Starknet's Dojo Engine Poised to Disrupt the Emerging Autonomous Worlds Space

# On-chain gaming on Starknet and Autonomous worlds

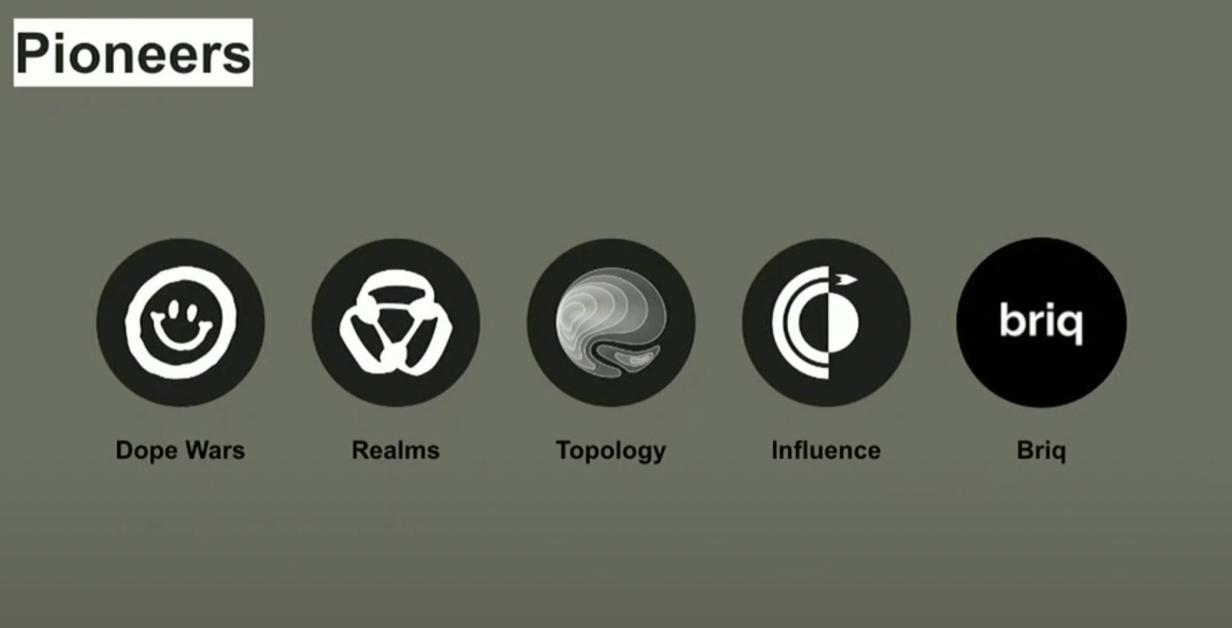
In our recent [exploration](https://starkware.co/resource/dojo-on-starknet-game-on/) of the Dojo gaming engine, we talked about the world’s first provable on-chain game engine that enables transparency, provability, and scalability for the gaming ecosystem on Starknet. While the realm of on-chain gaming is still emerging, the revolutionary advancements in the technology stack for creating games on Starknet have led to the swift development of novel gaming experiences. This opens up new experiences for those who wish to experience a universe that game developers do not strictly dictate.

From a gamer's perspective, there are only two major differentiating factors for on-chain gaming that are not enabled in the traditional gaming universe: **asset ownership** and **limitless interoperability**. Asset ownership signifies that any in-game assets a gamer owns belong to the gamer (the gamer's wallet address, to be more specific). Any assets gamers may win, purchase, or exchange will belong to them no matter what the game's developers decide. Even if the game developers' centralized server goes offline, the player can still use their assets. This seamlessly transitions into the second key advantage of on-chain gaming— interoperability. Because all crucial historical data, including asset ownership transactions, is stored on-chain, gamers can easily transfer characters and assets from one game environment to another. This effectively removes the dependency on a single game or developer, enriching the overall gaming experience by allowing assets and characters to have value and utility across multiple platforms and universes.

**Autonomous worlds** are a crucial infrastructure for on-chain gaming to succeed truly. The key idea behind autonomous worlds is having a universe that operates independently without any authority. These worlds are designed in such a way that independent architectures (not only gaming-related but others as well) can function efficiently without any authority presiding over any of the systems.

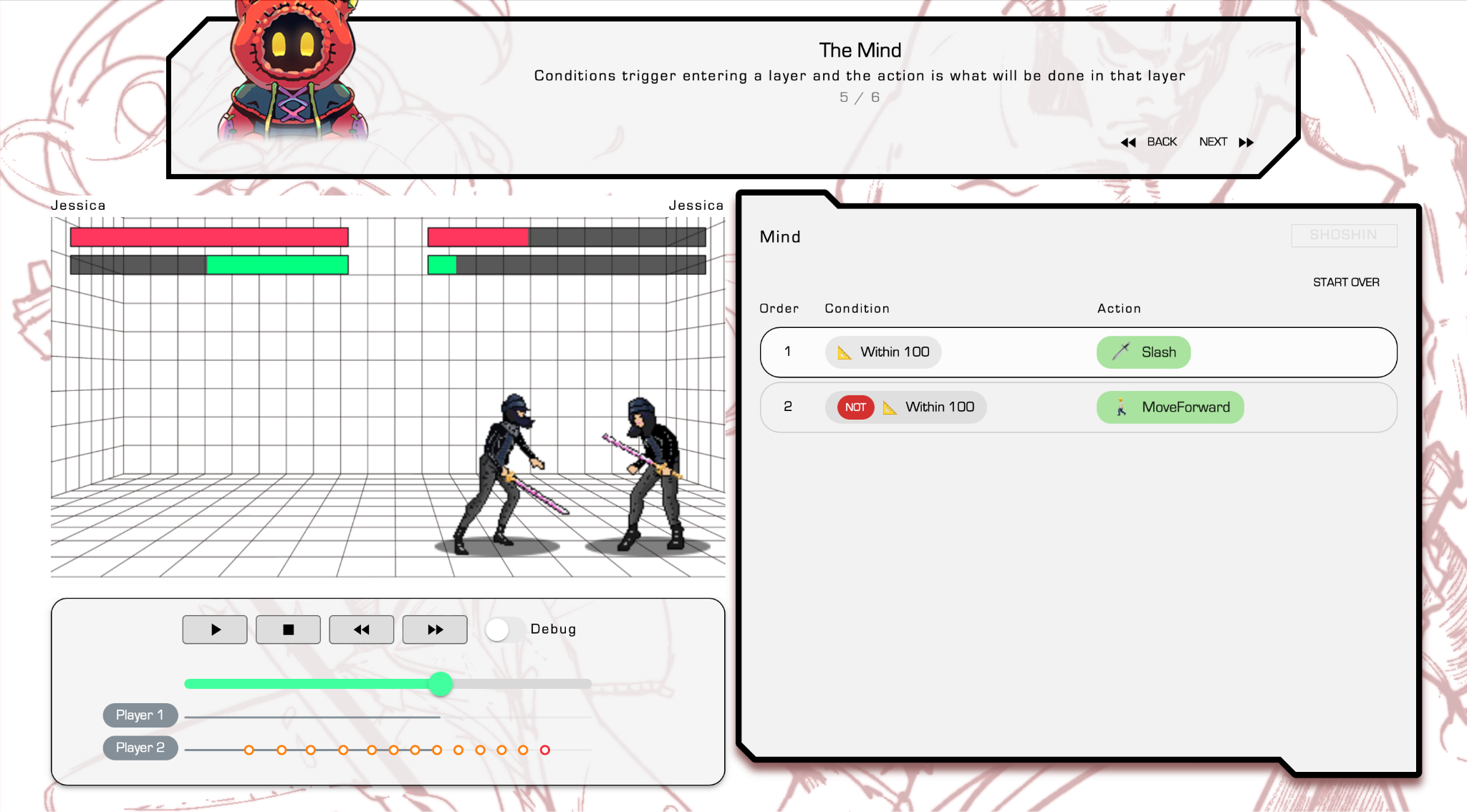
# Early Adopters and Upcoming Projects

Starknet released its Alpha version two years ago, and a mysterious new language with no documentation was released - Cairo. The earliest adopters of Cairo ([Perama](https://twitter.com/eth_worm) and [Guilty](https://twitter.com/guiltygyoza)) started delving deeper into the language. They made valuable contributions so other developers (mostly from Solidity backgrounds) could start writing Cairo. [Realms](https://twitter.com/LootRealms), [Topology](https://twitter.com/topology_gg), [Influence](https://twitter.com/influenceth), and [Briq](https://twitter.com/briqNFT), were among the earliest teams that started working with Cairo.



After years of research and iteration, Starknet has become the first blockchain that can sustainably host high-TPS games. In a [previous article](https://www.starknet.io/en/posts/ecosystem/let-the-games-begin-redefining-onchain-gaming-with-starknet), we looked at some of the largest gaming projects built on Starknet - from space colonization strategy games to 'immutable arcade machines' enabled by ZK circuits.

[Shoshin](https://twitter.com/Shoshin_gg) is a novel way of on-chain gaming where the user has to program their character and the actions they take. Once this programmed logic is in place, players can battle with other players' characters. All in-game mechanics are executed within the Cairo virtual machine. Shoshin even had a recent in-person tournament in Palo Alto for pioneer gamers. To try out the game, log in to [shoshin.gg](https://shoshin.gg/) and unleash your fighting skills by programming the character that no one can beat!



episode from Shoshin of programming character action logic

# What makes on-chain gaming on Starknet possible ?

Among all the L2s that are live today, Starknet has managed to amass the largest number of developers building games on Starknet. Even though this must result from multiple factors at play together, let us decode what makes on-chain gaming and/or game development on Starknet as seamless as possible.

**Madara Sequencer**

The [Madara Sequencer](https://github.com/keep-starknet-strange/madara) is a high-performance Starknet sequencer that can create highly customizable and efficient appchains. Appchains are an L3-like structure built on top of Starknet (an L2 itself) that allows the developers to control virtually all the parameters configured in a blockchain: sequencing, data availability, settlement layer, governance, etc. Madara is built using the proven Substrate framework used by the [Polkadot](https://www.polkadot.network/) ecosystem. Most of the games being built on Starknet already are or have it in their roadmaps to implement L3 networks that they can customize as per their needs. For example, if a certain game wants player transactions to be included as soon as possible, they may implement some form of FCFS sequencing. Alternatively, Priority Gas Auction (PGA) sequencing could be implemented with a more profit-driven perspective to incentivize users to bid higher for quicker block inclusion. With plenty of other conceivable parameters (such as block times, frequency of settlement on Layer 2, or utilization of non-native data availability solutions), the Madara sequencer increases the power of the developers manifold.



**Dojo Gaming engine**

The **Dojo** engine is a software framework for game developers that helps them create fast, provable games on-chain. This engine is the brainchild of early innovators in Starknet-based game development—specifically, a collaborative effort between the Cartridge and Realms teams. Their collaboration was inspired by the insights they gained over a one-year journey that commenced in early 2021, during which they explored the most efficient ways to build games on Starknet. Consequently, they started building the ECS framework for Dojo so that new developers could smoothly transition to the Starknet tech stack. But the offerings of Dojo go beyond just the ECS framework, which we've thoroughly discussed [here](https://starkware.co/resource/dojo-on-starknet-game-on/). Dojo also provides three very useful tools for game developers: Sozu, Torii, and Katana.

**Sozo** is a migration planner that handles the complex task of deploying your autonomous worlds on-chain. With a simple `sozo migrate` command, deploying an instance of the game “world” on-chain is possible. Sozo has the ability for any participant in the ecosystem to propose new components to the on-chain gaming universe by using this simple CLI tool. This is in line with the philosophy of autonomous development of games that live on-chain (possibly outliving the game's creators) since interested contributors can extend the ecosystem by implementing their ideas (such as adding new assets, levels, characters, etc.).

**Katana** is a development sequencer built for local game development. Running the sequencer locally enables immense jumps in productivity. Katana offers RPC methods offered by Starknet on mainnet and allows the developer to test with various parameters such as the block time, base fee per transaction, etc.



Running the node (once configured) is as easy as running the `katana` command on the CLI

**Torii** is an indexing layer built on top of the Dojo engine that connects the on-chain infrastructure with game development clients such as Unity or Unreal Engine. Based on the developed game’s source, Torii can be used to easily start indexing game-specific events and expose a GraphQL API to query them. Simply running `torii` creates a GraphQL API running on [http://localhost:8080](http://localhost:8080/), ready to be queried.

Other than this, extensive research is happening on the Starknet ecosystem to develop off-chain provable games. Not every action that the player takes must be put on-chain. This means that for certain games, where the user’s actions must not be made public before the state of the game changes, an off-chain proof of the user taking a specific action can be generated on the client side, and only the proof would be submitted on-chain. Beyond multiplayer games, this infrastructure holds promise for betting, auctions, and voting systems applications, showcasing its versatile potential in various sectors.

Neural networks have changed the way we think of what the future looks like. Cheap provable computing on Starknet has not left machine learning and artificial intelligence behind. Teams like Gize and Modulus Labs are working on bringing the power of neural networks (and non-neural-network ML algorithms) on-chain. However, how ML algorithms work on Starknet can expand the current use cases.

To understand the new use cases, it's essential to understand that two fundamental components play a role in a trained neural network: weights and inputs. When processed with the weights, the inputs arrive at a certain output that needs to be proven. Utilizing ZK proofs on Starknet makes it feasible to obscure either the weights or the inputs. In certain scenarios, it might be useful to demonstrate that public inputs, when processed through a model with concealed weights, gave a certain output. It would benefit projects that would not want to reveal proprietary weights of their models. Conversely, there might be instances where a user would prefer to keep their input data confidential but still wants to process their inputs through a model with public weights and prove that a certain output was achieved. An example use case for this would be a financial score calculator that gives a certain output by analyzing the user's finances.

# Challenges of on-chain gaming

An apparent question that may arise when talking about on-chain gaming is why haven’t on-chain games yet taken off and delivered the benefits they offer? There might be multiple reasons for this, some of which are tackled by the Starknet ecosystem.

1. *EVM was not built for gaming or processing tons of transactions* that might be less valuable than a transfer of ERC20 tokens. Cairo, in turn, is designed to handle large provable programs whose proofs can be verified in O(logn) time complexity.
2. *Cairo's provability opens the door to limitless innovation*. One such technology is Dojo’s research into client-side proving a part of the transaction. This means that some of the transactions sent on Starknet games will be partly proven on the client side. This would permit games developed on the Dojo platform to obscure some of the data input by the users. A zkSNARK could be generated on the client side to confirm receipt of these obscured inputs. This proof can be forwarded to the network without revealing the secret input. Partial proving on the client side can also help decrease the transaction fees for processing the transaction (since some part of it could be done on the client side).
3. Client-side proving also opens the potential for having a model where gamers try out a *hybrid approach of on-chain gaming* where they only publish proofs whenever something significant happens in-game (e.g., a level is passed or the character finds a rare asset). This allows gamers to experiment and play the game as much as they want without paying for useless “test” transactions when they can't beat the final boss for the 100th time in a game!
4. Spinning up L3 chains for game-specific use cases is easier than ever with Madara. Spinning up app chains for developers using the Substrate framework (that Madara uses) allows them to experiment with the infrastructure without necessarily having to develop a completely new blockchain from its roots.

As we have explored the evolving future of on-chain gaming and autonomous worlds, the advantages of using a validity rollup have become apparent. With these technological advancements, we're not just refining the present ecosystem; we're shaping the future of how games are played, assets are owned, and communities are built.