saver which handle failures either to allocate the circular clipping region or to select it into the HDC used to paint the star map. This corrects the mysterious "random blank sky map" problem which occurred when the map window was large and memory was tight.

28 September 1994

Fixed labels in the entry for Uranus in PLANETS.CSV. The "km/sec" that's supposed to follow the escape velocity was appended to Earth masses item instead.

Integrated fix from Sky Screen Saver to correctly flip the Sky window South up for southern hemisphere observing sites.

Integrated fix from Sky Screen Saver to gracefully recover from failure to allocate the Sky window device dependent bitmap. An "insufficient memory"

message is now displayed in the window in this case. I implemented the same logic to handle the corresponding failure in the Telescope dialogue procedure.

Corrected the standard colours requested in the Sky window palette. Several colours used for legends in the Sky map were missing, which could cause unfortunate substitutions on some displays.

Implemented the "Align constellation names" option. If enabled, the Sky window generates custom TrueType fonts to rotate constellation names so they're aligned with the local horizon. This looks more like conventional sky maps, and reduces the likelihood of a constellation name being clipped at the horizon. Constellation names are never aligned in the Telescope window.

Installed fix in SCRNSAVE.C to correct "invalid HWND 0000" when no time zone warning dialogue is removed. Since changes made of late have blown away the ability to build the screen saver, testing this will have to be deferred for a while.

If the Map window was iconised, the Sky window was not updated. This was due to bad logic in updimage (SUNCALC.C), which conditioned updates of the Sky window on whether the Map window was iconic. I changed the test to invalidate the Sky window if it was displayed and was not an icon.

3 October 1994

Integrated a totally new planet position calculator based on the VSOP87 theory of planetary motion. This new module, VSOP87.C, replaces the old-fashioned PLANETS.C which used series from Meeus' 1985 "Astronomical Formulæ for Calculators". The VSOP87 module produces results with errors in the arc-second range over a wide span of time. Corrections for aberration due to the finite speed of light, nutation in longitude and obliquity, transformation from dynamical coordinates to the FK5 system, and compensation for the (very small) latitude of the Sun's apparent position are included. The nutation is calculated by the new function nutation() in ASTRO.C, which uses the IAU expressions. The calculation of the position of the Moon, Pluto, and asteroids and comets being tracked remains as before, except that Pluto's position now benefits from all the corrections mentioned above.

The planetary position calculation function recognises a new bit in the "which" argument FULL_PLANET_INFO. If set, all fields in planet_info are filled in and all corrections are calculated. If zero, only the heliocentric longitude, latitude, and radius vector are computed, the series used to evaluate these items are truncated, and no corrections are applied. This enables the Orrery to bypass these refinements when calculating the planetary orbits (the difference is much less than a single pixel at the scale of the orrery, so accuracy is not sacrificed).

5 October 1994

Installed a canned orbit resource which drastically speeds up the Orrery display--in some cases the Orrery is calculated and displayed 100 times faster than before. The overwhelming percentage of computation associated with the orrery is evaluating the planetary orbits by multiple calls on planets(), yet planetary orbits change but little over time. I included a resource which contains precomputed orbits (stored as scaled 16 bit numbers) circa J2000, which can be used to display the orrery for the millennium centred upon that date.

The orbits drawn by the orrery didn't always precisely close. You never noticed this unless an animation was being run, since the gap in the orbit was always smaller than the planet icon and occurred at the planet's current position in its orbit. I added a closing segment to eliminate gaps in the orbits.

Replaced the old high-precision Moon position calculation with the one derived from Chapront's ELP-2000/82 lunar theory as given in Meeus, "Astronomical Algorithms", chapter 45. This theory is accurate to ten arc-seconds in longitude and 4 arc-seconds in latitude in contemporary epochs. Note that the far less accurate calculation in phase() is still used in several places in the program.

6 October 1994

The asteroid and comet selection mechanism in the Object Catalogue contained logic to deselect a currently-tracked object by choosing an entry named "-none-" (actually, just the leading hyphen is tested, allowing localisation). However, the asteroid and comet databases (ASTEROID.CSV, ASTNUM.CSV, and COMETS.CSV) did not include "-none-" entries. I added them.

11 October 1994

I completed a total revision and update to the asteroid databases, using as a starting point the:

CATALOGUE OF ORBITAL ELEMENTS AND PHOTOMETRIC PARAMETERS
OF 4646 MINOR PLANETS NUMBERED BY NOVEMBER 2, 1990

Yu.V.Batrakov, V.A.Shor
Institute for Theoretical Astronomy of the Russian
Academy of Sciences, St. Petersburg

incorporating the errata from Francois Ochsenbein (CDS, Strasbourg). Data for asteroids 4646 through 5632 were taken from files posted on the CompuServe ASTROFORUM, in MPC format. I merged the resulting database with the earlier one, which contained several asteroids named after the ITA catalogue was published.

The evaluator for Kepler's equation used in ASTEROID.C used Sinnott's binary search method for eccentricities below 0.95 and the iterative Gauss method for higher eccentricities. This caused very eccentric (but still elliptical) cometary orbits such as Halley's to use the Gaussian method in the region where it is extremely slow. I changed the threshold to 0.995, which allows most comets to use the still-accurate Sinnott method, but defers to Gauss for parabolic and

hyperbolic orbits. This doesn't make much difference except for the Orrery, which calls the orbital position calculator numerous times to plot the orbit. The orrery display for Halley's comet, for example, dropped from 477 seconds generation time to 13.3 seconds--more than 30 times faster. I also added a "quick" argument to trackAsteroid which suppresses calculation of right ascension, declination, and radius vector and, in turn eliminates evaluation of the obliquity of the ecliptic. When planets() in VSOP87.C is called without the FULL_PLANET_INFO bit set, this mode is selected. This contributes to the speed-up of the orrery calculation.

The clever code which limits the extent of a parabolic or hyperbolic orbit was running afoul of other clever code which estimates the number of segments needed to plot its orbit near perihelion. Result: only he outbound leg of the orbit was plotted, complete with interplanetary U-turn to incorrectly close the orbit. Fixed.

12 October 1994

When the Orrery window was obscured or otherwise damaged while the refresh bitmap remained valid, repainting the image sometimes didn't restore the frame around the Orrery image. This was due to the update region not allowing a full copy of the backing bitplane. I added an InvalidateRgn call to guarantee the entire bitmap is restored.

When the Orrery was being painted, no clip region was used to restrict the image to the display box. If an asteroid or comet appeared outside the box (but still within the astronomical unit limit, due to perspective), it could be drawing outside the image panel. I added a clipping region to prevent this.

Similarly, I added a clipping region to the dragging code for the vertical and horizontal scrollbars. This prevents paint splattering outside the image panel when the Earth icon impinges on its edges.

I ripped out and totally reimplemented the code which plots orbits of comets and asteroids. Before, it shared logic with the planetary orbit code, but the differences were just too great to maintain the fiction that they were logically parallel. All the planetary orbits are close enough to circular that they can be plotted in equiangular segments without obvious errors. Further, they are always closed and have a known period. Asteroid and comet orbits obey none of these simplifying assumptions. I added a general orbit plotter which handles orbits of any eccentricity and size. It calculates the time of perihelion and draws the two arms of the orbit from that point, forward and back, until either they join somewhere on the orrery, or are clipped at the edge of the image. I modified the clipping logic to account for the square boundary of the image panel. More importantly, the code now calculates the instantaneous orbital velocity of the body and dynamically adjusts the time step for each segment of the orbit so the orbit remains smooth throughout, yet no time is wasted during distant, slow-moving sections of the orbit. The time saving from this were rather dramatic. For a bad-boy slightly hyperbolic Sun-grazer comet, orrery calculation time dropped from 880 seconds to a little more than 7 seconds. Music of the spheres to my ears.

I replaced the year-old satellite element databases with versions current as of mid-October 1994. Of course, they'll soon be out of date, but it's better than including hopelessly obsolete data.

13 October 1994

More fixes to the utility program ASTRELEM.C to handle additional quirks observed in the elements included in Minor Planet Electronic Circulars (MPECs).

Objects (for example just-discovered Pluto crossers) with an undetermined eccentricity are given as E = 0.00000 but with the mean anomaly (M) field blank. Home Planet handles a blank mean anomaly correctly but other programs may not, so I fixed the code to include an explicit "0.0" for mean anomaly if it is unknown.

Elements for recovered objects include an "Id." line as the second line (between the name and the Epoch) and omit the "From" line at the end. These are now processed correctly.

The orbital velocity determination code in ORRERY.C could divide by zero if Kepler's equation failed to converge for near-parabolic objects with mean anomalies near 180°. I modified ASTEROID.C to guarantee that the radius vector is returned as zero to indicate the orbital position calculation failed (this is our signal elsewhere, as in PLUTO.C), and fixed ORRERY.C to treat this failure as marking the end of the current branch of the orbit.

I further refined the calculation of time steps based on orbital velocity so that the step size for the inner solar system is always used when the object is within the inner system. This results in much smoother orbits near perihelion, without compromising speed in the much longer outer system branches. Since very eccentric orbits don't spend much time in the inner system, we can afford the additional precision there.

I fixed the horizon view to work properly in both the northern and southern hemispheres. Since in horizon view we're working in altitude and azimuth co-ordinates, which don't depend upon the observer's hemisphere (that being accounted for in the transformation from equatorial to local horizontal co-ordinates), all the vertical flipping and mirroring logic done in the Sky and Telescope views had to be bypassed. In addition, the inverse transformation from horizontal to equatorial co-ordinates done when a left click to aim the telescope or a right click to identify an object ran afoul of an error in the next to last equation on page 89 of Meeus' "Astronomical Algorithms" which should read:

$$\tan(H) = \sin(A) / (\cos(A) * \sin(\text{Phi}) + \tan(h) * \cos(\text{Phi}))$$

Added a Terrain configuration section to the Options dialogue for the Horizon panel. This allows enabling or disabling the synthesised terrain at the horizon, setting the fractal dimension of the terrain (its roughness), and choosing whether "scenery" such as cows, pigs, tractors, houses, etc. is drawn along the horizon. (These parameters are not yet saved in the settings file--that will be done in the final closeout of the settings file.)

14 October 1994

Updated the Spacecraft database (spacecrf.csv) to include all objects launched through the end of 1993, and added apogee, perigee, orbital period, and inclination where known. This database is derived from TS Kelso's SDF satellite database, using a C program (in the TOOLS subdirectory) called SATDBC.C to convert to CSV format, then SORT to put in alphabetical order.

Completed the first cut at the Horizon window and its Horizon options dialogue. This window, activated from the View menu of the Sky window, shows the sky above the horizon at any given azimuth for an observer at the configured observing site. The code that implements the Horizon window is very similar to that of the Telescope, except that co-ordinates are transformed from equatorial to horizontal (altitude and azimuth) before display. This introduces a few interesting wrinkles, the messiest of which is calculating how many segments to subdivide constellation boundaries that run along parallels of declination so that

they are drawn curved relative to the sky, not the horizon. Left clicking in the horizon window aims the telescope to the designated location, and right clicking identifies the closest object pointed to in the Object Catalogue (these operations require transforming back from horizontal to equatorial co-ordinates, naturally. The default field of view is 45%, corresponding to the high resolution portion of the eye, but you can configure larger or smaller fields of view, limited only at the point the projection would introduce ridiculous distortion.

The Horizon Options dialogue is very similar to the Telescope Options, except that the South up check box is missing (it would be equivalent to a "Stand on your head" option, which I don't think would be useful) and replaced by a box that controls whether fractal forged terrain is drawn along the horizon and, if enabled, its fractal dimension (lower numbers give smoothly rolling hills, larger numbers rougher, more mountainous terrain) and whether the terrain is decorated with scenery such as cows, houses, and the like. All the options for configuring the objects shown in the sky are identical to the Telescope window.

15 October 1994

To allow user customisation of the scenery icons optionally drawn if Terrain is enabled in the Horizon window, I moved the icons and all the logic for selecting them into a DLL called SCENERY.DLL, which is maintained as a separate project in directory SCENERY. The user can perform simple customisation of the scenery simply by directly editing this DLL with a resource editor such as Application Studio. The DLL contains four numbered string resources:

```
1       "15         ; Number of icons to use at horizon"
2       "32         ; Size of icons"
3       "10         ; Density to sprinkle icons"
4       "16         ; Total icons in file"
```

and sixteen icons with resource numbers 1 through 16. You can edit any of the existing icons, replace them with new icons, and/or add new icons, increasing the number in string resource 4 (only the number is scanned; the comment is ignored). If the additional icon is eligible for random selection for drawing at the horizon, increase the number in string resource 1 as well. Icons with indices above that given by string resource 1 can be plotted in sky maps with the "Hnnn" phrase in a User Catalogue entry.

The icon size is specified by the second string resource, but this must presently be left as 32. The third string resource controls the density with which icons will be placed on the terrain. The chance of an icon appearing at a given pixel along the horizon is 1/(icon_size * icon_density), the parameters given by string resources 2 and 3. Thus, with the default values, an icon will be drawn, on the average, every 320 pixels along the horizon (note, thus, that decreasing the density parameter increases the frequency with which icons appear and vice versa). The scenery drawing code automatically guarantees that icons are not drawn on top of one another. If the image in an icon does not fill the 32×32 pixel icon frame, it should be drawn justified against the left and bottom of the icon frame.

Much more sophisticated customisation of scenery generation is possible by replacing the supplied version of SCENERY.DLL with a user-defined version. The interface between Home Planet at this DLL is through the following functions. Whenever the horizon window is redrawn and the user has enabled terrain and scenery, the function:

```
void FAR PASCAL _export sceneryInit(
            double julianDate,              // Time and date
            double siteLat,                 // Observer latitude
```

```
    double siteLon,             // Observer longitude
    double viewAzimuth,         // Azimuth of window center
    WORD imageHeight,           // Image height
    WORD imageWidth,            // Image width
    WORD randomNumber,          // A 15 bit random value
    WORD FAR *numIcons,         // Return: Number of icons
    WORD FAR *iconSize,         //   Icon size
    WORD FAR *iconDensity);     // Icon density
```

is called. This provides the DLL complete information as to the time and date (real or simulated) at which the horizon is being drawn, the

latitude and longitude of the observer, what direction the observer is looking, the height and width of the horizon window in pixels, plus a random number between 0 and 32767 that the DLL can use any way it wants. The function returns three values through pointers: the number of icons it can generate (if zero, scenery is disabled), the size of the icons provided, and the density parameter as described above. Although the SCENERY.DLL provided with Home Planet uses none of the information provided to sceneryInit, I'm sure you'll see how this information can be exploited to show leafy trees in summer, falling leaves in fall, and bare limbs in winter, kangaroos if the observer is in Australia, and more.

Each time Home Planet decides, at random, to place an icon along the horizon, it calls:

```
void FAR PASCAL _export sceneryIcon(
    WORD xPos, WORD yPos,       // Where icon is to be drawn
    WORD randomNumber,          // A 15 bit random number
    HICON FAR *hIcon,           // Return handle to icon
    WORD FAR *hWidth);          // Width (advance) after this icon
```

informing SCENERY.DLL of the X and Y co-ordinates of the top left corner of the icon within the horizon image pane and providing a random number between 0 and 32767. sceneryIcon should return the handle of the icon to be drawn in hIcon, and the width of the image within the icon in hWidth (this is used to keep icons from being drawn atop one another). The icon will be destroyed by Home Planet after being drawn into the terrain.

At the end of generation of the horizon image Home Planet calls:

```
void FAR PASCAL _export sceneryTerm(void);
```

to allow SCENERY.DLL to clean up any items it might have allocated in sceneryInit. Home Planet guarantees that SCENERY.DLL will not be called from any other application or instance between sceneryInit and sceneryTerm, allowing it to use local storage within the DLL without conflicts.

16 October 1994

I added three new catalogue types to the object catalogue: 6, 7, and 8. These all behave like type 3, in that if records contain fields labeled "Right ascension" and "Declination" they may be used to aim the Telescope with the Select button, and that items within them may be found by right clicking in the Telescope and Horizon windows. Which catalogue(s) are searched depends upon the state of the keyboard SHIFT and control keys when the right mouse button is clicked as follows:

Keys down	Catalogue searched
none	3 + special

SHIFT	6
CONTROL	7
CONTROL+SHIFT	8

This allows instantaneous switching between different classes of objects to be searched. Taking advantage of this new facility, a catalogue of Palomar Observatory Sky Survey plates was added to the Object Catalogue with code 6. Selecting a plate from the Object Catalogue aims the Telescope to its centre. Right clicking in the Telescope or Horizon with the SHIFT key down displays the POSS plate which contains the location clicked (assuming the declination isn't further South than the coverage of the POSS, of course). Note that the special databases (asteroids, comets, and solar system objects) are searched only if no keyboard modifiers are in effect when the right click occurs. If you shift-click on an asteroid to see the atlas field in which it appears, you don't want the search to point to the asteroid!

Open architecture computerised telescope pointing has arrived! Home Planet (application "HPlanet") now exports a new topic, "Telescope" with the following items:

Name	Type	Description
Active	Integer	1 if Telescope window displayed, 0 if not
RA	Double	Right ascension (in degrees) of telescope aim point. Divide by 15 to get hours.
Dec	Double	Declination (in degrees) of telescope aim point.
Azi	Double	Telescope azimuth, measured from the South.
Alt	Double	Telescope altitude.

A control program for a computerised drive telescope may subscribe to these items and, when Active is nonzero, indicating that the Home Planet Telescope window is displayed, respond to changes in the telescope aim point by steering the telescope's drive to the designated co-ordinates. The altitude and azimuth are provided along with the equatorial co-ordinates to simplify the task for rudimentary drive hardware. A notification is sent to subscribers to these items whenever the Telescope is pointed. (Note, however, that the azimuth and altitude are not updated in real time; the physical telescope is assumed to be endowed with at least a clock drive. If folks find this an onerous limitation, it can easily be eliminated.) Note that a notification can be sent even if the aimpoint has not actually changed; notification is based upon aiming commands, not changes in aimpoint.

The Excel DDE demonstration spreadsheet, HPDDE.XLS, now includes a Telescope section which shows the Telescope DDE exports. In addition, I added logic to display right ascension as hours, minutes, and seconds; and declination as degrees, minutes, and seconds. These alternative display formats were also added to the Sun and Moon topic displays.

17 October 1994

Added a "Related software" item to the Help menu to plug other programs that Home Planet users may be interested in.

Eliminated an inadvertent reference to an Evil Empire in the Help/Time Zone dialogue, supplanting it with that ever handy euphemism "North America".

Added a new catalogue to the Object Catalogue, "Sky Atlas 2000.0", which

contains the chart numbers, chart centre points, and right ascension and declination range of the 26 charts in Wil Tirion's "Sky Atlas 2000.0", available from Sky Publishing. This catalogue is installed with database type 7, which permits, pointing to an object in the Telescope or Horizon windows, determining which Sky Atlas 2000.0 chart contains the object simply by depressing the CONTROL key and right clicking over the object. Since selection of the chart is by minimum distance to the chart centre, if multiple charts contain an object (quite likely, since Sky Atlas 2000.0 contains substantial overlaps), the one chosen will be that in which the object appears closest to the centre of the chart. The object catalogue database file, SKYATL2K.CSV, is derived from an Excel generator database in TOOLS\SKYATL2K.XLS.

19 October 1994

I completed another new Object Catalogue, "Palomar Sky Survey", which lists all the plates of the Palomar Observatory Sky Survey (including the Whiteoak extension), providing most of the information about the plates in the Brotzman and Hill plate catalogue on the ADC CD-ROM. Thanks to Glen Deen, who kindly provided the mapping between POSS plates and MicroSky microfiche numbers, each plate entry includes the fiche and field number where the plate appears in MicroSky. This catalogue is keyed to the SHIFT right-click, so right-clicking on an object in the Telescope or Horizon will pop up the object catalogue description of the POSS plate containing the object. To avoid confusion, I deleted (commented out, actually) the entry for the plates with Luyten number 723, which were exposed with an incorrect centre point and subsequently reshot as Luyten number 724.

In order to accommodate the long records in the Palomar Sky Survey catalogue, I increased the maximum record length in OBJCAT from 175 characters to 350.

I also enabled automatic vertical scrolling in the Object Catalogue information box, even though no existing catalogue contains enough fields to require it.

I completed the Object Catalogue index to Uranometria 2000.0 (Tirion, Rappaport, and Lovi). This catalogue provides chart centres for all 473 charts in the two volumes of Uranometria 2000.0, and allows aiming the Telescope to the chart centre and right-click identification of the chart containing a given object.

I revised the assignment of right-click keyboard modifiers to catalogues. When you right-click, the catalogues searched are as follows:

Keys down	**Catalogue**
None	Normal object catalogues
SHIFT	Sky Atlas 2000.0
CONTROL	Uranometria 2000.0
SHIFT+CONTROL	Palomar Observatory Sky Survey

20 October 1994

When Edit/Default Settings reset the display mode, it did not multiply the BITSPIXEL by PLANES, and hence would improperly construe some 16 colour displays to be monochrome. Fixed.

Once the user set the display mode, it was impossible, even by restoring the Default settings, to return to the "auto-sense" mode in which Home Planet starts up (where the colour mode in the Map window is determined from the colour capability of the display). I modified "displaymode" to distinguish an auto-sense mode from a user specified one and, when Save settings is done, zero displaymode in the .INI file if the mode was obtained by auto-sensing.

Default settings didn't repaint the map window when the display mode changed as a result of restoring the default settings. Fixed.

Default settings didn't repaint the sky, telescope, and horizon windows when a relevant mode changed. Fixed.

I added the settings for the Horizon window to the list of settings saved in the .INI file and restored by Default settings.

22 October 1994

I moved the star maps (Yale and SAO) from the resource of HPLANET.EXE into a separate resource-only DLL called STARMAPS.DLL, which is maintained in the new subdirectory STARMAPS. There were several reasons for doing this. First of all, it reduces the size of HPLANET.EXE from 4 megabytes to a little more than 1. This makes it easier to split up the distribution into separate archives to avoid the FTP timeout problems people have been reporting with the one huge .ZIP file. Second, it avoids copying 3 megabytes of starmaps in the .EXE file every time you link, which speeds up the development cycle substantially. Third, it permits providing a "Lite" version of Home Planet with just the Yale star map, which would be 2.8 megabytes smaller in the complete distribution. People could try this version and then later, if they wished the complete SAO catalogue, separately FTP a replacement STARMAPS.DLL and install it. Finally, this provides a foot in the door toward opening the architecture of the star maps, where SKY.C would ask the starmaps DLL to enumerate the maps it provided and dynamically generate the map selection menu from that list.

To support the forthcoming "Chart Catalogue" feature, I added the ability to include anonymous items in an Object Catalogue. If the first field in a CSV catalogue item is void, that item doesn't appear in the catalogue list box. This requires some fancy footwork when a selection is made, since when we scan the database file to locate the record for the selected item, we have to skip database records that don't appear in the list box.

24 October 1994

The creation of the device dependent bitmap in the Object Catalogue if an (Image) .BMP file is specified was using the DIB_RGB_COLORS mode, which forced mapping of every pixel. I modified it to, in the process of constructing and realising the bitmap's palette, built a palette index table which is used in the CreateDIBitmap call with the DIB_PAL_COLORS mode. This substantially speeds up the display of object catalogue images on colour mapped displays.

Fixed some more palette-juggling dropped balls. When the Sky window was launched and realised its palette, the Map window lost its palette and was not repainted until it received the focus again. I made it respond to the WM_PALETTECHANGED message and repaint itself (and its children's) windows. This also corrected a similar problem where the Object Catalogue temporarily messed up the Map window's palette.

The View From window was clearing its handle in the WM_CLOSE handler rather than WM_DESTROY, which created a tiny hole where a bad window handle could be used. Naturally, Windows slime leaked through and caused debug messages. Fixed.

26 October 1994

Added a gimmick to the user catalogue to expedite plotting of full circles. If the start and end right ascension of a curved line are both zero, automatically plot a circle. This circumvents the assumption made otherwise that an arc is always

less than 180°.

Added logic to automatically subdivide arcs with long extent to avoid the clipping logic causing them to disappear.

Fixed plotting of curved lines from user catalogue in horizon view by forcing additional subdivision before projection into altitude and azimuth.

27 October 1994

Modified the Power, Field of view, and Limiting magnitude fields in the Telescope window to subclass the edit control and update the telescope when the Return or Tab keys are pressed. This eliminates the need to use the "Aim" button after modifying these fields.

I fixed the co-ordinate transform in the Horizon window to calculate the logical (alt-az) aimpoint when the Horizon window is resized to be taller than wide. Before scaling was calculated exclusively based on the width, which clipped at an incorrect altitude in this case.

28 September 1994

Fixed less-than-logical tab ordering of fields in the Set Universal Time and Set Julian Date dialogues.

I finally beat the handling of keyboard input within edit controls into a minimally satisfactory form for the Telescope. Previously, when you changed one of the telescope parameters, the scope did not get updated until you pressed the "Aim" button. This was really ugly, but it avoided the delay for multiple updates if you changed a sequence of parameters one after another. I modified it so that you can tab back and forth between the Right Ascension and Declination boxes with Tab and Shift Tab, without forcing an update until you move outside those two boxes or press RETURN. The other boxes automatically update when you press RETURN or take focus away from them with the mouse, but only if a change has occurred. Finally, I allowed the telescope to be pointed by a single click within the window. This makes the Telescope consistent with pointing in the Sky and Horizon windows. I disabled the mouse-down drag position readout in the Telescope, as this could cause double refreshes if the mouse moved during repaint.

For further consistency, a single click in the Sky window will now launch the Telescope if it's not already displayed.

Fixed some more bugs with plotting of curved lines from the user catalogue in the Sky, Telescope, and Horizon windows.

I added a new interface function to SCENERY.DLL:

```
void FAR PASCAL _export getSceneryIcon(
        WORD iconNumber,        // Icon number to get
        HICON FAR *hIcon);      // Handle to icon or NULL = error
```

which returns the handle to the icon iconNumber defined with the DLL in hIcon. If the iconNumber is out of range or, for whatever reason, the icon cannot be loaded, NULL is returned. It is the responsibility of the caller to destroy the icon after use with DestroyIcon. This allowed implementing the "Hnnn" phrase in the chart catalogue, which selects the icon with the given number from the scenery DLL. Since the scenery DLL is user-extensible, this completely opens the range of icons which can be plotted from the chart catalogue. Note that the icon number given in the H phrase can be multiple digits.

29 September 1994

Updated the CSV asteroid databases from the master TOOLS\ASTEROID.XLS worksheet. Fixed the labels of the numbered asteroid databases in OBJECTS.CSV to be "1-4000" and "4001-5632" to agree with the contents of the files.

Moved the capital of Belize from Belize City to Belmopan (moved inland due to hurricanes).

Added a new View from option, "Above observatory", which looks down upon the observer's location from one astronomical unit. This lets you see (assuming you're using the full colour map) the terminator with respect to your location and the advance of night and day in neighbouring regions. The title in the View From panel says "View above <location>", and the altitude is suppressed in the viewpoint field, as only the latitude and longitude are relevant.

Changing the Observer's Location didn't force a repaint of the View From window. Fixed.

30 September 1994

Implemented optional display of star magnitudes in the Telescope (only) window. The telescope options dialogue now contains a new check box, "Show magnitudes" with edit controls for "Max" and "Min". If the box is checked and the magnitude of a star falls between the given Max and Min values (to avoid confusion since magnitude numbers are inversely related to brightness, the values may actually be entered in any order), its magnitude is plotted in the Telescope display centred beneath the star. Following the convention for such maps (for example variable star comparison charts), magnitudes are given to the nearest tenth with the decimal point omitted (decimal points are too easily confused with stars). Thus, a star of magnitude 4.57 will be labeled with magnitude "46". Magnitude display is disabled by default. Whether magnitudes are displayed and the minimum and maximum limits are saved in the .INI file by Save Settings.

A mere several hours debugging were required to discover that a user-defined resource with data supplied directly within the RC script (as opposed to being imported from a binary file) will, if it exceeds 1024 bytes, totally screw up the structure of the resource file and crash any program that accesses it in a variety of innovative ways. This happens, of course, without the slightest warning or error message from RC nor, as one expects when dealing with Microsoft, any indication of such a limitation in the documentation. I stumbled into this banzai buggeroo when the settings variable table crossed the threshold of doom. I moved the settings to an external file called SETTINGS.TXT, which is imported into the resource by a reference in SETTINGS.RC. I modified SETTINGS.C to parse the CR/LF delimited text file, supplanting the semicolon delimited string used before.

Corrected longitudes of Australian cities. An earlier update had accidentally entered them as West longitude rather than East.

The year field was randomly ignored in the Set Universal Time dialogue because of an obsolete check on "ok" which should have been removed when the year was changed to a long to support deep time. Fixed.

31 October 1994

The screen saver hadn't been rebuilt since the conversion from Quick C to Visual C++, so I decided the time had come to smoke out the inevitable bugs. The major effort required was constructing a new resource definition, HPSSAVE.RC, since App Studio doesn't allow conditional compilation of the resources it maintains (boo, hiss!). The tradeoff is not pleasant--either maintain

two separate resource definitions which largely duplicate one another, or sacrifice visual editing of the resource. With an eye to easier editing when the source code is published, I decided to go with App studio and make a duplicate resource for the screen saver. I reviewed the #ifdefs in ASTRO.C and discovered I could turn off the new nutation calculation function which isn't used by the screen saver. After final tweaking and production compilation with "Minimise size" options selected, the resulting HPSSAVE.SCR was 9K smaller than the original Quick C version. There is absolutely no change in functionality in this release; it's just nice to ship a version built from the same source, compiled with the same compiler used to build Home Planet.

Multiple-digit references to icons in SCENERY.DLL from the User Catalogue with the "Hnnn" phrase weren't working. Fixed. Also, while I was at it, I rebuilt SCENERY.DLL with production, rather than debug, options.

Added ASTRELEM.EXE and ASTRELEM.C to release, and rebuilt ASTRELEM.C in production mode and re-tested.

Updated site locations for New Zealand cities using co-ordinates provided by Sam Crupi.

2 November 1994

Finally debugged the text alignment mode specification and the related accurate justification based upon text extents for the Chart Catalogue.

Okay, I guess that about does it for the Chart Catalogue, so let's document how to use the beast. A Chart Catalogue is a CSV format object catalogue, accessible from the Object Catalogue panel which contains fields named "Right ascension" and "Declination" and at least one of two fields named "(Label)" and "(Type)". The (Label) and (Type) fields do not appear in the Object Catalogue panel. The first field in the CSV record is used, as always, as its label in the Object Catalogue. If this field is blank (in other words, the CSV record begins with a comma), the record will not be included in the Object Catalogue list box.

When an object catalogue which meets the above requirements is selected, the "Chart" check-box to the right of the "Category:" drop down list box is enabled. If you check it, any displayed Sky windows (Sky, Telescope, or Horizon) are re-drawn with the objects in the selected Chart Catalogue plotted at the positions given. Positions in a chart catalogue are specified in J2000.0 and precessed, if necessary, to the epoch being plotted. No provision is made for proper motion of chart catalogue objects.

What is plotted at the given co-ordinates is specified by the (Type) field, which selects an icon, and the (Label) field, which specifies text to be drawn at the given location. Either of these fields may be void on a record by record basis. Icons in the Type field are selected by the following codes. Multiple items may be plotted in a (Type) field by separating them with semicolons.

Deep sky icons:

DK	Dark nebula
DN	Diffuse nebula
DS	Double star
EG	Elliptical galaxy
GC	Globular cluster
IG	Irregular galaxy
OC	Open cluster
PN	Planetary nebula
QS	Quasar
SG	Spiral galaxy

Star icons:

S1	Smallest star icon
...	
S9	Largest star icon

Planet icons:

P0	Sun
P1	Mercury
P2	Venus
P3	Earth
P4	Mars
P5	Jupiter
P6	Saturn
P7	Uranus
P8	Neptune
P9	Pluto

Satellite icons:

B0
...
B9

Scenery (horizon) icons:

Hnnn (nnn: 1 or more digits. 1 = first icon)

In addition to icons, user catalogues may draw lines on the sky map, either straight lines between co-ordinates or lines that follow parallels of celestial latitude and meridians of celestial longitude. Lines can be drawn in any of 10 predefined colours. The line drawing commands are:

Cn	Set pen colour to n, where 0 = black 1 = red 2 = yellow 3 = green 4 = cyan 5 = blue 6 = magenta 7 = white 8 = light grey 9 = dark grey
MV	Move to
LT	Straight line to
CT	Curved line to

The (Label) field supplies text to be drawn at the given co-ordinates. If a simple non-quoted string is given, it is drawn left-justified and centred vertically to the right of the co-ordinates. You can control the justification and colour of text by enclosing the text in single quotes (a single quote can be included in the string by writing two single quotes in a row) with text mode specification characters following the closing quote:

Horizontal alignment:

<	Left justify
>	Right justify
|	Centre

Vertical alignment:

^	Top

- Middle
_ Bottom

Colour:

0-9 as for line drawing above

Chart catalogues can be used to add arbitrary information to the sky map displays. Note that since icons can be selected from the Scenery DLL used by the Horizon window, to which icons may be added by the user, that the selection of icons is open-ended.

Here's the first few lines of a chart catalogue that has no items for user selection, but which plots the chart boundaries of Sky Atlas 2000.0 and displays each chart number at its centre.

```
,Right ascension,Declination,(Type),(Label)
;    +50 parallel
,0h,50d,c3;mv
,0h,50d,ct
;    +50 - +90 meridians
,0h,50d,mv
,0h,90d,lt
,8h,50d,mv
,8h,90d,lt
,16h,50d,mv
,16h,90d,lt
,4h,70°,,'1'|-1
,12h,70°,,'2'|-1
20h,70°,,'3'|-1
;    +20 - +50 meridians
;              . . .
```

Chart catalogues, thus, permit arbitrary user-defined information to be added to the sky maps with these items, at the user's discretion, selectable from the Object Catalogue and identifiable by right clicking.

To illustrate the chart catalogue facility, I created a catalogue named 3C.CSV which contains all the radio sources from the 3C catalogue which were cross-identified in the 4C survey. This catalogue (configured "not for searching: 5" in OBJECTS.CSV to avoid confusing misidentifications) uses the radio source icon (H16) from SCENERY.DLL to illustrate access to icons defined there.

4 November 1994

The currently selected object catalogue was supposed to be remembered by Save settings but, due to the variable used by SETTINGS.C never having been set in OBJCAT.C, wasn't. Fixed.

Added a new Object Catalogue category, 9, intended primarily for use by Chart Catalogues. Objects in category 9 catalogues are expected to have Right ascension and Declination fields and can be used to aim the telescope with the Select button. If a (Type) or (Label) field is present, they can be charted in the sky maps as described above. Right-click selections, however, are only made in category 9 catalogues if they are currently charted at the time the right click is made. This permits extensive optional catalogues from capturing identification requests when the objects within them aren't visible on the map.

The 3C catalogue of radio sources is now a category 9 catalogue.

The check box legend "Ecliptic, equator, poles" had the "s" truncated when displayed on a vanilla VGA. Fixed.

The help key used by the Help button in the Telescope options dialogue had a comma missing and failed to connect to the item in the help file. Fixed. I also slightly revised the keywords use to access the Horizon and Horizon options help topics to be a little more usable as keywords in their own right.

6 November 1994

I implemented COMETEL.C, a new MS-DOS application in the TOOLS directory. This is the equivalent of ASTRELEM, but for comets. It scans a file containing one or more sets of cometary elements in the form issued in the IAU circulars, and extracts and reformats them into a ready-to-use .CSV file which can be loaded as a comet object catalogue. Unlike ASTRELEM, which due to the less well-defined form in which asteroid elements are published, COMETEL can digest whole circulars in raw form without any cutting and pasting. Further, it can be fed entire mail folders in which circulars have been archive along with other irrelevant material. All valid cometary elements will be extracted into the output file.

Fixed truncated text in About dialogue on vanilla VGA. Sigh.

I really liked the convenience of being to feed a whole mail folder archive to COMETEL, so I cranked up the hack-o-matic and massaged ASTRELEM until it, too, can digest a mail folder and sift the asteroidal elements from a mass of irrelevant material. I added a "x records written." message to stderr in both ASTRELEM and COMETEL to indicate the degree of success. If ASTRELEM is later found to be passing "false positives", the format checking can be tightened up quite a bit--for the moment I've left it pretty loose due to prior experience with small inconsistencies from one MPEC to another.

7 November 1994

Updated COMETS.CSV to include principal periodic comets with perihelion dates in 1993 and 1994, and renamed the category "Comets" in OBJECTS.CSV to "Comets: periodic".

The "View from" window wasn't being receiving periodic update notifications as the terminator moved when set to "View above observatory." Fixed--twice, as it turned out. First I had to make sure that the terminator's moving triggered an update to the View From window, but also I had to suppress the optimisation based on the viewing location not having moved. Note that this fixes a bug whereby the terminator wouldn't be updated for geostationary satellites as well.

The orrery wasn't being automatically updated when the user changed the selection of an asteroid or comet to track. Fixed.

9 November 1994

Both SUNCALC.C (map window) and SATVIEW.C (view from window) assumed the width of a line in the map bitmap was a multiple of a LONG and didn't add the required pad if it wasn't. This tripped up attempts to use custom map bitmaps which don't happen to be a multiple of 4 pixels in width. Fixed.

10 November 1994

Re-integrated the VERSIONINFO resource which got lost during the switch from a hand-coded .RC file to App Studio. The version is now defined in the file VERSION.RC, which is included verbatim in the resource definition of both Home Planet and the screen saver. Since it's an include file, it can use conditional compilation based on ScreenSaver to act as a version definition for both programs.

Cleaned up all the *.DLG and *.RES files left over from Quick C-style dialogue

editing. They are now all consolidated into the .RC file maintained with App Studio.

Integrated resource IDs definitions into RESOURCE.H (for Home Planet) and HPSSAVE.H (for the screen saver), and edited the corresponding .RC files to eliminate unnecessary hard-coded numeric references to resource IDs.

Added logic in SUNCALC.C and SETVIEW.C to use DIB_PAL_COLORS consistently, and to replace the previous horrid practice of copying the palette mapped colour table over top of the bitmap in the resource with making a new header in a separate buffer. This both speeds up the display of the map window and the View from window, it also fixes several palette juggling bugs encountered previously.

If the Horizon view window was displayed before the Telescope, a bad procedure instance was used to subclass the Tab key input to the right ascension and declination edit fields in the Telescope window. Changed to create the Return and Tab key procedures independently, as needed in SKY.C.

Verified correct handling of no TZ specification for screen saver.

If no TZ was specified for Home Planet, the warning message was truncated due to an apparent (undocumented) string length limit in MessageBox. I reworded the message to squeak under the limit.

11 November 1994

The calculation of when to update the map window based on the terminator's moving was done, when using a bitmap image, based on the size of the bitmap rather than the current size of the map window. If the source bitmap is much larger than the window, this leads to unnecessarily frequent updates which aren't visible at the scale of the window. I changed the calculation to always be based on the current window size, as has always been the case when the vector map is selected.

When the user selects a grey scale bitmap map window, the code in SUNCALC.C used to transform the palette in the bitmap header into grey scale values, then build the system palette from that. Unfortunately, if the user switched back to colour and the bitmap resource was still in memory, the grey scale palette would continue to be used. In addition, the code was writing into an in-memory copy of a loaded resource, which is distinctly antisocial. I changed the logic to flatten the palette into grey scale at the time the palette is built, leaving the palette in the bitmap header intact.

The large tables of coefficients in VSOP87.C, which had to be moved to a separate data segment to avoid overflowing the default 64K DGROUP caused Home Planet to be deemed a "large model" program and thus incapable of running multiple concurrent instances. To circumvent this problem (after

a large number of false starts), I changed the data segment of VSOP87.C to __based(__segname("VSOP87D")) and then forced that segment to be READONLY in the .DEF file. At the same time, I made all of the CODE segments other than the entry point segment LOADONCALL rather than PRELOAD, which makes the program initial load much faster. With the change to the auxiliary data segment, multiple instances run with no problem.

To open the architecture of the map window bitmap specification, I moved all the map window bitmaps into subdirectory MAPBITS, where a DLL is built which includes all the bitmap as resources and implements an API used by the main program to access them.

The file MAPBITS.DLL allows the definition of any number of Earth map

bitmaps. Each is assumed to be a file containing a Mercator projection

of the globe, centred upon Greenwich (0° longitude), twice as wide, in pixels as high, as befits a Mercator map. You can customise the library of Earth images either by directly editing the MAPBITS.DLL file with a resource editor such as App Studio, or by replacing the supplied version of MAPBITS.DLL with one of your own.

To edit MAPBITS.DLL, first examine the String Table, where you'll find a string resource:

```
1       "2          ; Number of bitmaps in DLL"
```

If you're adding a bitmap, increase the number in this string resource. Only the number is relevant; the comment is for documentation only.

For each bitmap in MAPBITS.DLL, there is a string resource, numbered from 100, which specifies the description of the bitmap that appears in the Display/Map image menu item and a bitmap resource bearing the same number which contains the actual bitmap. To add a new bitmap to MAPBITS.DLL, edit string resource 1 to increase the number of bitmaps in the file, then add the description string resource and the bitmap resource to the file.

Map window bitmaps have a special format upon which the display code counts--if you develop new bitmaps for the map window, you must be sure they conform to its expectations. All map window bitmaps contain at most 117 unique colours. The palette in the bitmap consists of up to 117 "subdued" colours starting at palette index 0, which are used to paint night regions of the Earth, a filler of the size required, and then starting at palette index 128, up to 117 "day" colours used to paint the portion of the Earth illuminated by the Sun. Pixels in the bitmap are assumed to have the 0x80 bit set, selecting the day portion of the palette.

To replace MAPBITS.DLL with a compatible library, you need to implement the following functions:

```
extern int FAR PASCAL _export mapbitsCount(void);
```

Returns the number of bitmaps defined within the DLL. Return 0 in case of error.

```
extern void FAR PASCAL _export mapbitsGetBitMap(
            WORD bitmapNumber,      // Bitmap number to get
            HANDLE FAR *hBitmap);   // Handle to bitmap or NULL =
error
```

Returns a handle to the bitmap, as obtained by LoadResource(). Note that this is *not* the result of LoadBitmap! Home Planet needs to examine the original Device Independent Bitmap in detail, and thus must look directly at the in-memory copy of the bitmap resulting from direct access to the resource.

```
extern void FAR PASCAL _export mapbitsGetDesc(
        WORD bitmapNumber,          // Bitmap description to get
        LPSTR sbuf, UINT buflen);   // Buffer and length
```

Returns, for the bitmap with the given bitmapNumber, the description from its corresponding string resource, stored into string sbuf, with a maximum length of buflen. If an error occurs, sbuf will be set to the null string.

12 November 1994

If the user displayed the Telescope or Horizon options dialogues, which are modal, the parent window of the dialogue was disabled, as usual, but the Sky

and Map window remained active. This was rather nice, since you could refer to them and launch other windows while the options dialogue remained outstanding. What was not nice is that if you terminated either the Sky window or Home Planet entirely, the following message appeared on the debug terminal:

err HPLANET USER: Dialog should be dismissed with EndDialog, not DestroyWindow

Well, after spending 3 days, off and on, trying various ways to fix this in an elegant manner (hint: all the neat tricks you think of that involve processing WM_CLOSE in the main window and sending a WM_CLOSE or WM_COMMAND / IDCANCEL to the dialogue don't work), I beat it to death with a sledgehammer by disabling both the Sky and the Map windows while this dialogue is outstanding. This, in essence, makes the dialogue truly application modal, locking out all input to any window that could terminate the dialogue behind its back. You can still get to other windows such as the Object Catalogue, etc. but since they can't terminate the Sky or Map window, no harm can be done by accessing them. It's a bloody shame to lose the ability to go back to the Sky or Map, but a Windows developer soon becomes acclimated to both blood and shame.

The code that processes the WM_WINDOWPOSCHANGING message for the Sky window to force the sky map to be square was incorrectly interpreting the size passed to it as the client area, not the entire window. This caused the Maximise box to set the window height larger than the screen by the height of the window title, menu, and frame. No harm done, but it was ugly. Fixed.

The automatic font sizing for the sky map windows was generating fonts that were too large when the windows filled most of a high-resolution screen. I changed to logic to enforce a minimum size for legibility and scale less rapidly with screen size.

Mine eyes have beheld the Gates of Hell, and on them is writ the number C8.0. Ever since I posted the Sky screen saver, people with machines that lack a coprocessor have reported bizarre errors which I was never able to reproduce even when I set NO87. Finally, I pulled the coprocessor chip out of my back-up Compaq and discovered that, indeed, the problems were real. But what could be the cause? Well, after about 12 hours of total debugging over a month or so, I've staked the bloodsucker, only to discover, thanks to the Microsoft Knowledge Base (propriety does not permit me to comment upon the various connotations one might attach to that phrase), that it's a "known problem" for which a "work-around" exists. Well, what is this arcane problem anyway? Certainly some strange code optimisation bug, interaction with system configuration in an unexpected way, or something else that is almost inevitable in a large software system. No. The problem is that if you don't have a coprocessor the tan() function returns values with the wrong sign for arguments in the domain -Pi/2 to Pi/2. The Knowledge Base states that the reason for this error is "unknown". Note that this is not, or at least not exclusively, an error in emulation of the coprocessor. The Quick C math library, and the C 7.0 library did not encounter this error--it's new in C 8.0, even though all use the same Windows 80x87 emulator DLL. Well, there's nothing for it but to include the definition:

```
#define MICROSOFT_CAN_T_COUNT_WORTH_BEANS
#ifdef MICROSOFT_CAN_T_COUNT_WORTH_BEANS
#define tan(x)   (sin(x) / cos(x))
#endif // MICROSOFT_CAN_T_COUNT_WORTH_BEANS
```

in the master include file and soldier on, musing that if I were the world's leading software company and tens of millions of people were counting on my software tools and were using them, naïvely perhaps, in mission critical applications, I

would subject my new releases to extensive regression testing on configurations representative of the bulk of my customer base before shipping the product.

Rebuilt MAPBITS.DLL in production mode.

13 November 1994

Converted the development log (this file) from ASCII text to RTF, and marked it up so it can be compiled into a LOG.HLP file accessible from the main help file. As a help file, it's pretty primitive--just one big topic, but given that the log is just a linear narrative, fancier organisation doesn't seem justified and may be counterproductive. The LOG.HPJ Help project file redefines the "Contents" button in the toolbar to go back to the contents of the main help file, rather than the start of the log. This, along with a link to the log in the contents page of the main help file, maintains the illusion that the log is simply another topic in HPLANET.HLP. Keeping the log as a separate .HLP file allows redistributing just the log with increment updates that don't require a change to the main help file.

15 November 1994

Integrated all markups to HPLANET.HLP and recompiled.

The screen saver suffered a slight case of code rot associated with the advent of MAPBITS.DLL. The code that loads the image bitmap in SUNCALC.C needed to be modified to load the bitmap directly from a resource in the screen saver when built with ScreenSaver defined, as opposed to accessing the DLL when built as Home Planet. In addition, SCRNSAVE.RC needed to be changed to find the Earth bitmap in its new location in the MAPBITS directory.

Made sure all the DLL's contained suitable DESCRIPTION items in the file header, and rebuilt all in production mode.

The View From window didn't release its bitmaps when manually closed with the Close button or from the system menu. Since the bitmaps were released when the main window terminated, no storage was lost but it caused the View From to incorrectly assume that it had a precalculated projected image if re-launched. Modified to release the bitmaps in the WM_DESTROY handler of SATVIEW.C, which also guarantees the storage is released as soon as possible, without waiting for the user to exit Home Planet.

If updates were stopped, the Moon didn't appear in the Sun/Moon information panel until something caused it to be repainted. This was because (*gasp!*) the initial paint of the dialogue was done in the WM_INITDIALOG handler rather than WM_PAINT. Fixed, after a great deal of twiddling with the use of Frames to align user-defined images in dialogues.

When displaying the sky map at times far in the past or future, precession could cause the constellation boundary endpoints to wrap around to negative right ascension. This doesn't cause a problem for any other object, but it messed up the logic that draws curves for the boundaries that run along parallels of latitude. In particular, observing the conjunction between Mars and Uranus from Jerusalem at 16:00 UTC 25 January -5 caused an Integer Divide by Zero, bringing down Home Planet. Fixed by adding a fixangle() to precessObject() in ASTRO.C to constrain the right ascension to 0 to 360 degrees.

16 November 1994

In the process of flailing around trying to fix the EndDialog error in the Telescope Options dialogue (see 12 November 1994 above), I accidentally deleted the processing of the WM_CLOSE message in the Telescope dialogue procedure. Windows, of course, provides no default handling for this message for modeless

dialogues and, if a modeless dialogue attempts to close itself without a handler for this message, intelligently hangs the system in an endless loop of WM_COMMAND and WM_CLOSE messages. I re-installed the WM_CLOSE handler and audited all the rest of the modeless dialogues to make sure they contained a handler. In the process, I changed several dialogues which cleared their dialogue handle in WM_CLOSE to more properly clear it in WM_DESTROY (SUNMOON.C, SUNDIAL.C, PLANETP.C).

When I built a production version with optimisation, the phase of the moon was wrong. On closer examination, it proved to be *very* wrong--incorrect by 58 orders of magnitude. I tracked this down to the rather simple code in phase() in ASTRO.C and discovered, in the grand tradition of optimisation bugs, that any attempt to print intermediate values made the bug go away. I repaired to the Microsoft Knowledge Base on the Developer Network CD and discovered a note to the effect that global register allocation (enabled by default by both the smallest code and fastest code options) "can cause errors in floating point intensive code". Indeed. I set the Release optimisation options to the following custom list:

Assume aliasing across function calls
Global-level common subexpression optimisation
Generate intrinsic functions
Loop optimisation
Favour fast code
Inline: Any Suitable Inline size: 4

I decided to turn off register optimisation entirely for all modules since if something as simple as phase() stumbled into the error, it could take years to find all the places other optimisation bugs might be lurking.

8 January 1995

Indeed...if the user selected the SAO star catalogue, stars dimmer than 6th magnitude would not be displayed even if the limiting magnitude in the telescope or horizon windows were set to a higher value. This turned out to be yet another optimisation-induced bug in SKY.C, where bad code was generated for walking a __huge pointer through the locked-in-memory star map resource. I uglied up the code to manually copy data byte-by-byte from the resource into a local structure. This persuaded the compiler to generate correct code (at least there). Should Microsoft ever deliver a compiler which correctly handles __huge pointers, #defining COMPILER_DOES_HUGE_POINTERS_CORRECTLY will restore the original, more efficient code. Note that both Quick C for Windows 1.0 and the "fast" version of the Visual C++ compiler (without optimisation) generate correct code for this case--only the so-called "optimising" compiler produces the error.

I changed the contact information in the About and Related software dialogues from netcom.com to fourmilab.ch.

I updated the contact information in the help file.

Pushed out the door as bug fix release 2.1.

Release 3 Development

15-20 October 1996

Made the first-cut of a port to 32-bit Windows, using Monkey C version 4.0 under Windows NT 4.0 beta 2. I changed all the ‘‘repacked’’ API calls and call-backs, mostly WM_COMMAND, scroll bar related stuff, and made hundreds of little changes all over the place to get rid of compile errors and warnings resulting

from gratuitous incompatibilities and changes in the compiler and library. I had to include BASE= statements in the various DLL .DEF files to keep the DLLs from having to be relocated, which causes screams from Monkey C about how terribly inefficient that is (look who's talking).

I ripped out all the code in SKY.C to access the star map resources through file I/O, since that facility has been removed on 32-bit Windows (which is supposed to allow access to large resources through paging anyway).

Scroll bars in the Telescope, Horizon, and Orrery needed casts to (short) rather than (int) so that negative values are treated as 16 bit signed integers.

Since application icons have been supplanted by the bozo bar, I ripped out all the icon-display code and simply write the date and time as the title of the bozo bar item if we're minimised.

Added code to use the GetTimeZoneInformation() API call to obtain the time zone names and offsets, eliminating the need for the TZ variable to be set. Ripped out all the code for TZ handling and warning of its absence. Since incredibly wordy time zone names are used, I eliminated "Home Planet" from the map window title to make more room for the time zone to blither away.

In a great leap of faith, I turned off the redefinition of tan() which was used to work around the bugs in Monkey C 1.5, in the hope the kode kiddies may have fixed it in 4.0.

Removed the user identification in the "About" dialogue box as inappropriate for free software. It will eliminate the need to enter such information during the installation process.

21 October 1996

Got the screen saver edition to build as a subproject. This involved primarily ripping out the internal password code present in a 16-bit screen saver and letting the system handle that part of the job.

The "frame" item used as a placeholder for the Moon image on the Sun and Moon information panel insisted on drawing a silly little border regardless of how I set its parameters. I replaced it with a "Bitmap" control which behaves more reasonably.

22 October 1996

Put an end to "Civil Julian Dates" throughout the program and documentation. All Julian dates, including those exported via DDE, now use the astronomical convention that the day changes at noon. Before, modified Julian dates were used in an attempt to reduce the number of questions about dates from drooling AOL morons, but only resulted in outrage from the knowledgeable. Well, the morons are going to be confused about something else anyway, so we might as well make the non-knuckle-walkers happy.

24 October 1996

Created a new StarLight subproject, which builds a starmaps.dll with only the Yale BSC. Fixed code in SKY.C to detect which star maps are present in the DLL and only allow selection of those actually available. If the user has somehow selected a star map which doesn't exist in the current DLL, the first star map in the DLL (Yale for the standard ones) is selected. This allows upgrading from the "Lite" version to the full version simply by replacing the DLL —before a new .EXE was required as well.

25 October 1996

Remade all the screen shots for the help text.

Integrated a fix to the code that shows moon phases surrounding the epoch, phasehunt() in ASTRO.C, which I found in the process of debugging the recent upgrade to Earth Viewer. Phasehunt() was using the mean phase to decide when the current lunation begins and ends rather than the true phase. This could result in a discrepancy in which the current time was outside the range of phases shown in the Sun and Moon information dialogue.

Removed the menu items, related code, and settings file entries for the iconic display mode. Since we don't show an image when minimised, this is no longer necessary.

Save Settings wasn't working because the resource (even though marked "PURE") was cached in memory under 32-bit Windows and the writes done to delimit end of strings ended up corrupting subsequent accesses to the resource. To avoid further pointy-headed behaviour, I changed the code in SETTINGS.C to always copy from the resource to a local buffer and make the modifications there. I included code to detect an overflow the local buffer used to copy the individual lines from the resource.

The new "grey flannel" defaults for dialogue box layout left extra space in several of the panels. I re-sized them to better fit the new (for the millisecond) "standard".

Bob save us from anything Mangysloth call "smart". Now it's sorting of items in list boxes, which the kiddies have decided to "enhance" by ignoring ANSI collating sequence and the locale in favour of ad-hoc futzing based on their own Seattle-centric idea of words. This caused the "-none-" item, deliberately intended to sort at the start of the Track Satellite dialogue, to wind up at the head of the "n" entries. I turned off sorting; if the elements aren't sorted to start with, mail them to Billy Boy to sort—he's the one that blew away the capability in the control.

The EARTICON optimisation in the resource file and SUNCALC.C, intended to avoid regenerating the minimised icon view from the vector database, is no longer needed since 32-bit don't need no steenkin' icons. Eliminated.

27 October 1996

Captain, Captain, the stars have gone out! Well, actually, the code that allows automatic switching between the "Lite" and full STARMAPS.DLL in SKY.C was clobbering starMapHandle when initialising the Display menu. This caused the stars to disappear in all subsequent repaints until the user explicitly re-selected a star map. Fixed.

What with the introduction of "full drag", dragging the size of the Sky window became painfully slow because innumerable repaints would be done. I modified SKY.C to use the new WM_ENTERSIZEMOVE and WM_EXITSIZEMOVE to detect when paint requests are being generated due to dragging and paint only the outline of the sky, not the contents.

I turned off the code which processed the WM_WINDOWPOSCHANGING message in SKY.C which attempted to guarantee a square client area which avoided wasting any real estate for the Sky map. Billy Boy has so thoroughly screwed up client to window transformation and drag behaviour that attempting to override drag requests causes screwball misbehaviour all over the place during dragging. From now on, I'll just let the user worry about wasted space in the window—he was the one that bought Windows in the first place.

After hours of flailing around, I think I have undone the damage "full drag" did to

the Telescope and Horizon windows in SKY.C. The interpret WM_SIZE in real time, but do not paint objects in the panel until the end of the drag operation. Naturally, Windows sends the WM_EXITSIZEMOVE out of order to a dialogue, so an explicit invalidate and repaint is required at the end of the resize.

The Orrery and Object Catalogue windows could be resized so that the entire window was the size of the original client area. This could (and did, in the case of the Object Catalogue) lead to controls within the window overlapping. I modified the initialisation code in ORRERY.C and OBJECT.C to use the initial dialogue window size, not its client area, as the minimum drag size.

Added a gratuitous repaint in ORRERY.C when EXITSIZEMOVE indicates the end of a resize drag. Some Windows bug or other causes the relocated buttons not to repaint unless you do this. I didn't observe the problem in OBJCAT.C, but I added the code just in case it breaks on some future pointy-headed "update".

Naturally, the "View Earth From" panel in SATVIEW.C needed the same changes as the Sky and Telescope to immunise it from the depredations of "full drag". Note the telltale marks of "strategic incompatibility" here—change something in the underlying system which drastically modifies the behaviour of all applications, them make the change the default and provide no way for an application to disable it application-wide.

28 October 1996

I created a new map_lite project which builds a STARMAPS.DLL with only the small topographic map. This further reduces the size of the "Lite" version of Home Planet.

31 October 1996

The screen saver lacked the title string (first item in string table), so it displayed an erroneous name in the Control Panel display applet. I shuffled the string tags so that "Home Planet" now appears as ID 1 in the string table.

When the screen saver is displayed by the control panel applet, the start-up screen is suppressed to as to give a better idea of what the user will see. I also suppressed display of the Moon and the time and date if fChildPreview is set, since they don't scale appropriately to the applet window and look odd.

4 April 1997

Modified the IAUC to CSV comet orbital element converter, TOOLS/COMETEL.C, to accept the new form of elements in which a periodic comet is flagged solely by the presence of an eccentricity in the elements.

Modified both TOOLS/COMETEL.C and TOOLS/ASTRELEM.C to open their input file in binary mode. Why? Because if they're processing a Unix mail folder, they may encounter a ^Z embedded in the file and, treating it as a DOS end of file marker, skip the rest of the file. This is conditional on being built for a non Unix system.

Release 3.0.

2000 January 3

The Sun and Moon display in SUNMOON.C displayed the local time for years after 1999 as 19xxx, where xxx is the number of years after 1900. I modified it to display the correct year for all cases.

The title in the main window abbreviated the year to two digits, resulting in a potentially confusing single digit year for years 2000-2009. I changed it to

display the full four digit year. The abbreviation was done in the interest of fitting on the screen with mid-1990s displays. With the higher resolution displays now prevalent, there's no reason not to display the full year number.

Microsoft morons broke compatibility with help file language designators, resulting in some people reporting that the help file could not be displayed when they "upgraded" to Windows 98. I rebuilt the main help file and the development log with the language reset to U.S. English in the hope that all versions of Windows will now be able to display it.

Changed version in About dialogue and VS_VERSION_INFO to 3.1. Changed language block header in VS_VERSION_INFO to U.S. English from UK English in an attempt to dodge future Microsoft language torpedoes.

Updated information on where to obtain current satellite orbital elements in the help file.

Deleted obsolete CODE and DATA definitions in .DEF files for all modules to avoid warning messages.

Deleted obsolete _cdecl declarations in SCRNSAVE.C to eliminate "anachronism" warnings from Monkey C 5.0.

Installed current orbital elements for the sample SATELITE.SAT and TVRO.SAT databases.

Added -none- entries as the first line in all asteroid and comet databases. This makes it easier for a user to cease tracking an object regardless of the database they're using.

Even on a machine with hundreds of megabytes of RAM, pointy-headed Windows 98 runs out of one of its idiot resource pools when trying to stretch a large DIB into a large full-colour or high-colour screen. I modified the code which paints the screen in SUNCALC.C to detect this failure and attempt to paint the screen in successively narrower vertical bands in order to work around the problem. This fixes the "stripey" display of the large NOAA topographic map in large full-colour windows.

Release 3.1.