

# **SPACEWAR**

*Dan Rosenblatt*

Systems  
Sigma Design, Inc.

## *ABSTRACT*

SPACEWAR is a multi-user, asynchronous, space battle game in a 3-dimensional universe. You play with/against other players while the computer controls the aliens (which are always against you). The game does not wait for you to 'take your turn'. The 3-dimensional universe is infinite, and it gets a bit complicated (and frustrating) figuring out where you are, where you want to go, and how to get there. Besides all this, you even have to build (configure) your own ship, and how well you do that will greatly affect how you fare in battle.

Changes to the game/documentation from the previous version (Jan 17'86) are marked with '|' in the right margin.

24 February 2023

# SPACEWAR

*Dan Rosenblatt*

Systems  
Sigma Design, Inc.

## 1. INTRODUCTION

This game is based on a previous version written in F77 that ran on the Mark III\* timesharing system, which was unsuccessfully transported to DTSS at CSS (oh well), which was based on David Fotland's version on the UNIX\* system, which is loosely based on Nat Howard's star trek game, which is loosely based on a game written by Peter Langston at Harvard. (whew!)

It is a multi-player game; up to 30\* people can play simultaneously. All players are in the same 3-dimensional universe at the same time and can battle, aid, or ignore each other as they wish.

It is an ongoing game, with the universe and your ship returning to the appropriate spots each time you re-enter the game.

Input is asynchronous. The game doesn't stop and wait for you to give it a command. If you do nothing, the game goes on around you and you may be attacked, or fall into an object (crash), etc. Whenever you give a command, it is acted upon.

## 2. ENTERING THE GAME

For the moment, the game is available on wombat, and to play it you must have a login on that machine. Then run 'psw' (play spacewar). If there is spacewar news, it will be displayed first. If the game isn't running, you'll get a message to wait a few moments while the universe is initialized.

You will be asked for your name. It can be up to eight (8) characters upper and lower case, including blanks. You will be asked for your password. It also can be up to eight (8) characters upper and lower case, including blanks. Your name and password as given to and used by SPACEWAR has nothing to do with your UNIX login name and password. If you are prompted with:

Play/See/Info/Complain/Mail/reBuild/Usercmd/Logoff/Who>

then you are ready to play SPACEWAR. Respond to this command prompt with at least 1 letter, upper or lower case, and a carriage-return.

There are four (4) special characters for editing/signalling. The backspace (^H) key is the character erase. It removes the character just to the left of the cursor and backs the cursor up one (1) position. If there are no more user-entered characters, nothing happens. The cancel (^U) key is the line erase. It erases the entire line of user-entered characters and backs the cursor up to the beginning of the user-entry area. The interrupt (usually

---

\*Mark III is a trademark of GEISCO

\*UNIX is a trademark of Bell Laboratories

\*the actual number depends on the particular machine

^C) key is the attention/restart signal. It works anytime except when you are actually playing (in your ship in the universe). If you think the game daemon is broken, hung, ..., try this key; it can't hurt. The quit (usually ^^) key is the panic/escape signal. It works anytime AND when you are actually playing, will get you out of the universe (and your ship), sometimes destroying your ship.

### 3. LEAVING THE GAME

To leave the game in an orderly fashion, type an 'l' (L) upper or lower case to the command prompt. You can also type the quit key (panic/escape signal). You can also just break the phone/network connection which will have the same effect as typing the quit key.

To leave the universe when you are actually playing 'in your ship', you must be docked with an object (planet, moon, asteroid, starbase) OR nothing bad must be near you (the BAD sensor must be 0). If you force a logoff/disconnect by quitting or breaking the phone/network connection without being docked at an object AND something bad is near you, YOUR SHIP WILL BE DESTROYED.

If the game daemon dies, the most that can be lost is the last 59 game cycles and no ships will be destroyed.

### 4. THE COMMANDS

The Play command is discussed later.

The See command is for seeing players/ships standings. The first section shows (not in any particular order) each player's name, the craft name, the craft points, time, and kills. The second section shows (by player) two lines per player. The first line has the player's name, total number of active ships, then average/maximum/ranking for points, time, and kills. The ranking is based on maximums. The second line has number of ships lost, points lost, time lost, and kills lost.

Complain is just mail to the keeper of the SPACEWAR game.

Mail is for handling and sending mail. When you first login to the game, you will be told if you have new mail. Your mail is separated into 2 groups, old and new. When you say delete, only old mail is deleted. When you say new, new mail is shown to you and then becomes old mail. When you say old, old mail is shown to you and retained.

The Usercmd is for building user-defined commands to be used during playing. The command name can be 1 to 8 characters upper and lower case with no embedded blanks. Any basic playing command can appear in a user-defined command except for phaser, torpedo, and another user-defined command (no recursion). Argument substitution is accomplished via \$1, \$2, ..., \$9 appearing in the user-defined command with actual values being substituted from the original command arguments. See the example at the end of 'hopefully helpful hints'.

### 5. THE SHIPS

Before you can play, you must build your ship. There is no limit to the number of ships you can build. You must choose a ship name (up to 12 characters upper and lower case including blanks) and from 3 hull types. Type 1 is rocket shaped and has the expected capabilities, i.e. good rockets. Type 2 is spherical with lots of floor space and good shields. Type 3 is like the starship Enterprise and is designed around huge warp engines, so that's what its

best at along with good phasers. Type 3 ships are good at smash and run-away battles, while type 2 ships are best at sit and slug-it-out battles.

Each ship is comprised of up to ten subsystems: Solar energy, Antimatter energy, Warp drive, Rockets, Dilithium, Phasers, Shields, Torpedoes, Damage control, and Security. There are other yet to be implemented weapons but to get them you must pillage another ship that has them (such as aliens).

Each subsystem, if you choose to have any of it, uses up a certain base amount of floor space and crew. The more you ask for of the subsystem, the more floor space and crew you use up.

Your shields and phasers run off of energy which is generated by the solar and antimatter energy sources, and stored in your dilithium crystals. Antimatter energy is delivered at a fixed rate dependent on ship type and percent of subsystem. Solar energy is delivered at a rate dependent on ship type, percent of subsystem, and how far away from the suns you are.

Warp drive takes you from here to there instantly, without any change in velocity. Warp drive runs off its own energy supply, completely separate from your shields, phasers, dilithium crystals, solar, and antimatter. Warps of the same distance will always cost the same amount of warp energy, but the recharge rate depends on ship type and percent of subsystem. In addition, the warp drive is not perfectly accurate. The more of it you have, however, the more accurate it is.

The base floor space and crew for rockets covers the engines, and the additional floor space and crew covers fuel. The maximum thrust of the rockets is determined by your hull type (most for type 1, least for type 3).

The percentage of the torpedo subsystem determines the number of homing torpedoes you can carry. Their maximum thrust is 50.

The more damage control you have, the faster your ship will repair itself when it gets damaged in battle. There are two types of damage: temporary and permanent (lack of spare parts). Permanent damage can only be fixed at a starbase.

Security is useful to protect yourself against being pillaged by an enemy ship. Currently, there is no pillage command.

## **6. DAMAGE**

There are two types of damage: temporary and permanent. Temporary damage can be fixed by your damage control personnel, but permanent damage can only be fixed at a starbase. You can be damaged by enemy phasers or torpedoes. If your hull is damaged enough, it causes more damage to your whole ship (radiation leakage, atmosphere loss, heat loss, ...). Damage to any subsystem over 45% becomes permanent and damage control can only repair it back down to 45%. If you are docked and are damaged or need supplies, the reBuild command will replenish supplies and fix damage, though permanent damage can only be fixed at a starbase.

If your ship's solar or antimatter energy subsystems are damaged, you simply generate that much less energy to be stored in your dilithium crystals. Damage to the dilithium crystals affects the amount of energy that can be stored. If your warp drive is damaged, you still warp the same distance, but the accuracy is decreased (to 0 if there is enough damage). Damage to your rockets affects maximum acceleration. Damage to your phasers reduces the amount of damage they can inflict (they can't even be fired if there is too much damage). If your shields are damaged, they will valiantly retain the

same strength, but the amount of energy required to do so increases. If your torpedo tubes are too damaged, you won't be able to fire any torpedoes; otherwise they work fine. If damage control is damaged, it can't fix damage very well.

Some subsystems are easier to damage than others. You will notice after a few battles that the same subsystems always seem to be hurt the most. For example, the solar energy collectors are on the outside of the ship, so they are very vulnerable.

## 7. THE UNIVERSE

The universe is infinite and unbounded. It is populated by two stars, many planets, moons, and asteroids. There are also a couple of starbases, aliens, maybe a black hole, and of course other (enemy?) ships. Gravity exists, and works as expected with a constant of 1 ( $f = m_1 m_2 / (r^2)$ ). It is considered bad form to collide with a solid object (you crash and your ship is destroyed). The first time you enter the game with a new ship, you will be near an object and facing that object. If you leave the game after docking with an object, the next time you return you will still be docked. If the game daemon dies for some reason, you will return to within 59 game cycles of where you were when it died.

The distance between the two stars is about 200,000. The planets are between 1000 and 50,000 from the suns. The basic coordinate system is cartesian (rectangular, x/y/z), and helio-centric. The x and y axes are in the plane of the ecliptic. The z axis is perpendicular to them. The order of output is always x,y,z. Some of the screen readouts are in spherical (polar) notation; r (distance from center), theta (ccw from positive x axis), and phi (down from positive z axis).

## 8. THE SCREEN

The screen in front of you during the game contains a vast amount of information. In the top center is your viewscreen, your window into space, creating a cone of vision. This shows a picture of the space in front of you that depends on your direction and viewing angle which in turn determines your maximum viewing distance. The center of the cross-hairs is straight ahead. Objects are represented symbolically on the screen. Objects that are further than you can view but still within 5000 are shown as unidentified. Objects that you are docked with or that are docked with you are shown in the center of your viewscreen in background intensity.

Item	Symbolic display
Stars	*
Planets	O
Moons	o
Asteroids	o
Starbases	#
Human ships	1 2 3
Alien ships	4 5 6 7 8 9
Torpedoes	+
Unidentified.	

Around the sides of the screen are status displays. Down the left side are the direction you are pointed (polar) and possibly AUTO or FFWD, position (rectangular), velocity (rectangular), angle of view and viewing distance, how

many objects you can see within the given viewing distance, and how many bad objects (that can hurt you) are within 5000. Below these are the homing channels, six of them, showing distance.

Down the right side of the screen are your subsystems, what level they are at and how much damage each has.

System	Meaning of level
-----	-----
Hull	Current acceleration
Solar E.	Energy being produced
Antm. E.	Energy being produced
Warp	Warp energy stored
Rockets	Rocket fuel stored
Dilith.	Phaser+shield energy stored
Phasers	
Shields	Shield protection
Torps.	Number of torpedoes stored
Dmg.Con.	Current subsystem being fixed
Security	

At the bottom right corner of the viewscreen is game time. The bottom of your screen is divided into two halves. The left half shows your last 7 commands (command history) and the most recent command (indicated with a # just to the left of the command). The right half is for various reports and asynchronous events, like reporting attack damage and torpedo actions. Just above the command history area is a line for radio messages. Line 24, just below the command history area, is where your keyboard buffer is displayed.

## 9. POINTS

A score of points, game time, and kills is kept by ship. Losses of points, game time, and kills is kept by player (which happens when a ship is destroyed). You get points for inflicting damage (1/10 of the damage inflicted). In a multi-player battle, the player credited with the destruction of a ship is the player who fired the last shot at that ship (including torpedo hits). If a ship self-destructs, the player who fired the last shot is still credited with the destruction. You must be in the universe when a ship is destroyed to be credited with its destruction.

## 10. PLAYING

When you enter the Play command, you will be asked for your terminal type. SPACEWAR on UNIX uses termcap, so enter the termcap name for your terminal. There is the ability to change which background/foreground mode is used and also to reverse the orientation (typically they are either reverse video or half intensity). This is done by preceeding the terminal name with a period, comma, or semicolon (.,;). Period reverses the orientation of the standard so/se (standout). Comma uses us/ue (underscore). Semicolon reverses the orientation of us/ue.

Many of the play commands take some sort of argument, which comes after the command. Commands may be in upper or lower case, and you must type enough of the command so that it is uniquely identified (minimum length is shown in parenthesis after the command). If the command cannot be identified or the argument is bad (non-numeric, out of range, ...), the command is prepended with '???' when it is copied to the command history area. If the command cannot be carried out, a message of why is appended to the command when it is

copied.

**ERASE (1)**

Erases and redraws the screen.

**ANGLE (2)**

Changes your viewscreen viewing angle. Valid argument range is  $0.0 < n \leq 180.0$ . Your viewing distance is related to viewing angle by an inverse square rule ( $1/a^2$ ). Initially set to 45.0 with a viewing distance of 5000.

**RIGHT (2)**

Changes your viewing direction to the right (objects on the screen move to the left). Valid argument range is  $0.0 < n \leq 360.0$ . If no argument is given, the change in direction is 1/2 your current viewing angle. If you are autopiloted or facing forward, it is turned off.

**LEFT (3)**

See RIGHT.

**UP (2)**

See RIGHT.

**DOWN (3)**

See RIGHT.

**HOME (1)**

Locks the homing channel given as the argument onto what you are autopiloted on (you must be autopiloted on something). Valid argument range is  $1 \leq n \leq 6$ . Homing will not lose an object just because you or it warps (autopilot will). The homing channel will lose what it is locked onto if you tell it to lock onto something else, or if what it is locked onto leaves the universe, or if you leave the universe and it is not an object.

**UNHOME (2)**

Another way to make a homing channel lose what it is locked onto. Valid argument range is  $1 \leq n \leq 6$ .

**FACEFORWARD (2)**

This command will point you and keep you pointed in the direction that you are moving (velocity). You will get a complaint if you are not moving. If you are autopiloted, the autopilot is turned off.

**AUTOPILOT (2)**

This crucial battle command locks you onto the thing that is closest to center of your viewscreen and closest to you so that the thing is always in the center of your viewscreen. You cannot fire any weapons at something unless you are autopiloted on it. It will lock onto the closest to

center/closest to you thing that is within 5 degrees of center or half your viewing angle, whichever is smaller. If given an argument in the range  $1 \leq n \leq 6$ , it will lock onto the thing that is on the given homing channel. If the thing that you lock onto is docked, you will be locked onto the thing it is docked with (except your own ship). If the thing that you are autopiloted on docks, the autopilot will shift if the thing docks with another ship, and it will be lost if the thing docks with you or an object. If you are facing forward, faceforward is turned off.

#### **SHIELDS (2)**

Takes an argument in the range  $0 \leq n \leq 100$  and sets your shield protection to that percent. If you use up all your stored dilithium energy, the shield protection drops to where it uses up all the energy your solar and antimatter energy sources produce.

#### **THRUST (2)**

Thrust with the rockets. Fuel is expended as you accelerate or decelerate. If you specify a thrust greater than your maximum acceleration, thrust is applied over several iterations. This command takes up to 4 arguments. The first is how much thrust to apply in the forward direction (the direction you are pointing). It can be negative to slow you down (assuming you are facing forward). The other three arguments are thrust in the x, y, and z directions and can also be negative. If you are docked, you will no longer be docked. Multiple arguments must be space separated.

#### **STOPTHRUST (2)**

Cancels any pending thrust.

#### **WARP (2)**

The other way to get around the universe. Moves you forward (direction you are pointing) instantaneously. Give it an argument of the distance you wish to move in thousands of units (can be negative to move you backwards). If you are docked or autopiloted, you no longer will be. Others that are docked or autopiloted with/on you have theirs turned off also. Any warp takes at least 500 units of warp energy.

#### **LEAVE (3)**

Ejects you from the universe in an orderly fashion, but only if you are docked with an object (planet, moon, asteroid, starbase) or if nothing bad is near you. Otherwise you remain in the universe. Any torpedoes that you have fired keep going but no credit is given to you when they inflict damage or kill.

#### **DOCK (3)**

Docks you with the nearest thing (except stars or black holes). You must be within  $5 * (\text{thing radius} + 1)$  to dock; be forewarned that crashing distance is  $2 * (\text{thing radius} + 1)$ . You must also be moving slow enough to be within docking distance for at least 2 game cycles. You cannot be directly attacked once you dock, though if you dock with another ship that is attacked you will sustain damage by being close to it. Any autopilots, docks, or torpedoes on you are transferred to the thing you dock with or are lost depending on the type of thing you dock with. The thing



you dock with will be shown in the center of your viewscreen in background mode.

#### **RADIO (2)**

Sends a radio message (starting with the first non-blank character after the command) in the direction you are pointing in a cone that is within 5 degrees of center or half your viewing angle, whichever is smaller.

#### **WHO (2)**

Tells (in the report area) who is currently playing and their ship name. Since more players can be playing than can be reported on, who takes an optional argument of how many output screens to skip before putting out to the report area.

#### **REPORT (3)**

Reports (in the report area) on all the objects that you can see on your viewscreen. Each line includes the x,y viewscreen coordinates of the thing, the symbolic picture, and the distance to the thing. For objects, the radius and mass are shown. For ships, the player's and ship's name, hull damage, and shield protection. For torpedoes, the hull damage. For aliens, hull damage and shield protection. After all this follows optional information in parenthesis telling you if you are autopiloted (A), docked (D), or homed (1/2/3/4/5/6) on the thing described. Since more things can be on your viewscreen than can be reported on, report takes an optional argument of how many output screens to skip before putting out to the report area.

#### **PHASERS (1)**

You must be autopiloted and not docked. Give it an argument of how much energy to apply,  $1 \leq n \leq 1000$ . The amount of damage inflicted depends on how much phasers you have, how much energy you apply, how far away the target is, and how much shielding the target has.

#### **TORPEDO (2)**

Launches a torpedo at what you are autopiloted on. You will be informed in the report area when the torpedo blows up, crashes, is destroyed, or misses.

#### **FIX (2)**

This command directs your damage control to fix the subsystem given as argument,  $0 \leq n \leq 11$ . If no argument (or 0) is given, it goes back to fixing all subsystems. Once a particular subsystem is fixed (remember that permanent damage, >45%, cannot be fixed), it goes back to fixing all subsystems. Due to the law of diminishing returns, damage control is not as efficient when fixing one particular subsystem.

#### **SENSORS (2)**

Reports (in the report area) on all the objects that are within 5000 units of your ship. Each line includes the x,y viewscreen coordinates of the thing if it is visible on your viewscreen, the symbolic picture, the distance to the thing, and the direction vector to the thing. After this follows optional information in parenthesis telling you if you are

autopiloted (A), docked (D), or homed (1/2/3/4/5/6) on the thing described. Since more things can be within 5000 units than can be reported on, sensors takes an optional argument of how many output screens to skip before putting out to the report area.

#### **LOCKON (2)**

Points you in the direction of where the last attack came from and tries to 'autopilot' on anything in that direction/location. Since aliens and other players can move after they attack you, the 'autopilot' may not be effective.

#### **REDRAW (3)**

Redraws the screen just like 'erase' but without first blanking out the screen.

#### **-cmd**

Execute a user-defined command with argument substitution. Arguments must be space separated.

### **11. HOPEFULLY HELPFUL HINTS**

#### **The reBuild command:**

When reBuilding your ship, you can give a null response (just carriage-return) to the percent of subsystem prompt and it will leave you with whatever percent you already have. You must cycle through all your subsystems to replenish supplies and fix damage.

Playing your ship in the universe:

Don't forget that if you force a logoff when you aren't docked with an object and something bad is near you, your ship will be destroyed. This is to deter players from avoiding battle by escaping from the universe in an unfair way.

The direction indicators in the upper left of your screen show the direction your ship is pointed (determining what you can see on your viewscreen), which is not necessarily the direction you are moving. The direction your ship is pointed is affected by the commands LEFT, RIGHT, UP, DOWN, FACEFORWARD, and AUTOPILOT. The last two commands continuously adjust the direction your ship is pointed until something turns the command off. These two states are indicated with 'AUTO' or 'FFWD' just after your direction indicators.

The direction and reports of radio and attack direction are all in polar coordinates. Position and velocity are in rectangular and don't relate very well (for humans) to those in polar coordinates. To see a radio broadcaster or an attacker, adjust your direction to be the same as what was shown in the report area.

If you can't see an object you're pointed at, it's probably because it's further than your viewing distance. Increase your viewing distance by decreasing your viewing angle, but don't leave it that way for too long because someone could more easily sneak up on you. Don't forget to watch the 'BAD' indicator and to use the SENSORS cmd to see how many potentially dangerous things are within 5000 units of you.

Since enough hull damage (>59%) will cause more damage to your ship, don't let it go too long when you're in battle (use the FIX command).

If your velocity starts changing for some unexplained reason, you probably are being pulled toward an object by its gravity. In other words, you are in danger of crashing, though not for a little while. Sometimes you won't crash, but will be flung off into space at a high velocity. Lotsa luck finding your way back.

It's strongly suggested, since docking is one of the most difficult things to do and the only way to fix a damaged ship (at a starbase), to create a user-defined command something like this:

```
Play/See/.../Who>U
```

```
command/L(ist)/.(quit)>fixvel
```

The currently defined command (if any) will be displayed.  
A single dot (.) at the first prompt will retain it.

```
>fac  
>thr -$2  
>aut $1  
>thr $2  
>fac  
>.
```

When playing, assuming the object you're trying to dock with is on homing channel 4 and you're thrusting toward it at say 25, the playing command:

```
-fixvel 4 25
```

would correct your thrust to keep you heading toward the object. Don't forget that all objects orbit in a circular path. The best chance for docking is if you follow the object and 'creep' up on it from behind. The 'faceforward' command will show you how often you need to correct your course by displaying the object you're following creep away from the center of your viewscreen.