|  |
| --- |
|  |
|  |
| **Star Trek**  **3D** |
| User Guide and Release Notes  Version 1.0  All work Copyright © 2021 by RCS Computing.  All rights reserved. |
| **Michael Raspuzzi** |
|  |

|  |
| --- |
| March 8th 2021 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Modifications |
| 1.0 | 03/08/2021 | M. Raspuzzi | Initial Draft for alpha use. |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Game Overview 4](#_Toc66106328)

[Game Initialization 5](#_Toc66106329)

[Connecting to The Game Server 5](#_Toc66106330)

[Main View 6](#_Toc66106331)

[Controls 6](#_Toc66106332)

[Ship Functions 6](#_Toc66106333)

[Game and View Controls 6](#_Toc66106334)

[Short Range Scan and Library Computer Windows 6](#_Toc66106335)

[Game World 6](#_Toc66106336)

[Release Notes 7](#_Toc66106337)

[Future Features 7](#_Toc66106338)

# Game Overview

Around the year 1980, a game called VTTREK was developed at the former Digital Equipment Corporation (DEC). VTTREK was written in MACRO-20 for the TOPS-20 operating system and took advantage of advanced graphics on a VT-100 terminal. The game was eventually modified to take advantage of REGIS graphics on terminals capable of supporting REGIS.

The game consisted of a 3 dimensional universe with 4 Federation Starships along with 4 Federation Starbases and 4 Klingon Ships and 4 Klingon Outposts. The game world placed all of the ships and bases throughout the game universe along with 16 planets. The planets could be approached and captured and you could attack any object within the universe.

VTTREK was designed using special page mapping functionality on the DECSYSTEM-20 running TOPS-20. It only ran on models 2060 and 2065. The 2020 model was not capable of the advanced paging features provided by the KL10 processor and therefore, the game could not be played on the KS10 based 2020. The shared page mapping was used to maintain game event synchronization. Each player’s game instance retrieved game events from the event queue in this area.

Although game play was limited to a single DECSYSTEM-20, it was developed with a client/server model. Each player could control a ship on a RS232 terminal line and there could be up to 8 players in a single game. Since all ships were equal, it would take teamwork and a coordinated effort for one side to be victorious over the other.

This game was designed with the underlying functionality of VTTREK in mind with the following goals:

* Use modern graphics and models for game objects
* Render the main view using the models and objects provided
* Maintain a client/server relationship for multiple player support
* Be as true to the original VTTREK whenever possible
* Make improvements to support a more modern game over the original VTTREK
* Use a modern language for implementation

In keeping with the above design goals:

* Highly details game objects were gathered from the Internet
* DirectX was chosen to render the game objects (limiting this to Windows only at the current time)
* A TCP sockets model was developed so the game could be played by multiple players thus maintaining the client/server model
* C++ was chosen as the game implementation language for its object oriented support

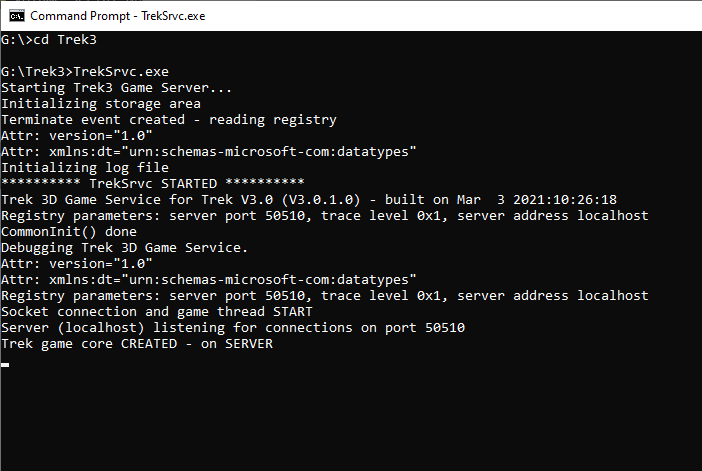
VTTREK presented the game world view to each player through the view screen of a starship as seen in the Star Trek TV series and movies. This game takes that view and moves it behind and slightly above the player’s ship instead of a view screen presentation. Note, this view can be modified with various controls to change the world view as seen by the player.

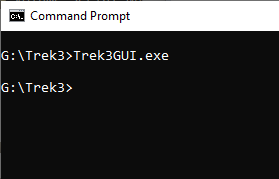
The game author would like to thank the various owners of the 3D objects that are used in this game. When possible, the credit for each object has been maintained in the object’s directory. These models were generously made available on the Internet by their authors and contributed significantly to the look of this game.

# Game Initialization

At this time, the game is not fully complete. It does not have an installation wizard. Instead, it is distributed in a ZIP file that must be unzipped into a directory. To play the game, you must start the server portion first and then the user interface.

Follow these steps to setup the game (this assumes rudimentary knowledge of Windows 10 usage):

1. Unzip the game contents into a directory. This procedure assumed the files are placed in C:\TREK3
2. Open a Command Prompt (DOS). Type ‘CMD’ in the Windows 10 search bar. It does not have to be run as an administrator.
3. In the command Window, type ‘cd C:\TREK3’. You should now be in the game directory
4. Run the game server:
5. Open another Command Prompt (DOS).
6. In the new DOS window, type ‘cd C:\TREK3’
7. Once in the game directory, run the UI program

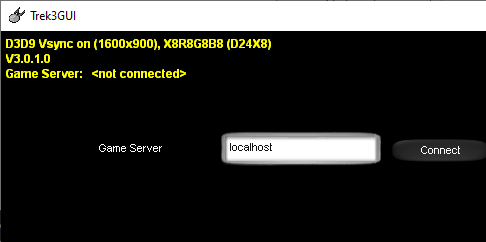
:

**NOTE**

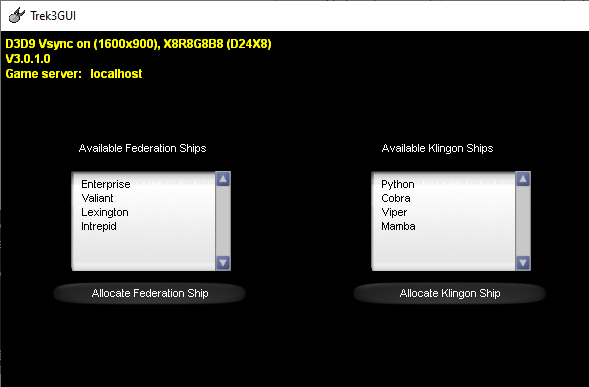
Due to the detail of each game object, it takes a while for the game interface to fully load and initialize.

# Connecting to The Game Server

By default, the above instructions show how to run the game server on the local system. When the game interface loads, it asks which game server to connect. You should not change the game server and simply connect to the localhost:



Next, you must select which ship you would like to control. Select the ship from the choice view (see below) and either double click on the ship name or select the ship and press the corresponding allocate button:



Select your ship quickly as there is a timeout period.

# Main View

Below is an image of the main game view. It is fixed in a window of size 1600x900 resolution. Currently, there are no plans to allow the main view window or the Short Range Scan and Library Computer windows to be resized.

**NOTE**

Due to the high quality and details of the game objects, it is recommended that this game be played on a system with a modern graphics card of at least 4GB of video RAM. An underpowered graphics card will show lag in attempting to render each scene.

# Controls

There are currently no plans to implement joystick handling at this time.

Below is a list of commands and controls available during game play.

## Ship Functions

### Keypad Enter/Return Key

Moves the player ship in the direction it is facing at the current warp factor. The higher the warp factor, the further each press of this key moves the ship. Higher warp factors consume more ship energy.

### Left Arrow Key

Rotates the player ship one degree counterclockwise.

### Right Arrow Key

Rotates the player ship one degree clockwise.

### Up Arrow Key

Rotates the player ship one degree up.

### Down Arrow Key

Rotates the player ship one degree down.

### Delete Key

Rotate the player ship 45 degrees counterclockwise.

### End Key

Rotate the player ship 45 degrees clockwise.

### Page Up Key

Rotate the player ship 45 degrees up.

### Page Down Key

Rotate the player ship 45 degrees down.

### Plus Sign (‘+’ Key)

Increase the player ship warp factor setting by 1. Ship cannot go above warp factor 8.

### Minus Sign (‘-‘ Key)

Decrease the player ship warp factor setting by 1.

### S Key

Perform short range scan. Short range information is displayed in the Short Range Scan window.

### Keypad 0

Perform long range sensor scan. If the player ship has a target locked, long range scan information pertaining to the locked object is displayed. If no object is locked, the long range sensors will scan in a tight cone straight ahead of the player ship. If an object is within the long range sensors, information pertaining to that object is displayed.

### R Key

Places the player ship on RED alert.

### Y Key

Places the player ship on YELLOW alert.

### G Key

Pleases the player ship on GREEN alert.

### L Key

Set the player ship target lock to the next object in the short range scan window. If no object is locked, then the first object in the short range scan window is locked.

### U Key

Clears the player ship target lock. Target lock is no longer active.

### C Key

Capture a planet. Note, to capture a planet, it must be target locked.

### E Key

Refuel the player ship and restock photon torpedoes. The player ship must be within refueling distance of a friendly planet or a friendly Starbase/outpost. Planets do not refuel or restock the player ship as much as a base or outpost.

### Energy and Shield Slider

Each ship has a limited supply of energy that can be divided between the shields and ship functions. When the energy bar is changed to increase or decrease the energy, the shields change in a similar manner (less energy, more power to the shields and vice versa). Ship energy can be controlled by either increasing or decreasing the ship energy as well as increasing or decreasing the shield energy.

Shields will absorb any weapons fire and will diminish by the amount of weapon energy directed at them. Once the shields are down, the amount of energy absorbed by incoming weapons fire is more potent and causes more damage to the ship. When energy and shields have been depleted, the ship is destroyed.

## Game and View Controls

### Escape Key

Quit out of the game.

### Mouse Button 2

Changes the camera view of the game for the player. Press and hold mouse button 2 and drag the mouse to modify camera position and world view.

### Mouse Wheel

Camera zoom control. Spinning the mouse wheel will increase or decrease the camera zoom. The zoom follows the current camera view.

### Insert Key

Restores camera and world view to its game start location.

### Home Key

Instantly changes the camera view to be directly behind the active player’s ship.

# Short Range Scan and Library Computer Windows

The information contained in the Short Range Scan window is information that was valid when the last scan was completed. If your ship has moved, the information in the scan window may no longer be valid. Simply perform a new Short Range Scan to have the information brought up to date.

Objects are added to the Library Computer as they are scanned and are updated with their latest location each time they are scanned. In other words, an enemy ship location in the library computer is its last known, scanned location. If it has moved and has not been scanned since moving, you may not find the ship in its Library Computer location. Bases and planets do not move so once they are scanned, their position becomes known and will not change.

# Game World

The game world is a three dimensional outer space arena. It is essentially divided up into 8 areas. The Federation has control of 4 game areas (each area has a ship and a base). The Klingons also have control of 4 areas each with a ship and an outpost. No planets are known in the Library Computer to the Federation or Klingons when the game starts. By exploring the world and scanned the areas, you will discover various objects. When objects become known, they are added to the Library Computer with their last known location.

# Release Notes

Here is a list of known issues for release 1.0.1.0. These issues will be addressed in a future build:

* If the player ship selection times out, the UI does not properly go back and ask to reconnect to the game server
* The target lock painted on the screen is not accurate
* The bearing and elevation values can be off by +/- sign as the game logic does not determine the ship orientation relative to other objects
* There are no weapons available at this time
* The information in the Short Range Scan and Library Computer windows are not properly aligned by column

# Future Features

Here is a list of game features that are not currently implemented by are planned in the future (in no particular order):

* Implement phasers
* Implement photon torpedoes
* Implement special effects for weapon hits
* Implement explosions for destroyed objects
* Implement planet rebellion when attempting to capture
* Toughen Starbases from weapons fire
* Add missions for non-human controlled ships
* Add ship to ship messages
* Implement sound effects
* Highlight locked target in Short Range Scan window
* Implement an “end game” to indicate when a team has won
* Create a software install package