20,000 Light Years Into Space

15th Anniversary Edition

Jack Whitham, March 2021

Preface

15th Anniversary Edition

“20,000 Light Years Into Space” was written for the second Pyweek competition. The first version was written in a week in 2006. I got involved in the competition by accident. I’d set out to learn the Python programming language, and then I stumbled on a thing called Pygame which made it quite easy to write games, and Pygame had put up an announcement about Pyweek, so it all happened by chance. The game is about steam because that was Pyweek 2’s theme.

Pyweek, Python and Pygame are all still going strong. Python is now at version 3.10 and is a hugely popular programming language. Pygame has reached version 2.0. There have now been over 30 Pyweek competitions. Some of the people who once participated in Pyweek have gone on to make successful commercial games.

For a long time there were no updates for this game because I didn’t think I had time to work on it, and the technical debt level was very high. The underlying Python technology became obsolete. I made various poor choices during development. The game had no test cases, making it hard to maintain. The design made assumptions about monitor resolution which were quickly invalidated by high-definition displays.

For the game’s 15th birthday I have taken the time to update the game for newer systems. It uses Python 3, there is a thorough set of test cases, and the software is technically much better, close to the state of the art for Python software development. Graphics and sound are also improved: I have returned to the original files to recreate them in higher resolutions.

This is, however, more of a “4K Blu-ray restoration” of an old movie than a remake or a George Lucas “special edition”. The technical quality is better, but the actual game is the same. Many criticisms of the game are still just as valid. If 20KLY was the game equivalent of “Plan 9 From Outer Space”, then it definitely still is, but the picture and the sound quality are better than ever. And you can “fast forward”, too.

Was it worth it? Well, I enjoyed the technical challenge of updating it, as well as replaying it after so long. Hopefully you’ll enjoy playing it too. Good luck!

Jack Whitham

March 2021

How to install and start the game

Windows 10

Download the .exe installer from https://www.jwhitham.org/20kly/

Run the .exe installer from your Downloads folder: follow on-screen instructions.

After installation, press the Windows key and begin typing "20K Light Years". Press Enter when the OS finds the game.

Linux

Download the Linux zip file from https://www.jwhitham.org/20kly/

Extract the Linux zip file to a suitable location on your hard disk.

Run the “20kly” program.

Source code

Use “git” to clone the source code from https://github.com/20kly/20kly

To run the game from the source code, clone the 20KLY repository, and then run “python lightyears”.

You may need to enter “python3” instead of “python”. You may also need to install Pygame, for example with “pip install pygame”.

Introduction

It is 1895 and you are a member of the British Space Expedition. You receive the following letter from your long-time friend and leader, Sir Phileas T. Verne.

> *The First of October, 1895*
>
> My dear friend,
>
> As you know, today we celebrate the opening of our first colony on the new world of Vaporia, some 20,000 light-years from England. Our colony is truly the most advanced structure that Science can create - a city on an alien world, entirely powered by steam from the planet's core. It has taken us five long years to get here by rocket-ship, and like the great Explorers before us, we shall make this foreign land our home: a new outpost of the Empire!
>
> As my most trusted lieutenant, I leave you in sole charge of the new city, while I begin exploration of Vaporia. There is much to be done:- you must expand the city, but before you can do that, you'll have to secure better supplies of steam.
>
> I wish you the best of luck, but must warn you that we may not be alone on Vaporia! I have detected evidence of civilisation here, so you may need to defend yourself. But I have faith in your abilities, my friend!
>
> The steam must flow,
>
> Yours,
>
> Phileas T. Verne

How to play

The objective of "20,000 Light Years Into Space" is to upgrade a City to a particular technology level. The City requires steam: and every upgrade will increase the requirement. The level you must reach is dependant upon the difficulty level of your game.

If the City runs out of steam before you can reach the required technology level, you lose the game. Therefore, you must built a network to carry steam to the City.

The links of the network are Pipes, and the connection points are called Nodes. If you build a connection point over a source of steam, it is a Steam Maker. It adds steam to your network. Other types of Node just route steam (and also store it, in small quantities).

The easiest way to learn the game is to play the Tutorial. The Tutorial is interactive, and takes you through the things you need to know in order to get started. However, if you prefer a traditional manual, read on...

Your First Game

Start the game by clicking 'Play New Game', then 'Beginner'. You can press Escape or click 'Menu' to pause the game at any time. When the in-game menu is on screen, the game is paused.

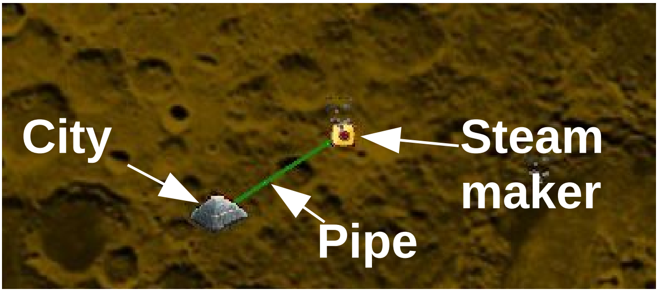


Figure 1: The City, a Pipe and a Steam Maker

Step 1: Secure more steam

When the game begins, your City already has one source of steam, a short distance to the east (Figure 1). That is enough for the moment, but you won't be able to upgrade the City very far without adding more sources. Click "Build Node" (as in Figure 2) and then click on one of the animated steam sources that you can see on the map. Choose one that is near to the City. An orange Steam Maker will appear over the source.

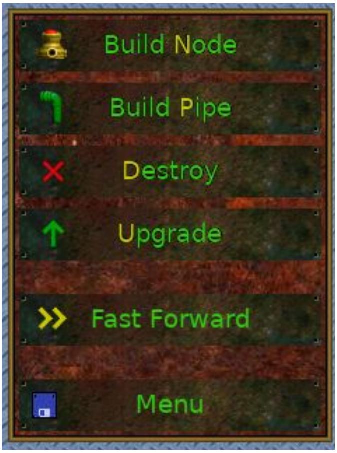


Figure 2: Game Controls

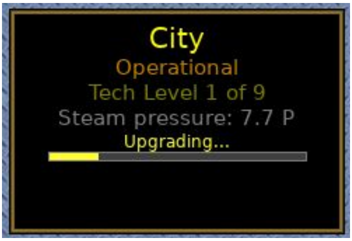
Next, select "Build Pipe", and click first on the City, and then click on the new Steam Maker. A red Pipe will appear between the two. The Pipe is red because it is not fully built: but building work will start immediately. Shortly, the Pipe will turn green: it has now been built, and is ready to route steam. You'll see the steam pressure increase if you click on the City or the new Steam Maker.

However, there isn't any steam to route, yet. The Steam Maker also needs to be built. That takes longer: all types of Node take a while to build. You can click on the Steam Maker to watch the progress of the construction.

Step 2: Upgrade your City

Now you have two Steam Makers. You are now ready to upgrade your City. Do that now by clicking Upgrade, then clicking on your City.

When your city is selected, you'll see information about the upgrade in a window on the right-hand side. This is illustrated in Figure 3.



City
Operational
Tech Level 1 of 9
Steam pressure: 7.7 P
Upgrading...

Figure 3: Upgrade in progress

As soon as the upgrade begins, the city will require more steam. You'll also get an extra work unit. Work units are used to build, repair and upgrade Nodes, Pipes, Steam Makers and also the City. Work units represent a capability to carry out work simultaneously: if you have 3 work units, then you can build, repair or upgrade up to 3 items at the same time. If more than 3 items need work, then they will wait in a queue. Understanding the queue is an important part of tactical play - see Hints and Tips (page 11) if you want to know the answer!

Step 3: Strengthen your Network

While your upgrade is running, you can expand your network. But first, you should strengthen it. You are about to be attacked by hostile forces! Mysterious aliens are present: they understand your network, and they will try to destroy the weakest points. There is one way to avoid this. You must build additional routes for the steam.

Figure 4 illustrates an example of this. There are two primary routes for the steam, directly linking the Steam Makers to the City. But there are also secondary routes, carrying steam through three Nodes. If the primary routes are damaged or destroyed, the secondary routes will be used.

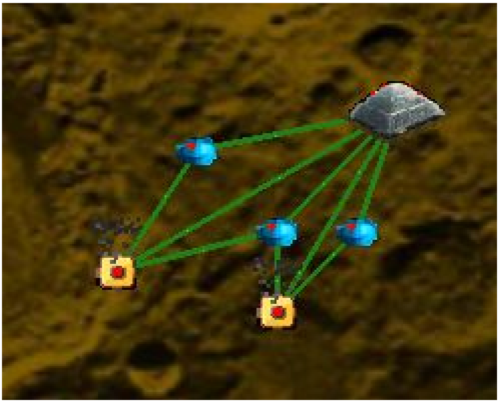


Figure 4: Primary and secondary routes

You don't have to do anything special to designate routes as primary or secondary. The steam moves automatically, following the path of least resistance.

For most of the game, you'll have to deal with attacks from aliens, sandstorms, and quakes. Only the early stage of the game is "quiet". You need to be ready to rebuild anything that gets destroyed.

Step 4: Expansion

The end goal is to reach a particular technology level at the City. But you can't upgrade the city unless you also obtain more steam. There are three basic strategies for obtaining more steam.

1. You can Upgrade your Steam Makers to make them produce more steam.
2. You can expand your network in order to build more Steam Makers.
3. You can improve your network, both by adding more pipes, and by Upgrading Pipes and Nodes, so that steam is routed more efficiently.

The safest option combines the three. You will always want to ensure that your network is resilient to the various types of attack. An upgraded City will provide you with lots of Work Units, allowing you to build and repair at high speed, but it will also require a great deal of steam. You can lose the game very quickly if supplies are interrupted! Monitor the City's steam usage and steam pressure closely.

Step 5: Stuck?

You may wish to look at the Hints and Tips (page 11) if you keep on losing the game. Scroll down....

Hints and Tips

- As a rule of thumb, you need one steam maker per city tech level. But you'll want more than that to survive attacks and damage from the environment.
- Don't start upgrading your city straight away. Get more steam first.
- Be on the lookout for pipes that have been broken by attacks. Repair them immediately. Always try to keep some Work Units available for emergency repairs.
- Things that aren't connected to the city don't get repaired, upgraded or built.
- Don't make your pipes too long. Have plenty of nodes to facilitate branching. Nodes cost more time to build, but they're really hard to destroy and they permit your network to branch.
- Having more than one route between two points is always good. It makes the link more resilient to attack, and it gives a lower resistance, so more steam can flow.
- Things that are connected to the city are repaired, upgraded or built in a breadth-first order. That means that the things that are closest to the city (on the network) will always be done first. You can use this property to set up a build order.
- There's no way to prevent attacks from the environment, except in Peaceful mode. In other game modes you must defend yourself by improving your network.

Changelog

Version 1.5.0

Compatibility improvements for modern PCs with support for high resolutions and arbitrary scaling. There is a new test suite, the code is now Python 3 with type annotations, and Pygame 2 is supported. Some graphics and sounds have been regenerated from lossless originals. Within the game, a "Fast Forward" button has been added.

Note: Version 1.5.0 is not savegame compatible with earlier versions due to major internal changes.

Version 1.4

Various bug fixes, better support for Debian, and new graphics created by Cris Grada. This update also added "Peaceful mode" where your network-building activities are undisturbed by the environment. Sound can be muted in the game.

Version 1.3a

Replaced some sound effects with new versions.

Version 1.3

Repackaged the game for Linux distributions.

Version 1.2a

Bug fix for 'SDL_ttf render failed'. Moved savegames and configuration data to the home directory (Linux) or APPDATA (Windows).

Version 1.2

Pygame.org release. Various presentation tweaks.

Version 1.1a

Fixed some Pygame version detection bugs.

Version 1.1

First version to show the flow of steam through the pipes! Many other minor improvements.

Version 1.0

Version made for Pyweek 2 in March 2006.

Legal stuff

License

Files that are (c) 2006-21 by Jack Whitham:

./manual.pdf

./lib20k/*

./data/*_ogg except earthquake_ogg

./data/32_png

./data/bolt_png

./data/header_jpg

./data/headersm_jpg

./data/mainmenu_jpg

./data/letters_png

./data/stormsample_png

Files that are copyright (c) 2003 by Bitstream, Inc.:

./data/Vera_ttf

Files that are copyright (c) by NASA:

./data/back_jpg

A picture of the moon taken from Apollo 15 in 1971, the original image is labelled “AS15-96-13085”. http://www.apolloexplorer.co.uk/hires/lunar_surface/default.htm.

Files that are copyright (c) Accustudio:

./data/006metal_jpg

./data/greenrust_jpg

./data/rivets_jpg

(source: http://www.accustudio.com/)

Files that are copyright (C) 1998-2007 Oliver Rauch

./data/menuicon_png

(source: http://www.xsane.org/)

Files that come from public domain sound archives and are not copyrighted:

./audio/earthquake_ogg

(source: http://simplythebest.net/sounds/WAV/sound_effects_WAV/nature_wavs.html)

Files that are copyright (C) 2011 Cris Grada

./data/bricks_png

./data/bricks2_png

./data/city1_png

./data/destroy_png

./data/maker_png

./data/maker_u_png

./data/node_png

./data/node_u_png

./data/upgrade_png

./data/well_png

Credits

20KLY has benefited from the contributions of the following people:

- Jillian Whitham helped with many, many things from 2002 onwards: including some early playtesting!
- Cris Grada drew up some new graphics for the game which appeared in version 1.4.
- Siegfried-A. Gevatter, who created the Debian/Ubuntu package, wrote the man page, and contributed code to make the game suitable for including in these distributions.
- Tom Dalton, who suggested the steam flow visualisation and a number of other improvements.
- Matthew Marshall, who contributed a patch for a different version of the same visualisation feature.
- Alexandru Toma, Terrence Brannon, Andrew Wilkinson, Neil Mitchell and James Wells tested early versions of the game and provided bug reports.
- Ivo Anjo, Enrico Zini, Denilson Sa, Matthew Scheffel and Jan Niemeyer sent in some early feature suggestions.
- Josh Holland, who suggested peaceful mode and provided a patch.
- Richard Jones, who organised the PyWeek competition that led to 20KLY's development.
- Andreas Lochmann encouraged me to start the process of updating the game for newer technology, leading to the 15th anniversary edition.
- Scott Johnson created the Github repository for 20KLY and worked independently on various improvements.

- Federico Ceratto made a fork of the game with many new features, which can be found on Github at https://github.com/FedericoCeratto/lightyears.
- David M. Marti, Anne Archibald, Christoph Lucking and Steve Cotton submitted patches for the game.
- Many other people who sent supportive and encouraging messages over the years.

The 15th anniversary edition was an opportunity to integrate some of the improvements contributed by others over the years, but I decided to focus instead on making the original game technically better. That is, I picked “restoration” over “remake”. Improvements can be subjective, and my idea of a “good” change could take away something that you liked. Some of the ideas and patches contributed were really excellent, but I didn’t want to start down that path, because I knew that I wouldn’t be able to stop. In gameplay terms, version 1.5.0 is almost identical to 1.4, there are regression tests for this, and the steam model’s “bugs” are now “features”.