1) Introduction

This document aims to outline the current state of Pixel Vision 8's business and share our strategic priorities for 2020. Our mission is to work with developers interested in creating retro-looking 8-bit games by connecting with them from the moment they get their first idea through completing a project. PV8 lives in a subset of the game development tool space known as Fantasy Consoles (see appendix 7.1). These Fantasy Consoles are small, self-contained, dedicated game development environments with built-in tools reminiscent of old 8-bit consoles or computers. While we continue to execute via our website, itch.io, Twitter, and Discord (see appendix 7.2), we still need to amplify our message to reach other developers interested in using a Fantasy Console.

Early on, PV8 gained attention from developers as an alternative to existing Fantasy Consoles due to a few key distinguishers. These include the ability to customize project limitations, a rich graphic UI (user interface), and the project is open source. Most of the competing Fantasy Consoles use some form of command-line interface for working with the underlying file system and tools. These Fantasy Consoles emulate early DOS and Commodore 64 style operating systems from the late 70s. We purposely modeled PV8 around a more user-friendly operating system akin to the first Macintosh computers of the mid-80s to increase adoption among millennials.

Given the recent crowding of this space, we need to reduce the time it takes to get started, unblock technical hurdles, and offer a clear path to export stand-alone Windows, Mac, and Linux games from PV8 to distinguish ourselves from the competition. By removing developer friction, we hope to increase organic adoption and, in turn, attract developers that produce higher quality games. Our focus in 2020 will help establish PV8 as the number one Fantasy Console developers want to use, ensuring a healthy pipeline of completed games as we head into 2021.

2) Goals

In 2020, we plan to focus on and achieve the following goals:

- **Increase stability**: Reduce the total number of active Github issues from 100 on 12/31/2019 to 25 by 12/31/2020, a decrease of 75% YoY.
- **Increase active users:** Grow the active developer community from 7.4k on 12/31/2019 to 20k by 12/31/2020, an increase of 63% YoY.
- **Increase game releases**: Create additional tutorials, documentation, and code examples that enable developers to create new games from 5 on 12/31/2019 to 20 by 12/31/2020, +75% YoY.

3) Tenets

The following tenets are guiding principles we use to evaluate and prioritize Pixel Vision 8 activities:

- **Quality over quantity:** We will not rush out updates on a set schedule. Instead, we will work towards more significant releases that fix multiple bugs and improve the overall stability of the tooling.
- **Keep scale in mind:** While we'd like to develop individual relationships with each user, we should focus on features and marketing programs that address the top of the funnel to increase developer adoption at scale.
- **Reduce developer friction:** We want to focus on educating developers through in-depth technical content and tool documentation instead of relying solely on community knowledge transfer.
- **Game first approach**: Developers should have a clear publishing path, and our tools need to assist in making this process as easy and intuitive as possible.
- **Play with developers who play with us:** We will prioritize relationships with developers who are more willing to support PV8 and collaborate. To have long-term success, we believe mutual interests will be the most sustainable.
- **Organic trumps paid:** Although some kind of paid promotions will be critical for building and accelerating adoption, we want to focus on driving organic demand through community building and not on paid user acquisition.

4) State of the Business

Pixel Vision 8 has three primary verticals that contribute to the business: the framework, Pixel Vision OS, and technical content. The framework represents the open-source C# codebase that runs the Fantasy Console itself on Windows, Mac, and Linux. This open-source project lives on GitHub and is licensed under the Microsoft Public License (see appendix 7.3). Also, PV8 uses a customized version of MonoGame, a popular open-source game engine under the same license. Pixel Vision OS consists of a custom-built operating system written in Lua and runs on top of the framework as a stand-alone application. The OS includes all of the tools that developers use to make games as well as a way to manage project files and export finished games. Finally, the technical content includes all of the documentation, tutorials, and code examples that not only help on board developers, but also generate the primary source of income to support the project.

In 2019, 66% of our total income came from itch.io, with the remaining 34% generated from the main website through direct sales. These two revenue sources accounted for $3.7k in gross revenue, thanks to the addition of 786 new paying customers (see appendix 7.4). While this covered operational costs (see appendix 7.5), it stunted the ability to scale up development verses the time put into supporting and building new features (see appendix 7.6). This data also points to a failure with the subscription business model since customers favored one time purchases instead of smaller monthly payments. Over time, the lack of subscriptions will impact the recurring revenue potential in 2020. In order to offset the lack of subscriptions, we will instead focus on ways to increase direct sales through complementary content such as tutorials and art packs to make the core product free to grow the user base. Finally, we look to expand alternative income opportunities such as itch.io's "pay what you want" feature and Github sponsorships to make the base product free.

We've continued to have success year over year in building an active developer community across Twitter and Discord. The long term goal for this community is to amplify awareness of the product and increase the number of games created with PV8. There are less than six live titles created out of a total of 5.6k customers (appendix 7.7). As we shift priorities to attract more developers, we will need to invest in continuing to fix bugs and add features that deliver more stability to the product. There are currently a total of 100 open issues representing bugs, features, or improvements to the open-source part of the project. This number is not entirely accurate due to the migration and consolidation of multiple Github repos that represented the entire Pixel Vision codebase. It doesn't take into account issues that were insignificant enough to move over into the current project.

The continual refactoring of the underlying code, maintaining the tools, and inadvertently breaking backward compatibility from release to release helped contribute to the low number of completed games when compared against the other two competing Fantasy Consoles: Pico 8 and Tic-80. On the other hand, substantial investment in the design and UX of Pixel Vision OS has built significant brand awareness, helping keep PV8 in the top three Fantasy Consoles developers look to use (see appendix 7.8). Our focus has been on continuing to add new users while increasing the percentage of completed games. We have made progress by improving the seven built-in tools that support the following functionality: editing text, modifying colors, creating graphics, building maps, generating sound effects, authoring music, and managing the underlying file system. With the majority of users requesting better built-in tools, it became critical that finishing these should be our main priority.

In addition to stability fixes and enhancements to the built-in tools, we have helped generate organic adoption of PV8 through additional technical and educational content. The extensive amount of documentation that currently exists has appeared to be a key distinguisher from other Fantasy Consoles that rely on community knowledge transfer. An early investment in building automation scripts on top of Google Docs to generate these files and tutorials continues to pay off. By owning the primary source of on boarding content, we are in a unique position to not only offer the application and tools themselves, but be the single source of truth when it comes to learning how to use PV8.

As we strive to unblock existing users that make games on top of our platform, we continue to be on the top of developers' minds looking to learn a new Fantasy Console. Each interaction point, whether it be marketing material on the website, source code on Github, in-depth technical documentation, or direct communication on the Discord channel, represents an opportunity to interact with developers from the beginning of their development to the final goal; successfully building a PV8 game. Only by engaging with developers earlier in the building process will we be able to increase adoption.

5) 2019 Lessons Learned

2019 was a successful year for increasing paid and non-paying customers, product stability, and reducing dependence on direct one on one customer support. As of December 31, 2019, we had a total of 7.4k customers. Our unique feature set helped contribute to this growth. Working with a well-known pixel artist, Christina Antoinette Neofotistou (@castpixel), also helped build additional awareness and credibility in the indie game developer community.

One challenge we faced last year was cultivating more developers to create games. Although there has been a great deal of interest on Twitter, very few games have been created by the community in 2019. As a result, developers are not widely using Pixel Vision 8, which continued to be a problem in activating the community last year. To achieve the 2019 goal of adding 1k new users by EOY, we decided to focus more on features, and the type of content developers asked for, instead of directly pursuing customer acquisition. Since there were substantial changes between releases in 2019, this required significant rewrites and editing, which proved to be time-consuming.

The current documentation and the breath of the engine it covers have been one area where developers see the most value. Having useful onboarding documentation had a direct effect on our hands-off education approach based on historical data collected over the previous four years. While adoption continued to grow in 2019, especially after breaking the product up into a free and pro version, the former has attracted new/inexperienced developers that become confused between the two versions. The additional complexity of our offerings made new customers hesitant to pay for a pro license.

As the user base grew, and developers learned how to use PV8, more advanced and sophisticated experiments had arisen. In 2019, we worked directly with a handful of developers by identifying which projects would be prime candidates for tutorials. We were able to leverage our pre-existing tutorial generation tools to create the new content without relying on the influencers to write it manually. These new user-generated tutorials inadvertently gave birth to a unique collection of demoscene tutorials (see appendix 7.9). They offered additional content outside of the standard documentation for developers to learn how to use PV8. Finally, we were able to work directly with these developers without licensing or paying for the content in exchange for exposure to the broader developer community.

Games that are built by developers using PV8 are typically more complex than competing Fantasy Consoles. Developers can code PV8 games in C# or Lua, which inadvertently forces developers to pick one or the other. For developers just getting started, there is little to no shared workflow between the two languages. Being able to fully customize the project's limitations, which is a key distinguishing feature of PV8, has also given way for developers to work on larger, more ambitious projects than our competition. Unfortunately, multiple barriers for developers using our framework lead them to prioritize our competition, Pico8 and Tic-80, over PV8. The same challenge our developers faced externally became a maintenance issue internally. We spent a considerable amount of time on creating feature parity between the two different language options used to develop PV8 games, which distracted us from fixing existing issues and cleaning up the actual tools themselves.

While Pico8 represents the source of the bulk of Fantasy Console games created in 2019, developers that looked to work outside of its predefined project limitations present a real challenge for them. Our last attempt in 2017 to engage with Pico8 developers by creating a custom project template that mapped Pico8 APIs to PV8 ones via a shim ended with an inability to offer a direct migration path due to the differences in how the underlying engines work. Pico8 is estimated to be the number one Fantasy Console with a market lead that will be difficult, if not impossible, to surpass. Tic-80, which started development roughly around the same time as PV8, is growing more popular but hasn't been able to surpass Pico8's market share. On a positive note, PV8 continued to gain popularity in 2019. Using Twitter followers as a point of reference, PV8 had roughly double the followers that Tic-80 did, and the dedicated Discord channel continues to attract new developers interested in PV8.

As we head into 2020, the biggest lesson learned was that without more people using Pixel Vision 8 and the tools, it's challenging to figure out what features are essential and what bugs are real blockers. Also, the smaller community is not producing enough games to help elevate PV8 above the other Fantasy Consoles. We believe that by increasing the number of developers, we will have a better chance of producing more games. If 2019 taught us anything, it's that without developers making PV8 games, the likelihood of getting new developers on board will continue to be a challenge.

6) Strategic Priorities

In 2020, our priorities are to reduce developer friction, improve stability, and encourage high-quality games to be created with Pixel Vision 8. To do this, we will work to enhance technical content, discoverability in the broader Fantasy Console community, and explore new marketing opportunities in a crowded space. However, we still control our destiny, and as such, we will leverage our channels to improve awareness around building games with PV8. Where we are unable to reduce friction or increase growth organically, we will work to achieve the following strategic priorities:

6.1) Focus on increasing stability of the product by reducing open issues below -25, -75% YoY.

Finish Pixel Vision OS tools: Currently, the Pixel Vision OS tools are in varying states of completion. Each one needs to be audited, overhauled, and templatized, so consistency exists across the board. That requires the standardization of a copy/paste and undo/redo system, easier to use UI components and a formal workflow for converting designs into actual working tools. Not only is this critical for our developers to have the best tools possible, but it also sets us up for success. As we head into 2021, we will begin to look into creating new tools to handle more advanced tasks like building meta-sprites, animations and sharing games online via the website's backend.

Refactor existing codebase: We plan to focus solely on refactoring the codebase and reducing the number of issues until the end of the year before adding new features or tools to the OS. There are currently +100 open issues from the underlying C# codebase to the built-in tools. Also, we will begin to work on feature parity between the C# and Lua game development experience. Currently, making PV8 games is easier with Lua due to a lack of features not available in the C# framework, such as the components. Not only will we build upon the same C# engine that powers the Lua development workflow, but we will also improve the existing C# workflow to make it as easy as possible for developers to publish games using either approach. Finally, we will phase out support for legacy APIs and focus on making the framework better at the expense of backward compatibility. While this may force developers to migrate their games when a new version is released, given the low number of projects created, we can use this opportunity to capture a new audience of developers that build better quality games on top of the new APIs

Optimize renderer for lower-powered devices: For the past few years, PV8 has collected substantial technical debt due to the change from Unity to MonoGame. A lot of the code works but is not optimized for lower-powered devices such as Mobile and SOC devices such as the Raspberry Pi. We will spend the time required to not only rewrite the renderer for optimal performance but use the opportunity to simplify some of the more challenging concepts developers continue to struggle to learn. Due to the generic nature of the color configuration and image parsing logic, there are areas in the workflow that need improvement for developers coming from other game engines or Fantasy Consoles to understand how to work with PV8 intuitively.

6.2) Reduce blockers in order to generate 20 published games, +75% YoY

Working closer with developers: In 2020, we will focus on three key areas: co-marketing, publishing, and owning the developer message for our platform. We aim to create a clear publishing path for PV8 developers willing and able to build their games on our framework. While we will also encourage developers to create new games from scratch, there is a value in learning the pain points of developers coming from other platforms that can inform us of ways to improve the platform. To this end, we should prioritize working with game developers looking to port existing games over since they can give us the best information on ways to improve PV8.

Co-marketing: For 2020, we will place a greater emphasis on showcasing developers' work to highlight these success stories within the developer community. The best way to inspire others is to lead by example of what the platform is capable of. This will also require updating all of the existing showcase games to make sure they work on the latest versions of PV8. Maintaining these games are critical for those developers looking for a reference project when getting started. Ideally, we'd look to partner with influencers to help us reach a broader audience with their games too. By leveraging our marketing channels to showcase game case studies and community created technical content, we can give new developers the tools and inspiration they need to be successful. This has proven to be an important tactic when working with influencers to help build PV8 and directly attributed to its early awareness so we can shift this strategy to work with influential developers building games with PV8.

Clear publishing path: While our number one goal with PV8 is to be part of a developer's natural creation process, this may require additional knowledge we can only learn from those developers themselves. Ideally, we want to have an easy-to-use publish mechanism built-into Pixel Vision OS. Also, this process should be in charge of optimizing performance when running the final game on Windows, Mac, and Linux. This feature is critical in building awareness with audiences that are not familiar with our platform since each exported game has a Pixel Vision 8 bootup screen to help promote the Fantasy Console. While the current version of PV8 has made significant progress in optimizing for desktop, we need to focus on making it more apparent to our developers what additional steps are required to create an optimal experience for their games. This will help with the planned migration of the codebase to support Raspberry Pi and other SOC ARM-based computers. Supporting Pi is a highly requested feature that can unlock new revenue in 2021 as we build the foundation to design and sell a self-contained PV8 computer via Kickstarter.

Increase submission from existing user base: Activating developers that use PV8 requires a more hands-on approach. We must address the needs of developers that use our tools. This usually includes not having access to the necessary documentation, code examples, or us not being connected with the larger Fantasy Console communities. To do this, we will focus our energy primarily on reducing friction for developers actively sharing their work publically. A successful channel program for these types of developers will include creating new support documentation outlining how to get a game over the finish line. Once developers understand the publishing workflow, we can offer additional incentives per submission like co-marketing, promotion across our marketing channels, and including their game as a direct download with the application from the main website. Since most game developers do not create apps in bulk, we will need to run on-going campaigns to promote publishing PV8 games to entice additional developers throughout 2020.

We will start by running quarterly promotions to incentivize submissions with free codes to the Pro art packs, Amazon gift cards, and running a game jam before the end of the year. Each promotion should increase completed games and identify marketing opportunities to promote developers who take part. We will also strategically take part in online events that give us access to new developer communities we previously didn't have access to, such as game jams and online conferences. As we engage these tail developers, we can better identify additional opportunities for case studies, developer interviews, or valuable feedback to make working with PV8 easier over time. We will also look for ways to maximize the investments we have made with incentivizing developers to have them also promote and market their game under the PV8 category on Itch.io.

Private publishing incentives: Our plan for working directly with influencers is to run private, invite-only, submission promotions twice before the end of the year. We will define the requirements to participate in these promotions based on vetted higher quality games from our list of known active projects. This promotion aims to incentivize new developers to submit at least one game for a $50-$300 reward (Amazon gift card or PayPal transfer) to achieve <$100 CPG (cost per game) while increasing the catalog of games created with PV8. By running promotions targeting specific developers, we hope to increase overall selection from these influencers without additional marketing budget investment. We should always prioritize getting quality games on the platform instead of paying for user acquisition. We plan to budget at least $1k for this initiative to yield at least ten games.

6.3) Identify and build key features that help increase adoption by 11.5k new users, +58% YoY.

Create a getting started guide: In 2020, we will focus on creating a standardized developer promotion to increase selection from active developers in our community. While we already have some form of getting started material, we must continue to lower the barrier for new developers. The initial documentation that developers read should cover the basics of using Pixel Vision OS, the built-in tools, and walk them through a single game code example from beginning to end. Ideally, we'd like to get new developers up and running in a matter of hours and have them build their first hello world project shortly afterward.

Own our developer message: Regardless of the partnership we strike with influential developers, our best strategy is going to be targeting potential developers through blog posts, online training, and documentation. While event sponsorship is a valuable option when PV8 begins to generate more income, we need to be strategic in how we activate new developers that publish PV8 games. To that end, we must own the developer message. We need to produce marketing material that engages developers and shows them why they should build their next game with PV8. Where influencers are unable to or unwilling to educate their followers about PV8, we need to identify the knowledge gaps and fill them in ourselves. We should target Fantasy Console user groups on our own and present our material to those groups to help win developer support. Finally, we must actively target these communities with our value proposition and incentivise these developers like we would any existing user.

Update our SDK and technical documentation: In addition to redesigning our website to better organize the products we offer, we will need to do a full audit of our existing technical documentation and SDK support based on changes to the underlying framework done in 2020. Our goal is to streamline this content by making sure all of our documentation is updated and reflects the best practices for making PV8 games. Also, we will ensure that we do not favor Lua games over C# games and prioritize parity between the underlying API. By removing legacy code documentation, we will make it clearer to the developer what to implement and reduce additional friction. Correctly covering the full APIs with documentation directly impacts a developer's success and minimizes the time we spend on helping answer questions.

6.4) Increase awareness across all channels through weekly updates +100% YoY

Consolidate messaging across website, Itch.io, and Medium: Historically we have relied on using Podia, the company we currently host our site on, to publish developer articles, product updates, and release announcements. While this has been helpful, messaging has been scattershot between this site, Twitter, and the Discord channel. Also, updates have not been consistent, and sometimes months go by without informing users of what developments are going on. Only posting content to a single channel alone is not enough to inspire developers to use PV8. We need to aggressively market our value proposition and provide the more in-depth technical content they require to successfully publish PV8 games. In 2020, we will redesign our current website to become a developer destination that teaches how to create games with our framework and post them online for others to play.

To achieve this, we will start by building non-technical content on our Medium publication in hopes of attracting a broader audience. From there, we will continue to post work in progress updates to Twitter. In Discord, we will carry on conversations about the work being done to the engine and helping developers with issues. To help sum up all of these activities, we will start to send out weekly updates to keep developers engaged in between longer release cycles. And finally, we will begin using Itch.io's tools to deliver updates to those users that don't directly get PV8 from the leading site. Since we do not force Itch.io users to create an account when downloading from Itch.io, they are required to discover our additional channels. This will also allow us to work around a current gap in our marketing to Itch.io customers since we don't account for free downloads in our reporting. By targeting Itch.io users more, we'll have an easy way to reach them via the content we are already producing on other channels and can build campaigns around converting no-touch free download users into registered ones on the main site.

In addition to a complete redesign of the Podia website, we will better highlight key framework value props and the up-front steps needed to complete a PV8 game. Outside of feature-specific landing pages, we will attempt to better surface-related blog posts, integration documentation, and case studies to remove experienced developers' friction. This new redesign will clearly outline our value proposition to the developer using PV8 with specific callouts highlighting the quickest way to get their game ideas up and running. For example, the current home page only showcases a few features of PV8 and not any of the other services and products we offer to help developers get started. Below this existing section that drives free downloads, we will focus on showcasing and highlighting great content built with PV8. Each dedicated framework feature will get similar treatment.

Showcase the best content from current users: We will focus on promoting top content created with PV8 to help drive awareness of the opportunity the platform presents to developers. In addition to merely offering help to developers making games, we should strive to highlight developers on our platform based on the complexity of the game they choose to build. Our first task will be to source developers working on games that are currently active in the Discord WIP (work in progress) channel. The second task is to find and entice top unmanaged tail developers to port their games to PV8 for additional marketing exposure via all of our marketing channels. This initiative also supports our commitment to offering developers incentives our competition cannot match. We will provide these developers a more curated marketing opportunity such as features on the website, interviews/case studies, and game promotion in each of our weekly updates. The goal is to show influential developers that partnering with PV8 is more than just picking our Fantasy Console over another, but that we are here to actively support and encourage their creations.

The best way for us to succeed is by identifying lighthouse games, which we can use to attract other developers to the platform. These types of games will be the focal point of what we will promote on the website. To better promote each lighthouse game, we will do case studies, technical blog posts, and prominently highlight them on our social media channels. Our goal with supporting these developers is also to get exclusive access to their feedback on using the tools, securing the game itself on our platform, and remove any barriers the developers may have when building with PV8. Finally, though these partnerships, the most valuable feedback we get can be used to improve Pixel Vision 8 in 2020 as we prepare to head into 2021.

7) Appendix

7.1) List of known fantasy consoles in 2020 from https://paladin-t.github.io/fantasy/

FC Name (# - M)	Language	Price & License	Platform	Display
1Bit-Wonder by Brastin	Instructions	Free	Windows, Linux	225x125
BASIC8 by Tony Wang	BASIC	$14.99	Windows, macOS, Linux	160x128
Bitsy by Adam Le Doux	Instructions	Free, MIT	Browser	16x16 1bit sprites (8x8 per sprite)
CHROMA-60 by Arkia	ASM	Free	Windows, Linux	240x135
Click4 by Josef Patoprsty	ASM	Free, MIT	Windows, macOS, Linux	64x64
CToy by Anaël Seghezzi	C	Free, zlib	Windows, macOS, Linux	128x128
DRAK-0 by Drew Wibbenmeyer	ChaiScript, LuaJIT (coming soon)	NYOP, MIT	Windows	320x240
DX8 by Robin Southern	ASM	Free, MIT	Windows	320x256
ECoS by MLJWare	Lua	NYOP	Windows, macOS, Linux	128x128
ESP LGE by Igor(corax89)	C like	Free, GPLv3	Browser, custom handheld	128x128
Fancade by Martin Magni	Visual Scripting	F2P	iOS, Android	Voxel
FAZIC by Michał Kalbarczyk	BASIC	Free	Browser	320x240
G-eon by Memorix101	JavaScript	NYOP	Windows, Linux, Raspberry Pi	640x480
Homegirl by poeticAndroid	Lua	Free, Pro, MIT	Windows, Linux, MacOS	Screen modes from 80x45 to 640x480, up to 256 colors from 12bit palette
IBNIZ by viznut	Instructions	Free, zlib	Windows, Linux	256x256
Leikr by Torbuntu	(Official) Groovy, (unofficial) Java, Kotlin	Free, Apache	Linux, Raspberry Pi 3B/3B+/3A+, ClockworkPi GameShell, Windows	240x160
LIKO-12 by RamiLego4Game	Lua	NYOP, MIT	Windows, macOS, Linux, Android	192x128
LowRes Coder by Timo Kloss	BASIC	$4.99 (F2P)	iOS	Up to 128x128
LowRes NX by Timo Kloss	BASIC	Free, LGPL3	Windows, macOS, iOS	160x128
LuaG Console by Vulcalien	Lua	Free, Apache	Windows, macOS, Linux	160x160
MakeCode Arcade by Microsoft	Blocks, JavaScript, TypeScript	Free, MIT	Browser, Raspberry Pi 0, custom handhelds	160x120

FC Name (N - Z)	Language	Price & License	Platform	Display
nano Jammer by Casual Effects	nano	Free, BSD	Browser	64x64
NEKO8 by Egor Dorichev	Lua, BASIC, ASM, MoonScript	NYOP, ?	Windows, macOS, Linux, Android	192x128
Nibble by Nibble Team	Lua	Free, GPLv3	Windows, macOS, Linux	400x240 24bit/7bit
Phosphor by Marc Lepage	Lua	Free, MIT	Browser	192x128
PICO-8 by Lexaloffle	Lua	$14.99	Windows, macOS, Linux, Raspbery Pi	128x128 4bit
Pix64 by ZappedCow	PNG	NYOP	Windows, Linux	64x64
Pixel Vision 8 by Pixel Vision 8	Lua	Free (Limited tools), $9.99 (All tools), $1/mo (All versions plus extras)	Windows, macOS, Linux	256x240
Prism-384 by Grapefruitopia Industries	JavaScript	$5	Windows, macOS, Linux, Raspberry Pi	384x216
PuzzleScript, DungeonScript by Stephen Lavelle	Instructions	Free, MIT	Browser	Configurable, (DungeonScript) Voxel
PX8 by hallucino	Lua, Python	NYOP, MIT	Windows, macOS, Linux	Configurable
Pyxel by Takashi Kitao	Python	Free, MIT	Windows, macOS, Linux	256x256
quadplay by Casual Effects	PyxlScript	Free, LGPL3	Windows, macOS, Linux	384x224
Raccoon by Lutopia & Linheha	JavaScript	Free, Unlicense	Browser	128x128
Riko4 by Bryan	Lua	Free, MIT	Windows, macOS, Linux	280x160
SCRIPT-8 by Gabriel Florit	JavaScript	Free, MIT	Browser	128x128
SmileBASIC by SmileBoom	BASIC	$9.99, $24.99	3DS, NS	320x240, 400x240
TIC-80 by Vadim Grigoruk	Lua, MoonScript, JavaScript, Fennel, Wren, Squirrel	Free, $5 (Pro), MIT	Windows, macOS, Linux, Android	240x136
VectorBoy by David Jalbert	Lua	NYOP	Windows	Vector
Voxatron by Lexaloffle	micro-scripting	$19.99	Windows, macOS, Linux	Voxel
VVpet by Gardrek	Lua	Free, ?	Löve	64x64x2, 128x128x4, Configurable
yuki-js by Nik Coughlin	JavaScript (subset)	Free, MIT	Browser	Configurable

7.2) Current list of sites, social networks, and stores Pixel Vision 8 is located on as of 6/15/20:

Site	Users/Followers	URL
Main Site	6,137	https://www.pixelvision8.com/
Itch.io	1265	https://pixelvision8.itch.io/
Twitter	4,006	https://twitter.com/pixelvision8
Discord	655	https://discord.com/invite/pixelvision8
Github	202	https://github.com/PixelVision8

7.3) Microsoft Public License (MS-PL) on Open Source Initiative (https://opensource.org/licenses/MS-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the

same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

7.4) Breakdown of income in 2019

Source	Income
Itch.io	
Direrct Sales	
Subscriptions	
Tutorials	
Artpacks	
Total	0

7.5) Breakdown of expenses in 2019

Expense	Costs
Hosting	
Contractors	
Hardware	
Tools/Software	
Misc. Services	
Total	0

7.6) Development Hours

Month	Hours
January	
February	
March	
April	
May	
June	
July	
August	
September	
October	
November	
December	
Total	0

7.7) Total live games

Game	Creator	Location
Reaper Boy LD 40	Jesse Freeman	https://pixelvision8.itch.io/reaper-boy-ld40
Jump N' Shoot Man	Jesse Freeman	https://pixelvision8.itch.io/jump-n-shoot-man-ld41
Reaper Boy LD 42	Jesse Freeman	https://pixelvision8.itch.io/reaper-boy-ld42
Terminal	Jesse Freeman	https://pixelvision8.itch.io/terminal

7.8) Reviews, rankings, and online publications mentioning PV8

Article	Date	Rank	Author	Link
Fantasy Console Wars: A Guide to The Biggest Players in Retrogaming's Newest Trend	Mar 21, 2017	#4	John King III	https://medium.com/@G05P3L/fantasy-console-wars-a-guide-to-the-biggest-players-in-retrogamings-newest-trend-56bbe948474d
Your field guide to fantasy consoles	July 25, 2017	#2	Spencer Hayes	https://itch.io/blog/5733/your-field-guide-to-fantasy-consoles
Fantasy Console Development	Jun 1, 2017	#3	Jesse Freeman	https://pixelvision8.itch.io/reaper-boy-ld42

7.9) Demo scene tutorials created in 2019

Tutorials	Total	Link
Julio Choy	6	https://www.pixelvision8.com/courses/demoscene/133893-julio-choy/394748-introduction