

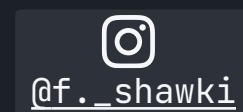
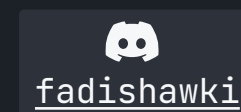
OrbitMines as a Game Project

A comprehensive guide on how to be frustrated with pixels. An open call for funding, collaboration or anyone curious to learn more.



[Fadi Shawki](#)

fadi.shawki@orbitmines.com



22 February 2024



[Discussion Channel](#)

Introduction

I'm finding myself having to write something along these lines more often recently, so I'm not even sure what a thing like this should be called. Perhaps this is a comprehensive guide on how to be frustrated with pixels. Containing a bit of personal history, a bit of future projections and too many vague ideas which need a practical handle - Ah, that'll be the future projection part.

Perhaps this should make communicating these things easier by just being able to share a link. An online copy of this can be found [here](#).

As a follow up of this [post](#) and this [other one](#), this is a letter I've sent to a plentiful of people, which outlines the category of project OrbitMines will be. Please feel to share this with anyone who could help me make these things happen. As with anything with me, this thing is so flexibly general that it should survive any sudden shift in direction I occasionally tend to make.

Probably more accurately, this is

OrbitMines as a (Programming) Language, Version Control, Compiler, Browser, Integrated Development Environment (IDE), Operating System, ..., Game Project

. Though "OrbitMines as a Game Project" is probably more provocative and more accurately represents the most important part of this project: its (visual) interface.

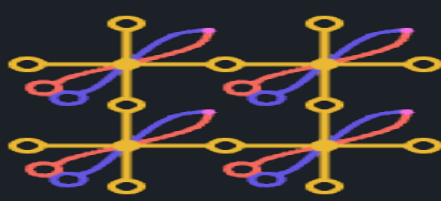
The very example that all those things are so isolated from each-other as concepts itself shows the entirety of this complex problem. But for a lot of good practical reasons, it is quite understandable that such a thing happens. Simplifying, - compression -, is hard. Exploring, - finding new things -, is hard.

A broader interpretation of this approach is probably more along the lines of instead of attacking any single problem directly, or to solve any definite problem. To instead assume we can't actually generally do that, and to find tools as generally as possible that can be applied as conveniently as possible. I don't care how its details work, what can you **do with it**? But not only that, it needs to be satisfying to learn, to explore. The only example I know of which has achieved this for a general audience are video games. Specifically for this category of problem, sandbox video games.

Ok, enough abstract vagueness without content, onward! ... to less abstract vagueness:

This problem, though filled with incredible technical complexity, in its essence is a visual design problem: What should it feel like? What should it look like?

That makes our lives in a certain respect much easier. Because it's definitely not remotely like anything I've seen before.



Technical Complexity

Though all that might sound interesting, how do you *actually* make that happen? A question which has been digging at me for a long time.

This starts with a few complicated but simple ideas:

See: [Indexing existing Abstract Models \(2024-2025?\)](#)

- Whatever it is (programming) languages are currently doing, they will never get there: This will **never** be possible in *just* the textual interface any programmer is familiar with.
- For some reason, it's incredibly hard to apply the mode of thinking which allows for the creation in, ... of a programming language, to the interface in which one programs.
- It is incredibly hard to throw away all historical context and actually start from something **new**, while still having a practical handle on anything others have done before. Try doing that, while everyone is doing that.

See: [OrbitMines as Game Project \(2025/2026?\)](#)

- Whatever this interface will look like, it necessitates conveniently being able to change **anything** about how it looks, ..., how it operates *from the interface itself*. This introduces a hard problem on the side of the implementer: How do you possibly account for that? Or perhaps: Why is that a problem in the first case?
- Whatever function it is that platforms and interfaces serve, they will probably converge to being more of a theme applied on a particular type of structure. Only as a supply of resources (access to certain kinds of information/compute) will they persist. They will not persist as separable interfaces.
- Whatever sets up this open world generation must rely on existing structure, information. You can set up something more random, sure. But the only possibility of some feedback on this generation must be some generalization of existing knowledge. Whatever (entropically relevant) information, structure, ..., items have been found.
- This means open world generation must support arbitrary information which will **not** be available in the design of this game/interface.

See: [On Orbits, Equivalence and Inconsistency](#) (Hypergraphic) Version Control System through Rays

Though I've not yet gone into a proper research trajectory towards designs. It will come down to generalizations of the patterns found in the following things:

- In broad terms how this game should initially feel: It looks like you're playing something like Minecraft. You might not even realize it can be used as a tool. Then suddenly comes the realization that it can be used to do/create *anything*.
 - It should be seamless, you shouldn't even notice that certain things you're doing could be interpreted as science, ..., engineering. The moment you realize you can, you can tap into that more.
 - The constraining aspects of the game are not necessarily item collection in the usual game sense: Certain kinds of copying are incredibly easy. The limiting factor is finding a particular kind of resource, or what it can be used for.
 - There must be some sense of stability in the interface. Though many interesting things will probably be more visually unstable.
 - Anything that's generated, ..., created, which is accessible, not forgotten and understandable can be visited as a location. This for example includes whatever intro-screen it is that the game has. It can be changed.
 - A big challenge is probably ignoring an existing generated world and instantiating new generation on that same location. Superposed, portalled through, ..., forked.
 - There's something entirely problematic about certain types of convenient solutions. An example of this may be the interface that is the cursor on your computer screen you are familiar with. Though easy to generalize as an interface to any kind of website or application, it heavily steers towards a particular kind of interface which directly goes against what this project is trying to accomplish. Similarly, there's something quite unsatisfying about the keyboard too, though it probably scales better. Its functionality is usually hidden and not easily visualized. It's even harder to ask the question of finding out what possible things you can do with it, let alone to adapt to it yourself: that's hard. A more general pattern along those lines is probably something like this: Once something seemingly convenient is found, it is seriously hard to explore and steer away from that.
- Enter more interesting tactile interfaces.



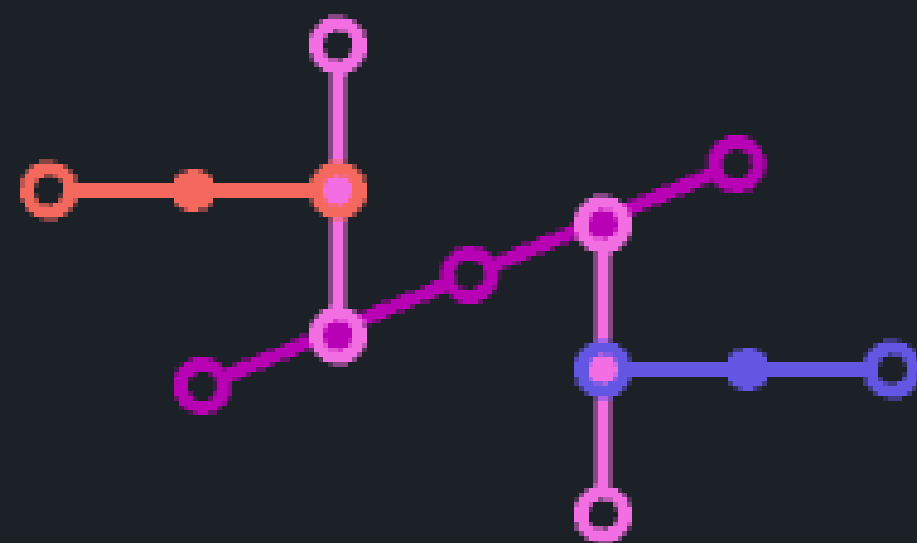
Something about me

Since the target audience for this one will probably be interested in me instead of thinking on these problems: This is making me wish I had already finished my archive project and could just point to [that](#).

I recently was in a call with someone going over some of OrbitMines history so that makes this thing a little easier. Let me just list a bunch of tangents of which only the Minecraft server wasn't a complete disaster:

- 24 October 2011: I login to [DarkOrbit](#) for the first time (Don't remember how we found it).
- 31 January 2013: I buy [Minecraft](#) after seeing [my brother](#) play Minecraft (hunger/survival)games on [MCSG](#) for the first time.
- July?-October 2013: I try to figure out how to make a Minecraft server, inspired by [one I loved](#).
- September?-October 2013: While on the bicycle next to [my brother](#), discussing whether MineOrbit or OrbitMines is the better name for the server. With the idea to make it like [DarkOrbit in Minecraft](#) which we later (2017) dubbed as "Fractals of the Galaxy".
- 15 October 2013: orbitmines.com is registered.
- 15 October 2013 - 11 June 2019: Insert OrbitMines Minecraft history I want to [expand on later](#). Including other Minecraft [disasters](#). Even things along the lines of: Surely you would like to make a Minecraft plugin by decompiling [Java/Ruby](#) and recombining program [ASTs](#).
- 16 October 2018: First exposed to [cellular automata](#) as an exercise given to me by [a company](#) I ended up working at.
- 2020: I stop attending [Leiden University](#). If you could call what I did there as attending in the first place. Perhaps more of an (immature) severe disinterest.
- 2018 - May 2022: I [do work for several companies and attempt to start up several doomed-to-failure project/companies with partners](#).
- 2021: [Dune](#) makes me interested to start reading.
- 2021-2022: I attempt writing compression algorithms with zero knowledge of established fields. That [Hutter Prize](#) sure looks interesting.

- December 2022: I try organizing a year of incredible confusions, reading, ..., listening. Which turned into a dabble too close to being a descent into uselessness. I compiled it into a confusing piece of philosophy if you're interested in that sort of thing: [On the Intelligibility of \(dynamic\) Systems and Conceptual Uncertainty](#)
 - January - July 2023: Confusion turns to [more interesting confusion](#)
 - 26 June 2023: I get [a random notification](#) of this little thing called [Society for Multidisciplinary and Fundamental Research \(SEMF\)](#) organizing a summer school a few weeks later. Where some of the most interesting people I've ever met seemed to be. And evidently a good place for my confusions.
 - December 2023: Again the most interesting confusions of the year compiled in a writing: [On Orbits, Equivalence and Inconsistencies](#), this time in a more acceptable form (for now).
- 22 February 2024: And now we're here. I think I finally understand the quest that is this project which at its center must be the tool with which to find new quests.



[Fadi Shawki](#)

fadi.shawki@orbitmines.com



[FadiShawki](#)



[@_FadiShawki](#)



[fadishawki](#)

Footnotes & References