Using Open Games to Model Real-Life Financial Services

Sam Bacha
Manifold Finance
noreply@manifold.com

Fabrizio Genovese
20squares
noreply@20squares.xyz

Philipp Zahn
20squares
noreply@20squares.xyz

In this software demonstration proposal, we will show how we used the Open Game Engine [5] to investigate the market conditions within which users have incentives to use a specific financial primitive.

The Open Game Engine is an implementation in Haskell of Compositional Game Theory [7, 3], a recasting of classical game theory in categorical terms. Whereas games were traditionally thought of as monolithic structures, Compositional Game Theory defines them as open processes that can be combined with each other in multiple ways. In practical applications, this allows for building complex models from simple parts and, most importantly, to make changes to the overall model by just changing some parts while all the rest stays fixed. This gives a noticeable advantage in that model prototypation can be fast and easily adaptable.

In detail, the work we performed went on as follows:

- Manifold Finance [8] is a company operating in the cryptocurrency ecosystem. More specifically, Manifold operates in the MEV [6] and block-building space by providing services to both protocols and their end-users. For instance, they guarantee to a user of their services that their transactions will not be censored, or that any arbitrage profit arising from ordering transactions in a block in a specific way will be redistributed to the end users themselves.

- Manifold has been investigating the idea of offering *future gas price services*. In the Ethereum [4] ecosystem, adding a transaction to the blockchain costs *gas*. Gas represents 'how much computation is needed to include a transaction in a block', and varies proportionally to the complexity of the transaction. So, a transaction may cost e.g. 10000 gas, or 25000 gas. Depending on various factors such as protocol congestion, a single unit of gas will have a volatile *gas price* expressed in *ETH*, the basic Ethereum currency. So, the amount of ETH a user has to pay for a given transaction is

$$\text{gas} \cdot \text{gas price}$$

  As we said gas price fluctuates, and tends to rise with network congestion. Future gas price reservation is a mechanism that allows the end user to reserve some gas, at current gas price, for a future block. Clearly, this is convenient for the user if the gas price rises in the future, whereas it is convenient for the service provider if it drops.

- Tsabary, Manuskin and Eyaler provided a mechanism for on-chain future gas price reservation in their Ledger-Hedger paper [9]. Manifold wanted to use this work as the theoretical basis on which to build a new service they wanted to offer. They tasked 20squares [1], a consulting company specialized in mechanism design, modelling and analysis, to formalize the Ledger-Hedger model to investigate in which market conditions it would make sense, respectively, for a user to use the service, and for the provider (Manifold) to provide it.

- 20squares developed a fully documented, publicly available [2] model of Ledger-Hedger, implemented in Haskell using the 'Open Game Engine' [5].

Submitted to:
ACT 2023

- Analyzing the model, it was found that the *sine qua non* condition for both Buyer (the user using the service) and Seller (the service provider, in this case Manifold Finance) is to be *risk-averse*. This result was already contained in the original Ledger-Hedger paper, but the model allowed to quantify to very granularly how high the service fees had to be with respect to the risk-aversity of both Buyer and Seller so that ‘choose to provide the service’ and ’choose to use it’ provided an equilibrium strategy for the game.

In our talk, we plan to explain what we have written above. We would like to do this by showing bits and pieces of the codebase we developed in [2], explaining to the audience how we modelled things and how we were able to run analytics on the model. In simple words, we aim to present our software and techniques by means of a full-flagged practical example.

References

[1] 20squares UG. *20squares Website*. 2023. URL: https://20squares.xyz/.

[2] 20squares UG. *Ledger-Hedger Model in Open Games*. 2023. URL: https://github.com/20squares/manifold-finance/tree/main/manifold-hedger.

[3] J. Bolt, J. Hedges, and P. Zahn. *Bayesian Open Games*. 2019. arXiv: 1910.03656 [cs, math].

[4] V. Buterin. “A Next-Generation Smart Contract and Decentralized Application Platform”. In: *Ethereum* (January 2014), pp. 1–36.

[5] CyberCat Institute. *Open Game Engine Repository*. 2023. URL: https://github.com/CyberCat-Institute/open-game-engine/.

[6] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels. *Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges*. 2019. arXiv: 1904.05234 [cs].

[7] N. Ghani, J. Hedges, V. Winschel, and P. Zahn. *Compositional Game Theory*. 2016. arXiv: 1603.04641 [cs].

[8] Manifold Finance. *Manifold Finance Website*. 2023. URL: https://www.manifoldfinance.com/.

[9] I. Tsabary, A. Manuskin, and I. Eyal. *LedgerHedger: Gas Reservation for Smart-Contract Security*. Cryptology ePrint Archive, Paper 2022/056. https://eprint.iacr.org/2022/056. 2022.