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

# On-chain gaming on Starknet and Autonomous worlds

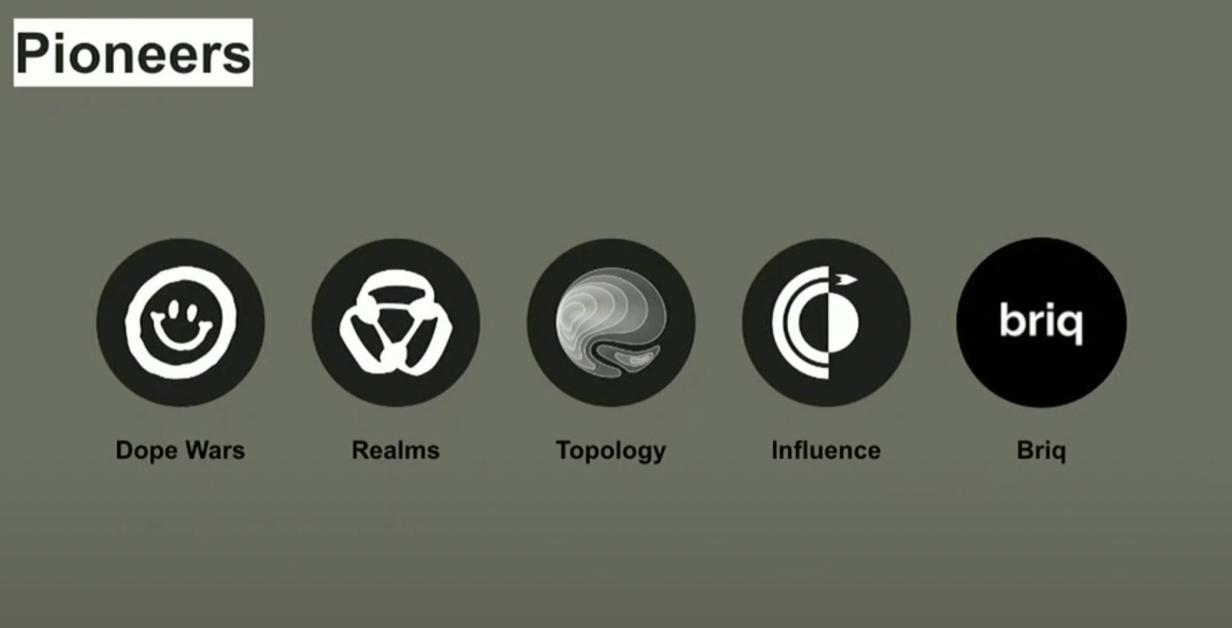
In our recent [exploration](https://starkware.co/resource/dojo-on-starknet-game-on/) of the Dojo gaming engine we talked about the worlds’ first provable on-chain game engine that is enabling transparency, provability and scalability for the gaming ecosystem on Starknet. While the realm of on-chain gaming 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 is not strictly dictated by game developers.

From the perspective of a gamer, 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 that a gamer owns, actually belong to the gamer (the gamer’s wallet address to be more specific). This means that any assets that gamers may win, purchase or exchange will belong to them no matter what is decided by the developers of the game. Even if the centralized server that is run by the game developers goes offline, the player can still make use of 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, this enables gamers to transfer characters and assets from one game environment to another with ease. 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 truly succeed. 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) are able to function efficiently without any authority presiding over any one of the systems.

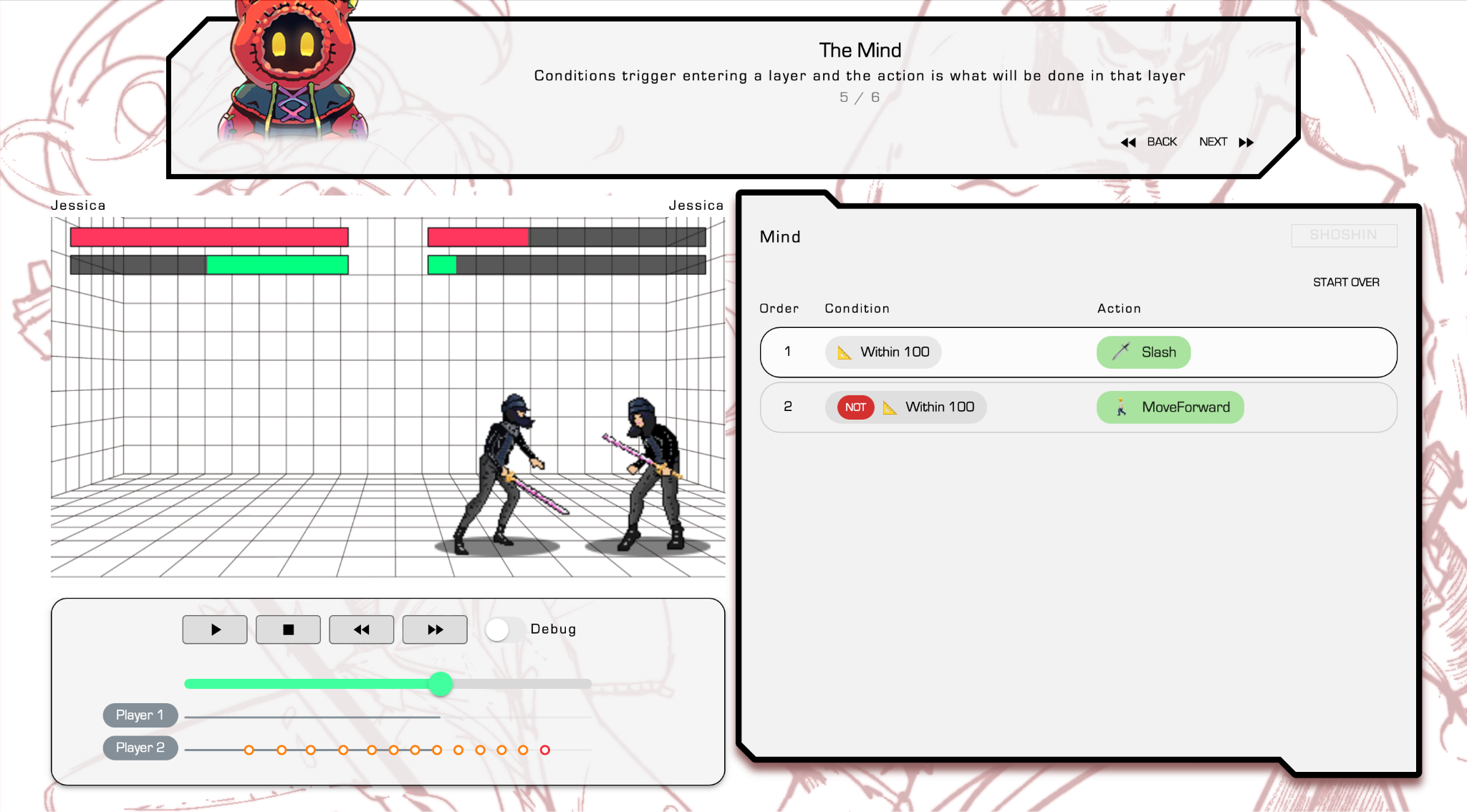
# Early Adopters and Upcoming Projects

Two years ago, Starknet released its Alpha version and a mysterious new language that had 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 and made valuable contributions so that other developers (most of whom came from Solidity background) can 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.



Today, 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 have looked at some of the largest gaming projects that are being built on Starknet - from space colonization strategy games to ‘immutable arcade machines’ enabled by ZK circuits.

[Shoshin](https://twitter.com/Shoshin_gg) is a completely novel way of on-chain gaming where the user has to program their character and the actions that the character takes. Once this programmed logic is in place, players can engage in battles 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. In order to try out the game, login 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 L2’s that are live today, it would be fair to assume Starknet has managed to amass the largest number of developers building games on Starknet. Even though this must be a result fo multiple factors at play together, let us try to 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 provides the ability to create highly customizable and efficient appchains. Appchains are an L3-like structure built on top of Starknet (which is an L2 itself) that allow the developers to take control of virtually all the parameters that can be configured in a blockchain: sequencing, data availability, settlement layer, governance etc. Madara is built using the proven Substrate framework that is also 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 e.g. if a certain game wants player transactions to be included as soon as possible, they may implement some form of FCFS sequencing. Alternatively, with a more profit-driven perspective, Priority Gas Auction (PGA) sequencing could be implemented 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 non-native data availability solutions) Madara sequencer increases the power in hands of the developers manyfold.



**Dojo Gaming engine**

The Dojo engine is a software framework for game developers that helps them create fast provable games on-chain. Intriguingly, 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 can have a smooth 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 very much in line with the philosophy of autonomous development of games that live on-chain (possibly outliving the creators of the game itself) since interested contributors will be able to extend the ecosystem by implementing their own 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> ready to be queried.

Other (Guiltygioza)

Physics engines

Offchain provable games

Neural network

# 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 actually tackled by the Starknet ecosystem.

1. *EVM was not built for gaming or processing tons of information* that might not be as valuable. Cairo, in turn, is designed to handle large provable programs whose proofs can be verified in O(logn) time complexity.
2. *Cairo provability opens the door to limitless innovation*. One of such technologies is Dojo’s research into client-side proving a part of the transaction. This means that of all the transactions sent on Starknet games, some will be partly proved 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 ever having to reveal 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 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 a rare asset is found by the character). This allows for gamers to experiment and play the game as much as they want without having to pay 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 as easy as ever with using Madara. Spinning up appchains for developers using Substrate framework (that Madara uses) allows them to experiment with the infrastructure without necessarily having to develop a complete new blockchain from roots.

As we have been able to explore the evolving future of on-chain gaming and autonomous worlds, the advantages of using a validity rollup 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.