# CT Vector

## Version 0.43 beta

### Introduction

In 1977 Game Designer's Workshop published the science fiction role playing game known as Traveller. As a young role player and inveterate fan of science fiction, it was only natural that I adopted the system of little black books (LBBs) as a favorite system for role playing adventures in deep space. The original game was laid out in three books. The first (LBB1) detailing player character development and combat mechanics, the second (LBB2) concerned itself with ship creation, mechanics and combat, and the third (LBB3) detailed worlds, systems, trade, etc. It is primarily the ship combat system detailed in LBB2 that this application was created to facilitate. Traveller presented a system of vector movement, plotted on paper, to represent the interactions of ships maneuvering in an interplanetary environment.

While I was intrigued by this approach for its realistic treatment of the space environment, it quickly became apparent that it was difficult to employ with ease due to the time involved in plotting vectors and the restraints of scale involved. Scale being 100kms per mm on paper, to draw Sol would still require a sheet of paper over 9 meters across! It also had an unfortunate tendency to change a roleplaying game into a ponderously slow wargame, which many gamers would have had little patience with. I ended up fudging the space combat stuff in my games. Future editions of Traveller largely abandoned this combat system, likely for the same reasons, but I always preferred the idea of the old system's approach in spite of its drawbacks.

The following years saw the advent of home computing, and it seemed a natural thing that this system could be adapted to a computer that could accept the input of navigation data and display the turn based results of that input in a more user friendly manner. Even so it was more than a decade before personal computing power reached the point where I deigned to make an abortive attempt at such a tool. My first attempts failed due to my own limitations in mathematics and programming, but even given those limitations, it was the inability of my chosen programming language (Qbasic) to handle the range of numbers generated that doomed that attempt. I'm not presently, nor was I then, of an inclination to learn entirely new programming languages. The project got shelved indefinitely.

Modern computing, and the face lift given to Qbasic by the folks at QB64.ORG, have now made it possible for me to advance the project into something that might actually be useful. CT Vector is the result, CT being a reference to "Classic Traveller".

### Just what do we have here?

CT Vector brings a 3D cartesian coordinate system model of a local star system to your PC desktop. The system's primary star being the origin at (0, 0, 0). Coordinate resolution is 1 km and has a range that can easily accommodate large star systems, even far companion systems. Ships may be placed anywhere within the loaded star system (*or even far outside of it*) and maneuvered through it. Any ship on the board can be designated as the "active" unit, and given navigational parameters which are then executed when committed to a "game turn". All other ships will display bearing and distance from the "active" unit, unless sensors are blocked by planetary bodies or extreme distance. Target locks may be established between combatants and/or lost from distance and planetary occlusions.

A number of visual aids are included, among them are dynamic zooming, jump diameter shading, range bands, 3D tilt, orientation and directional displays, scale grids and orbital tracks. These may be toggled on and off as needed to reduce display clutter, and/or speed program execution.

The star system details are dynamic, with planets and satellites moving with each turn. Since I am not an astrophysicist by trade, the celestial bodies themselves do not move according to gravity; rather moving by rote data, but they will exert gravitational influence upon the ships in play as well as presenting potential dangers from impact. There is no orbital eccentricity and all planets follow the same ecliptic plane, but the ships themselves are not limited to that plane. You may take your battles to 3 dimensions if you wish. Because there are no gravitational mechanics for the planets, there are also no provisions for barycentric phenomena, though it may be possible for a clever GM to set up a system data file that would mimic something like that, as I have done with the Pluto/Charon complex in the default Sol system included.

Star systems, when loaded, will query for a date reference which will determine the relative orbital positions of planetary bodies on that date. If player characters leave the system and return at some point in the future, the planet and satellite positions will have changed accordingly, adding a subtle layer of realism to game play...or so it is hoped...

CT Vector does not handle combat details other than range, position data and presence or absence of target locks and transponder signals. It is felt that such things as the firing of weapons and details of damage control are best left to the players, their character skills and their dice rolls. It is not envisioned as a game in itself, but rather as a game aid. It is designed primarily to get rid of paper, compasses and protractors as well as provide some additional visual feedback to given situations. How far is it to the jump point and how fast is that pirate vessel closing with you? Now there's a nav computer screen to show you.

## Basic work flow and file system

**System loading:** Appropriate star system files for the campaign/scenario should be prepared by the GM ahead of time and these should be stored in the "systems" subdirectory. The user may specify additional subdirectories within  the "systems" folder, such as sector/subsector directories, but the user will be responsible for providing the path to those directories. It will be possible to enter the path and filename at the initial splash screen, or the program will default to Sol, which is provided for practice purposes. After entering the main program it will still be possible to load other systems, though only one star system can be loaded during a session. Star system files are identified their "**.tss**" extension.

**Ship saving/loading:** The program starts with a predetermined group of ships, which can be added to, edited or deleted as necessary. Once the desired details are entered/edited, the vessel group can be saved for future use, whereupon the group is saved in the "ships" subdirectory under a desired name with a "**.tvg**" file extension.

**Scenario saving/loading:** Specific scenarios can be saved, and are placed by default in the "scenarios" subdirectory and recalled from there later. System, ship and other data are saved under a common scenario name in that directory and are not to be confused with system and ship files in the "system" and "ships" subdirectories as this <name>**.tfs** scenario file saves the exact scenario position and turn data for picking up where the players left off. CT Vector will also auto save to provide some crash protection. Use "auto" as the scenario name to recover from a program or computer crash.

**Navigation:** Each participant vessel requiring movement for the turn is chosen as active, whereupon it's azimuth heading, angle of inclination and thrust can be entered. These inputs are not executed until a game turn is committed. Once a turn is initiated the ships move according to these navigation orders and any pre-existing vectors. The ships will continue to execute the same orders on subsequent turns unless they are altered. During this phase it is possible to add, edit, move and delete ships.

**Turn:** All ships move according to their navigation orders, all planets and satellites move according to their orbital periods, turn and time counters are updated. Display data is updated to reflect the new positions, and certain old data is stored for vector computations or to undo a mistaken turn. Only one turn can be undone, there is not sufficient data storage for more than one undo operation.

## Shall we begin? The splash screen.

When opening the program we are greeted by the title screen and queried to provide a system/scenario file. If the user knows the name and path of a specific file, it can be entered here, or pressing <enter> will default to a demo of the Sol system. After this, the program queries for a year and day input, followed by choosing between 2D and 3D mode.


All system files are stored in the "systems" directory with initial "zero year" planetary positions. Date input (year & day), where dates are tracked in game campaigns, insures that the planetary 'ephemeris' is corrected from one visitation to another. Return to the system a few months later and the planets will have moved according to their orbital periods. This is not to be confused with a saved scenario, which is solely for picking up a game where it left off. Rather it is designed to give a degree of continuity to the game universe as the GM develops it. It may be considered a minor thing and safely ignored, if not needed, by simply pressing <enter>, when queried for the dates, to bypass the positional computations. The system will still function from its zero year positions normally. Following the date input, a query for 2D/3D choice is displayed. Entering "2" at this point will place the program in 2D mode, disabling 3D tools and putting all units on the ecliptic plane. While some gamers may welcome the additional realism of 3D, others may find it an undue complication to game play. The option for 2D is provided, however the 3D is the default.

## Main Astrogator Screen

Once we are through the splash screen phase, the main display appears. This is where we will live from now on and consists of several elements, a ship data display, orientation display, planetary distances and directions, controls block, Z-panning bar and an "**astrogator**" display. The astrogator, on the right half of the screen, is the visual centerpiece of the program and displays the relative positions of the units in play and the planets and satellites moving nearby. The active unit is displayed at the center of the astrogator and changing the active unit updates all other positions relative to the new active unit. The display may be zoomed in or out to include as little or as much of the star system displayed as desired. Except in cases where ships are widely separated, a default zoom factor of "1" is whatever field is necessary to include all units in play, making the maximum use of the astrogator display.


**Zooms:** Displayed below the lower left corner of the astrogator are the zoom controls and the present zoom factor. The user may either left click on the blue button fields to actuate the desired zooms or use the hotkeys "+", "**-**", or "**x**". The Zoom Extents function "**x**" always returns to zoom factor 1. All planetary details will resize to the new scale when zooms are used. Labels will be dropped as details crowd each other, with the exception of the active unit label, which is always visible.

**Mouse zooms:** When the mouse cursor is in the Astrogator display, fast zooms can be accomplished via the mouse wheel. Rolling the wheel forward zooms in, rolling backward zooms out and clicking the mouse wheel button reverts to zoom factor 1. Display button effects are bypassed to speed the response of mouse zoom.

## Display toggles


Just to the right of the zoom controls are several display option controls. It will be noted that some are "lit" while others appear dark. A left mouse button click on these controls will toggle their state on/off. The button labels also display their associated hotkeys in red which will also toggle their state. Some are off by default, others are on, their state is also saved when saving a scenario file. The inclinometer button "Inc" will only be visible, in any state, in 3D mode.

**Range:** The range toggle displays combat range information in the form of shaded ranging bands and circular targeting limits. It is toggled on and off via this button or hotkey "**r**". These are based upon the LBB2 rules for combat ranges and sensor detection. This display is off by default to reduce initial screen clutter. The range values can be altered in the "Options" button dialog box, if house rules differ.

**Orbit:** The orbit toggle is an on/off toggle that displays or hides the orbit tracks of planets and satellites that are present in the astrogator screen. The orbiting body need not be within the astrogator screen so long as it's orbit track is. The hotkey for this toggle is "**o**". This display is on by default. Because of issues with rendering large circular paths, orbit tracks will render as dashed lines on tight zoom ins. This is normal and indicates linear representation of the particular track, and not a true arc.

**Grid:** This, along with hotkey "**g**", toggles the scale grid on and off. When present the grid dynamically resizes, and also rescales by a factor of 10, according to the zoom factor and display limits. When the grid is on, the size of it's smallest block is displayed in the lower left corner of the astrogator. This display is on by default. The grid's relative origin in the display is always centered upon the active unit. Like all other screen elements it is an orthographic rather than a perspective projection and is always parallel to the chosen rotational plane. It does not give foreground/background information, which must be obtained from the data. It does, however, give relative display scale at a glance.

**Azi:** The azimuth display gives 360° "longitude" directions relative to the ecliptic plane. Toggled by hotkey "**a**", it is oriented from 0° at Coreward, {*a Traveller canon direction indicating the galactic core*}, and ascends clockwise through Trailing (90°), Rimward (180°), Spinward (270°) and back to Coreward. When present, it also displays a small yellow circle which indicates the direction to the system's primary star. In addition, it displays the direction of travel for the active unit by a faint line extending from the active unit in the center to a purple arrow situated upon the azimuth scale. The arrow indicator is a measure of "Heading" of the active unit and does not indicate thrust "Yaw", which may differ, often significantly and can be seen in the orientation screen. When rotating out of the overhead view in 3D mode, the Azimuth wheel will "squish" until the rotation exceeds 85°, at which point it will disappear, even while enabled, to reduce screen clutter. It has ceased to be visually useful before that anyway. This display is on by default.

**Inc:** The inclinometer display, toggled by "**i**", gives a 90° "latitude" display divergent from the ecliptic plane. Its primary use being in showing the inclination from the ecliptic of the active unit. The inclinometer, if enabled, will begin to be visible when rotating the display beyond 5° from the overhead view. Very much like the Azimuth display, a directional indicator will be visible, showing the inclination of travel, but also like the azimuth indicator, may not reflect the actual thrust "pitch" of the active unit. This display is off by default to reduce display clutter, and would only be used if 3D operation is desired. Indeed, it is disabled when the program is run in 2D mode.

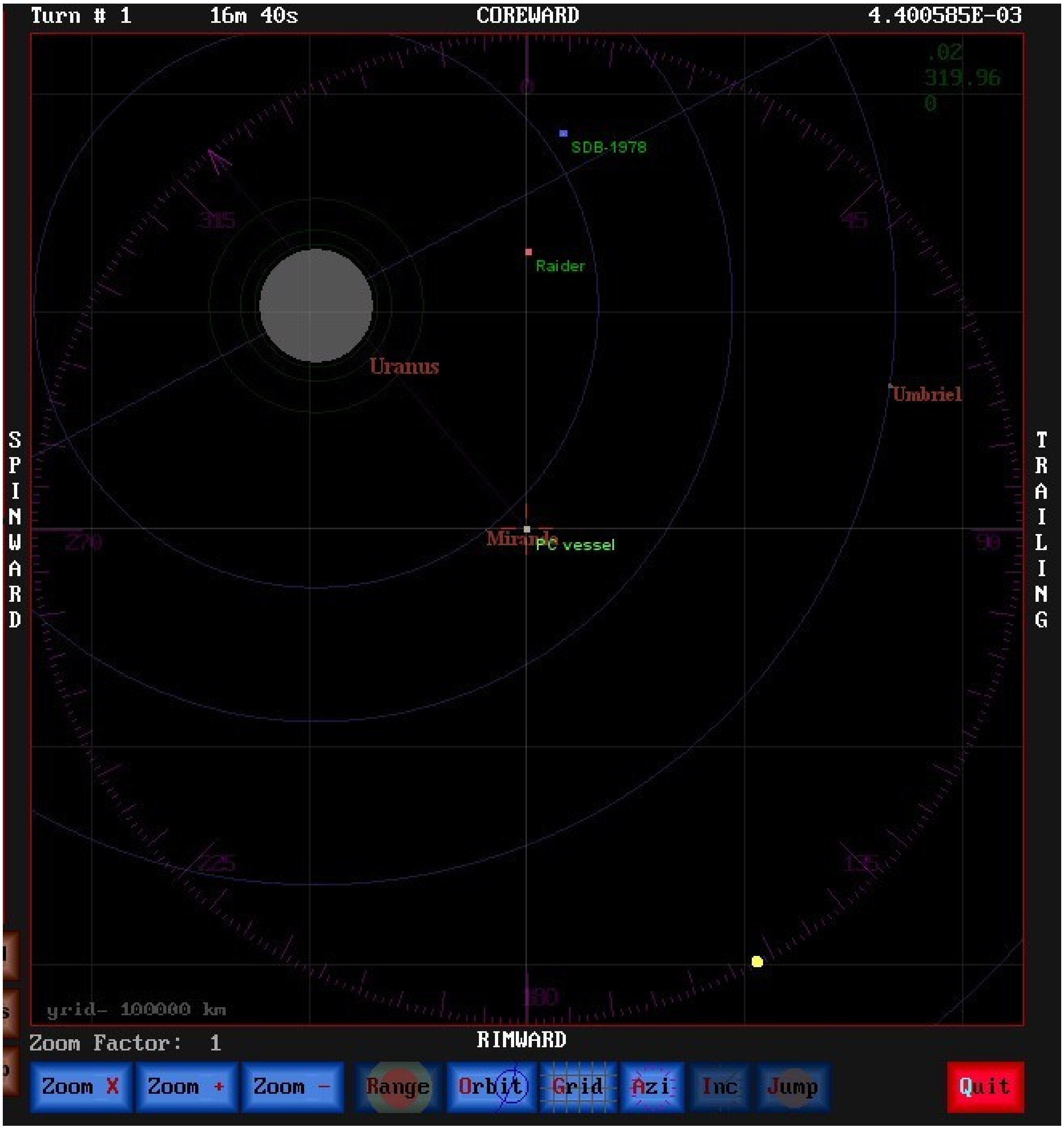
**Jump:** The jump shader is toggled by "**j**", and displays shaded spheres around gravitationally massive bodies at 10 and 100 diameters, as per Traveller canon. This gives a visual reference for when a ship is at a sufficient distance from the body to safely enter jump space. When enabled, it displays an additional toggle adjacent to it, which allows the GM to toggle; via left mouse or hotkey "**d**", between jump zones based upon planetary diameters or planetary densities, which can alter the size of the zones for planets and stars of different density than the standard "Earth" density of 1. Use of this would be dependent upon house rules to that effect and can be ignored if desired. Under high zoom factors, where a jump zone would normally fill the screen, such zones are not displayed, while smaller zones will still be displayed. Another mechanism to save on screen clutter. This display is off by default.

**Quit:** This ends program execution, saving a final autosave before exit.

## Elements of the Astrogator display

Below, we see an example of the main astrogator display, in its default toggle state. On the upper left it is indicated that this is the result of executing turn #1 and that 16 minutes, 40 seconds have elapsed, the same length of time as Classic Traveller's 1000 second turn. On the upper right, this version presently displays a scaling factor, which is a debugging tool that normally would run behind the scenes. Future versions may dispense with, or replace this with something useful.

The cardinal points 'coreward', 'rimward', 'spinward' and 'trailing' are displayed around the edges of the screen. These are stationary galactic orienting references which generally do not change except to indicate 'zenith' and 'nadir' orientations when a 3D tilt is enabled. Finally, we see at the bottom left, the present zoom factor located between the screen and the aforementioned display toggles.



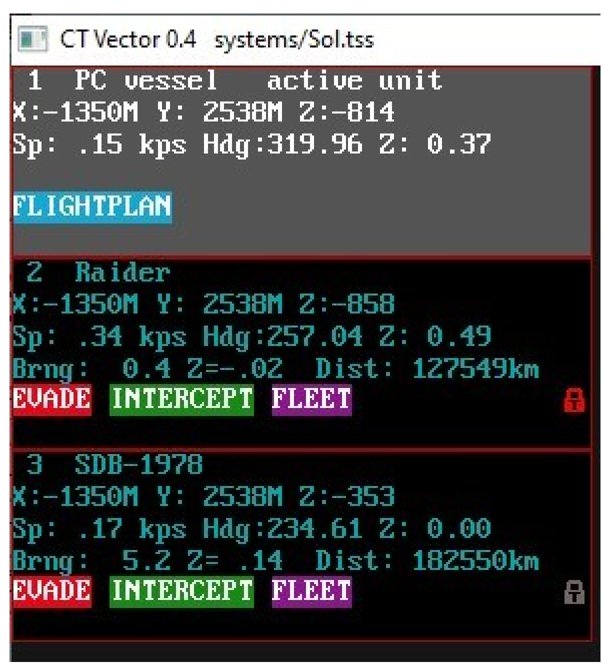
Referring to the example on the previous page, we see that “PC Vessel” is presently the active unit, in bright green, located in the center of the screen and the faint line and arrow through Uranus (*no snickering please*) out to the azimuth scale reveals its heading to be approximately 320°. Other units, the “raider” and “SDB-1978”, are displayed in darker green, located approximately 130,000 and 180,000 kms coreward, respectively. Additional information on all units is displayed in the data fields on the left side of the program display.

We see that the grid scale is presently 100,000kms per square at zoom factor 1. We know that the primary star is at approximately 152° as indicated by the yellow circle on the azimuth scale. We can also see the primary gravitational influence vector in the faint green numbers at the upper right, which display Gs, azimuth and inclination, respectively from top to bottom. This reveals the gravitational influence of nearby bodies being exerted upon the active unit, and explains the present heading of that unit, which has not yet applied thrust against the pull of the gas giant. If left to drift, the unit would eventually be pulled into the radius of Uranus and crash, assuming some other fate did not befall it first. Noting how close the active unit is to the moon Miranda, the possibility exists of a crash there although the pull of small moons is notoriously weak compared to their parent planets.

As zoom settings are changed the grid resizes to accommodate the new settings. If the grid becomes excessively large it will subdivide by a factor of 10 and a new scale value appears on the lower left. If zooming out, the opposite effect will occur. Every tenth grid line is a heavier line than the scale sized grid lines. Grid scales beyond 100,000,000 km will display in astronomical units.

All planets and satellites are displayed in grey, their orbit tracks shown in blue. Stars are displayed in a scale of colors depending upon their general spectral class. The presence of ring and belt systems is also displayed by default which may be toggled on and off in “**Options**”, or by the “**b**” hotkey. Belts and rings, like orbit tracks, can slow 3D rendering, so turning them off when not needed helps with rendering speed. Belts and rings also share a similarity with jump zones in that they are not displayed when they occupy the totality of the display limits. Instead, a watermark label is displayed to indicate their presence.

Planets of sufficient size and density will also display a series of faint green circles around them. These are gravity zones, graduated in .25G steps. While the planet’s gravitational influence is not limited to these zones, they give a visual idea of the relative strength of the planet’s attraction to nearby objects. Depending upon the active unit’s thrust potential, some of these zones will display brighter. This indicates zones where the active units drives cannot overcome the planets pull without additional velocity to aid it. A weak unit at standstill relative to the planet will not be able to escape from these zones. The hotkey “**z**”, or “**Options**” controls, will toggle the grav zone display.




## Ship Data Fields

Meanwhile, over on the upper left of the screen we have the ship data display. The active unit in this display is labeled so after the name, as well as being highlighted as white text on a grey background. All other units are shown as blue text on black.

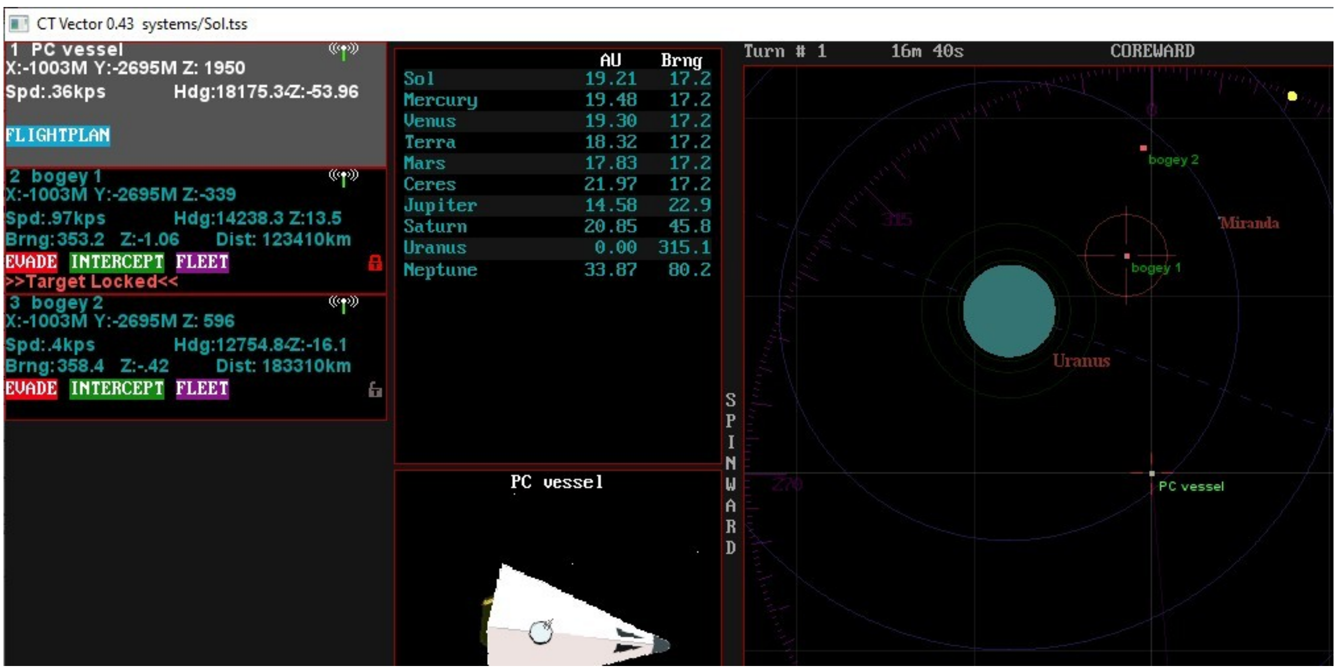
All units display a unit number, name, absolute positions in X,Y & Z, speed, heading and inclination angle. Non-active units, unless occluded, also display a bearing, inclination and distance from the active unit.

**Target Lock:** The small padlock "T" icon displays in either red or grey; red indicates that a unit is within target lock range, grey indicates that the range is too far to establish a target lock. The target lock range is dependent upon whether the ship is equipped with civilian (½ *light second*) or military (2 *light seconds*) grade sensors.

The GM may elect to click on the target lock if player(s) roll for, and establish a lock, whereupon the subject of the lock will display "target locked" in it's data field and the unit point in the astrogator will be haloed by a red reticle. We see here that "Bogey 1" is within target lock range, "Bogey 2" is too distant yet.

If the lock icon is clicked in the Bogey 1's box "Target Locked" will show at the bottom of the it's display box, the icon's shackle will close, and a reticle will appear on the astrogator display, as seen below. If the active unit is later moved to the Bogey 1, the message in its block will change to "Targeted by: 1," a reference to the unit targeting it. A dashed "target laser" between the two units will also appear in the Astrogator, indicating that the active unit is being targeted by another.

Once a lock is established, there are three ways it may be broken. One is for the distance between the vessels to exceed 3 light seconds (900,000kms), or for the sensors to be occluded by a planetary body. It may also be terminated by clicking on the lock icon, whereupon the shackle will open.




**Transponder:** The transponder icon on the upper right is on by default. It indicates the presence of a transponder signal for that unit. When a unit is active, it may be toggled on/off with the "**Trnspndr**" button or all units toggled at their icons. It transmits unit identity to the active vessel. When a unit's transponder is off, the unit's ID is not visible to the active unit. It does not mask a unit's presence, but indicates military or clandestine intent. If a contact's transponder is off, target lock ranges to that contact are halved. The contact's data display will reveal its data only when within that range.

**Flightplan/Evade/Intercept/Fleet/Cancel/Break:** As of this writing, these routines are still in development. I hope to have properly functioning algorithms in future releases. The idea of these controls is to introduce automated maneuvers and situational editing to the active unit vessel. Accessing these controls is done by left clicking the particular field in the data display box for the unit that these maneuvers concerns.

[**FLIGHTPLAN**] Opens a dialog box which presents the user with a number of flight choices.

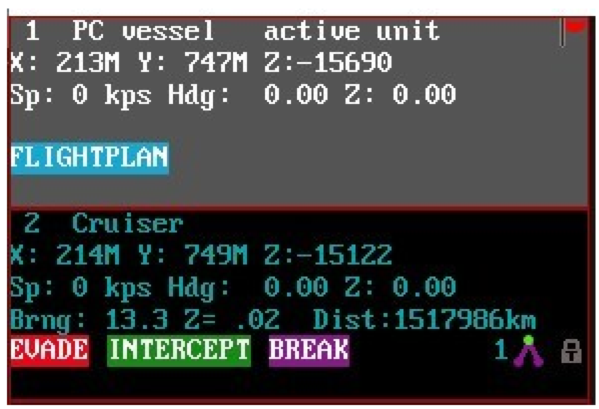
- Jump in Matching: will match the active unit's vector to the nearest rank 2 planetary body, useful to set a unit's vector at near zero with respect to a nearby major body. This is more of an edit than a flight command, and is envisioned to emulate the vector matching of a properly executed jump space exit.
- Planet Fall: directs unit to proceed to and land upon a chosen planetary body. Units that have landed are not able to execute flight missions, and must launch first.
- Orbit: in development, will establish parking orbits
- Hold Station: Causes a unit to maintain a position relative to the system primary, a nearby planet, or a nearby satellite.
- Safe Jump: Causes a unit to move to the nearest safe jump point.
- Abort: Exits the Flightplan submenu.

[**INTERCEPT**] will set the active unit to seek to close to target acquisition range with the unit for which the "intercept" button was clicked and then to match the target's vector on arrival, maintaining the unit within optimum combat range. This is still in development and operation is sometimes erratic.

[**EVADE**] will set the active unit to seek to avoid closing range with the unit for which the "evade" button was clicked.

[**CANCEL**] appears in any data box for which an automated intercept/evade order is issued. When clicked any such order is canceled and ship control is returned to the user.

[**FLEET**] when clicked, will slave the particular unit's maneuver orders to the active unit, causing it to move with the active unit, mimicking its maneuvers. A number and 'slaved' icon will appear in the unit's data box and a flag will appear in the active unit's box. The number indicates the 'flagship' from which the unit receives its orders. We see at left, that the cruiser has been placed in a fleet formation with PC vessel aka #1. A flagship can have any number of fleet subordinates and can have other flagships slaved to it, which can then be arranged into squadrons if desired. Any ship in a fleet will duplicate, to the best of its ability and circumstances, any thrust vectors that the flag ship executes. Care should be taken that a flag ship not run its subordinates into an obstacle.



*Display showing fleet icons*

[**BREAK**] will cause a fleet slaved unit to break formation with its flagship. This will not break the formations of any fleet subordinates that it may have. Each unit must break formation individually.

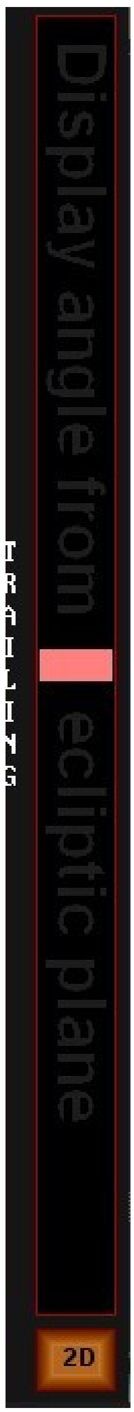
**Choosing the active unit:** There are several methods of choosing the active unit. The up and down arrow hotkeys will scroll through the data display, moving the active unit as it goes. Alternately, the user can left click on the astrogator near the desired unit, the non-active unit closest to the click point will be chosen, or the desired unit data field can be clicked anywhere except on the target lock icon, transponder icon or the "evade", "intercept", "fleet" or "cancel" buttons.

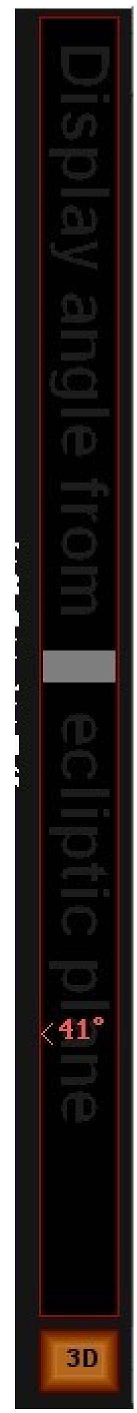
**Viewing unit details:** Right clicking in a unit data box will reveal a dialog giving more detailed information on the unit.

## Operations in 3 dimensions

Even as space itself is an immense void with width, height and depth, Traveller was always envisioned as a 2 dimensional environment in order to simplify game mechanics. Both interstellar and interplanetary settings are represented as 2 dimensional phenomena within the game. In that spirit, CT Vector defaults to a view that is set perpendicular to the ecliptic (z=0, XY) plane upon which all planets move, as though one were looking down on a sheet of paper on the gaming table. It displays the azimuth scale by default, but not the inclinometer scale. Nav command input requires heading and thrust, but inclination is easily bypassed, whereupon, all units will continue to move on the ecliptic plane. Gamemasters and players can easily ignore the presence of the 3D model.

However, the ease with which a third axis can be defined in a computer makes the capability a no brainer. The trick then becomes to represent a 3D environment on a flat screen. While there are numerous ways to do this, some of them quite complex, I have opted for a simple, single axis rotation scheme, which makes it possible visualize relative "Z" positions and spherical radii phenomena. It is here that we introduce the Z Panner shown on the right.




Similar to any scroll bar the Z Panner permits mouse click/drag selection of alternate views rotated around and gimbal locked to the relative X axis, as defined by the presently active unit and always remains horizontal. Other vessels and celestial bodies are rotated about that axis. The panner has a range of 180°, with 0° being parallel to the ecliptic plane, facing coreward; to the default 90° perpendicular to the ecliptic plane; to an inverted 180°, parallel to the ecliptic and facing rimward.

Once an alternate view has been set in the Z Panner bar, a degree value is displayed in the bar and the bottom button will display "3D", indicating that a 3 dimensional perspective is now active. No data is shown on the bar when the display is in 2D mode, as shown at left, and the button will display "2D". When a rotation is made, the angle is displayed in the bar and the "3D" button is displayed as shown at right.

This bottom button will also toggle between 2D and 3D view via a left mouseclick or the user may toggle modes with hotkey "**3**". Toggling in this manner will alternate between the default overhead view and the last angle from the ecliptic plane that was set on the Z Panner.

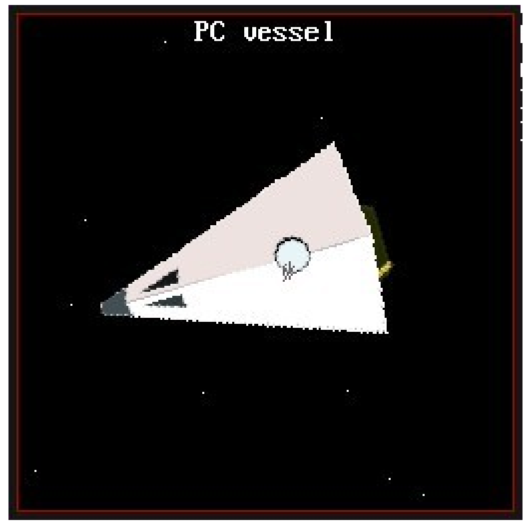
It should be noted here that there is a difference between 2D/3D mode and a 2D/3D view. Setting 2D mode disables all 3D functionality while setting a 2D view in 3D mode simply selects an overhead view of 3D functions.

The center block will be lit in overhead view and grey when in 3D view, clicking on the block defaults to overhead 2D view. The lower half of the bar orients the view between 0° - 89°, with the top of the screen trending toward galactic 'zenith', while the upper half orients from 91°- 180°, 'standing on one's head', with the top of the screen trending toward galactic 'nadir'.

The 91°- 180° angle display is visually indistinguishable from 0° - 89° angles in most circumstances, so when rotations exceed 45° from 2D view the upper and lower galactic directions will change to indicate facing directions. Additionally, a small line is displayed over planet names.

**Ship placement in 3D mode:** It should be noted that when a 3D viewpoint is selected any right mouse click placement of the active unit will occur relative to the newly defined view plane and not the ecliptic. This means that placing a ship in 2D mode will not effect its Z coordinate, but in 3D mode the vessel will be relocated with both its Y and Z coordinates set according to the definition of the viewing plane and not the ecliptic plane. If a ship is placed while viewing at near 0° or 180°, then more emphasis will go to altering the unit's Y & Z coordinates, when moved while viewing at near 90°, the ship will be placed with more emphasis on its X &Y coordinates.

Active unit coordinates may also be adjusted via the EditShp button (hotkey "**e**") if necessary to fine tune a position. This will be explained in a later section.



*Orientation display, showing no thrust*

## Orientation display

The orientation display gives the user a general visual cue of what the active ship is doing. A graphic representation, modeled upon the ubiquitous Sulieman class scout/courier, rotates to give the general yaw of the vessel, motion of background stars gives general heading, and the presence or absence of a stern plume indicate powered flight or coasting flight respectively. The active vessel name is displayed at the top and the bottom is reserved for any other messages or info.

## Planetary Distance Display

Above the orientation display at top center is the planetary distance display which gives distance and heading to all the system's rank 1 & 2 celestial bodies. Left click in this display initiates a planetary information dialog box where the user can view a particular bodies data.

## Controls Block


At the lower left of the program screen is the controls block. The bulk of the action is initiated from here. Some of the control buttons are instantaneous controls, where others are mode toggles. A few, such as the flight controls on the far left are available provisionally. When a button's light goes off, it is either simply off, or not available under the circumstances. Some of the control buttons will display a red uppercase letter in their label, this indicates their corresponding hotkey, if one is available. An explanation of the various controls follows:

**Vector**: The vector control and hotkey "**v**" provide the primary navigational input. There are two possibilities here, left click on the control and thrust direction and inclination can be entered graphically on the astrogator, or pressing the hotkey allows for keyboard numerical input. The former method will be explained in more detail in the next section. The later allows for a more mathematically precise helm control by entering values via text input. This is unavailable when the active unit is either disabled or part of a fleet formation.

**Gs= 0**: This is essentially an all stop command. The ship continues to coast at the speed that it has attained, subject to any gravity perturbations, however it is no longer actively thrusting. Unavailable when disabled, or in fleet formation.

**Brake**: Velocities can grow quite high in the vacuum of space. The brake command gives a quick method for applying a braking thrust against a present coasting vector. This is also unavailable when the ship is disabled, or in fleet formation.

**Options**: Changes gunnery ranges and display toggles otherwise inaccessible. Additional functionality may come in future versions.

**Details**: Displays a more detailed dialog box of the active ship's status. Precise coordinates, flight status, yaw, pitch, heading and speed are displayed. Targeted, and any targeting, units are listed.

**Col/Chk**: Toggles collision checking on & off. Collision checking is enabled when this is lit. When off, the program ignores the presence of massive bodies, ships will fly through planets unharmed, although those planets will still exert gravity and occlude sensor operations.

**Turn**: After all navigation orders have been finalized for the ships in play, this button executes those orders, including any automated orders, committing all units to their new vectors, coordinates and incrementing the turn counter.

**Undo**: Unavailable until a turn has been executed, this control will roll back the actions of a previous turn. Since only previous turn data is saved, it will only undo one turn.

**Help**: Displays a dialog box listing controls and hotkeys. Still under construction.

**EditShp**: Opens a dialog to edit the active ship's information and capabilities. At each data point the user can change the existing data by entering new data, or simply press [enter] to default to the already existing value.

**AddShp**: Introduces a new unit located 100,000kms coreward & trailing of the active unit and after accepting a max thrust value, makes it the active unit from which additional editing and placement can be done.

**MovAll**: Normally off, this control allows for right mouse click editing of the positions of either all units at once or singly. When MovAll is lit, placing the active unit will move all other units to the new position relative to it. Primarily a GM edit/scenario setup tool.

**Delete**: Opens a dialog box asking for confirmation that the user wishes to delete the active unit from the session. The user may confirm this or abort the operation. Once confirmed, the unit and its data is gone. Be careful here, as this cannot be undone.

**Purge**: Units that have crashed on planetary bodies are shown as having done so. A ghost spot where this occurred remains on the screen. The Purge command removes these ghosts and their associated data from the session, condensing the ship data display. This cannot be undone.

**Adrift / Repair**: This toggle alternates between "Adrift" & "Repair". When a ship takes battle damage that disables its drives, the Adrift button can be pressed. At which point the ship is incapable of thrust until it has been repaired, an opportunity for the ship's engineer to show his qualities. While adrift the vessel name is display in red on the Astrogator and is subject to any and all gravity perturbations and may crash on a celestial body. When this vessel is active all gravity zones will appear brightly lit if those zones are enabled. If the vessel has been deemed destroyed then its point on the Astrogator can be considered its wreckage and debris field, coasting through space.

**LoadAll**: Opens a dialog querying for a scenario name, whereupon all present data is deleted and the new scenario data is loaded. This is useful for picking up where a previous session left off, or for a GM to prepare a situation ahead of game play. The program looks in the 'scenarios' subdirectory for the necessary files.

**LoadSys**: Opens a dialog querying for a pre-generated system name, after which the system is loaded. Vessels present are retained in their original positions and may require position editing to put them "where the action is". The program will look in the 'systems' subdirectory for the necessary file.

**LoadShp**: Opens a dialog querying for a saved vessel group name, which either removes the old vessels and substitutes the new ones in the presently loaded scenario/system, or adds to the existing units. The program will look in the 'ships' subdirectory for the necessary file. If the file is found then the user will be queried as to whether they wish to replace the existing units or add the new ships to the existing units. The chance to abort the load is also offered here. No maneuver data is included and the newly loaded ship(s) will have to be given any such orders, though they may have preexisting vectors.

**SaveAll**: Opens a dialog querying for a scenario name to be saved. All present program data, thrust orders and toggle/mode states are saved in the 'scenarios' subdirectory. These can later be recalled by LoadAll. If the 'scenarios' subdirectory does not exist, the program will create it and warn the user to retry the save.

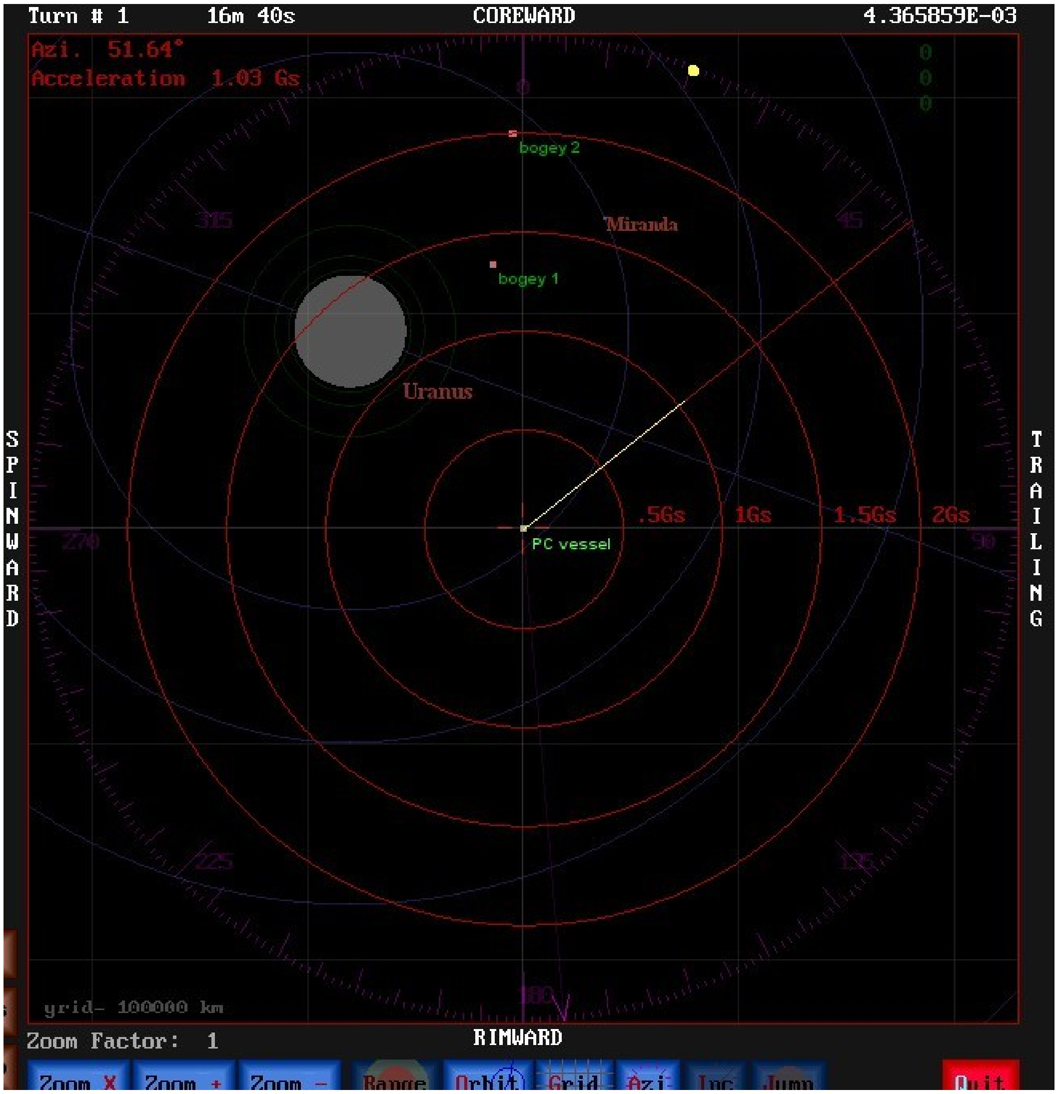
**SaveSys**: Opens a dialog querying for a system name. If none is given the program will adopt the name of the primary, rank 1 body, typically the system's central star. The system will be saved in the 'systems' subdirectory, but care should be exercised here. Pre-generated systems are saved with zero year ephemeris positions. If the same system name is reused, that original data will be overwritten. If this is not desired, then an altered system name should be used. Best practice is to save the system as a scenario if it already exists in 'systems' and save the 'systems' data for pre-generating initial zero year systems. This can be done using the SysInputIII system creation utility. If the 'systems' subdirectory is not present the program will create it and warn the user to retry the save.

**SaveShp**: Opens a dialog querying for a vessel group name. This may be an entire fleet or just the PCs ship. This can be used to save recurring vessels in a campaign setting. Present ship position and heading data is saved but no maneuver orders are saved. If the 'ships' subdirectory is not present the program will create it and warn the user to retry the save.

## Nav Controls: Graphic Vector Input

The vector button on the upper left corner of the control block gives the quickest, if not the most accurate, method of spherical coordinate entry for ship maneuvers. A left click upon the Vector button will result in the Astrogator switching back to overhead view and a series of concentric red circles appearing on it. These circles are labeled by G values, the outermost of which matches the maximum G rating of the active unit. Moving the mouse pointer into the Astrogator will initiate a red indicator line that moves with the mouse pointer as it pans around the center and extends out to the Azimuth scale. If the Azimuth scale was not enabled, it will still appear during this input phase.

While panning the mouse around the Astrogator screen, it will be noticed that part of the red line that extends from the center to the mouse cursor is actually yellow. This yellow line represents the relative thrust input and terminates at the mouse cursor position. In this manner both azimuth and thrust can be set with a single mouse click. It is possible to set a thrust somewhat faster than the vessel's maximum G rating by clicking beyond the outermost G circle, and if doing so, the user will be notified that the drives are being overdriven by the unit's speed indicator being displayed in red. Time for your engineer to show his qualities.




*Graphic Vector Input, showing azimuth and thrust*

When operating in 3D mode, the next phase of vector input will be to set an inclination. The view will change to show details along the ecliptic plane, an inclinometer scale will display and a Z-plane direction can be set in a similar fashion to how the azimuth was determined. A green shaded region is also displayed at the center of the Astrogator, clicking within the green shade will set the inclination at zero, precisely, which would otherwise be difficult to do with freehand mouse motions.

## Text Vector Input

It will be noted during use that there are problems with setting precise headings at low thrust using the graphic method, given that fine mouse movements can be difficult to achieve. To alleviate this problem, using hot key "**v**" opens a text based dialog box, where the vector data can be manually entered. A dialog box will open and query the user for precise Yaw, Pitch and Thrust values.

## Gs = 0 & Brake

These two controls address different concepts in inertial movement. Given that any navigation commands applied to a vessel will continue until changed or canceled, the "**Gs = 0**" control is that cancel command. The active vessel will cut thrust until directed to do otherwise. Yaw, pitch, speed and heading will remain unaffected and any coasting movement that the ship has attained will continue. The ship will simply not add any velocity from that point.

The "**Brake**" command acts differently. It will cause the vessel to apply a counterthrust against its accumulated velocity relative to the absolute coordinate system. It will not stop once it reaches zero velocity, but will begin accelerating in the opposite direction unless told to stop with "**Gs = 0**". Neither of these controls are functional when a ship is part of a fleet maneuver or its drives are disabled.

[more to come...]

## Hot key quick reference

| | |
|---|---|
| **A** | Azimuth display toggle |
| **B** | Belt/Ring display toggle |
| **D** | Density/Diameter toggle- when jump enabled |
| **E** | Edit ship menu |
| **G** | Grid/scale display toggle |
| **I** | Inclinometer display toggle |
| **J** | Jump display toggle |
| **O** | Orbit track display toggle |
| **Q** | Quit- autosave on exit |
| **R** | Range band display toggle- active unit |
| **S** | Save scenario dialog box |
| **T** | Turn execute |
| **U** | Undo previous turn |
| **V** | Vector input- text dialog box |
| **X** | Zoom extents- zoom factor 1, all ships in display |
| **Z** | Gravity zone display toggle |
| **3** | 2D/3D view toggle- alternate overhead to previous rotation |
| # | Show planet rank toggle- displayed after name |
| + | Zoom in |
| **-** | Zoom out |

| | |
|---|---|
| **Ins** | Add a new ship and make active |
| **Del** | Delete presently active ship |
| **Up** | Move active unit up list |
| **Dn** | Move active unit down list |

Star System Worksheet for _________________ Sector:_________________ Subsector:______________

| Rank | Name | Parent | ------Radius (kms) ------ | | -------Period------- | | Density | Azimuth | Class | Notes/ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Planetary | Orbital from parent | Orbital (years) | Rotation (days) | 1=Earth | zero year degrees | (stellar) (Gas giant) (etc.) | Zero year azimuth |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |

For use in arranging data for entry with SysInputIII.exe