

AMM experiment

Gal Harel³, Gabriele Camera¹, Lin William Cong², Nir Chemaya³, and Ro'i Zultan³

¹ESI, Chapman University

²Cornell University, ABFER, IC3, & NBER

³Ben-Gurion University of the Negev

ARTICLE INFO

Article history:

Received 15 May 2024

Accepted 15 July 2024

Available online 19 August 2024

JEL classifications:

C92

Keywords:

AMM

Economic experiments

DeFi

Liquidity Pools

Fintech

Behavioral economics

ABSTRACT

This study explores the dynamics of Automated Market Makers (AMM) in experimental settings. [cite: 24] We investigate participant behavior in decentralized liquidity pools to ensure randomized results and transparency. [cite: 25]

© 2026 Elsevier B.V. All rights reserved. [cite: 26]

1. Introduction

The growth of experimental methods in economics and related social sciences has been driven not only by methodological progress, but also by advances in digital infrastructure for recruitment, administration, and execution of experiments. Building on this development, AMM Game contributes a modern, integrated platform for blockchain-based market experiments by combining participant management, experiment setup, live session control, and data capture in a single system. The project follows the same broader trajectory as earlier landmark tools in experimental economics, while addressing current requirements such as real-time coordination, transparent workflow tracking, and reproducible assignment procedures. As experimental environments become increasingly connected and information spreads rapidly across online channels, robust control of recruitment and session flow becomes essential; AMM Game is designed with this challenge in mind. In parallel, the platform emphasizes auditability through explicit records of player assignment, experiment lifecycle transitions, and participation activity, supporting the documentation standards expected in high-quality empirical research. Ongoing open-source development and deployment for direct use further position AMM Game as a practical and evolving infrastructure for contemporary experimental research.

*Corresponding author at: Ben-Gurion University of the Negev.

†We thank the participants of the AMM experiment sessions for their feedback.

2. Technical features

2.1. Development environment

The AMM Game is implemented as a full-stack web platform with a Python FastAPI backend and React-based frontend clients (Vite + TypeScript) for player and manager interfaces. Core backend services use SQLAlchemy and Alembic for data modeling and schema migrations, while PostgreSQL is used for persistent data storage and Docker Compose is used to standardize local infrastructure and deployment setup.

2.2. How to Get AMM Game

We use the open-source platform GitHub to develop and distribute AMM Game, enabling transparent collaboration and continuous improvements by the community. The source code is publicly available at [GitHub](#), where users can access the latest version, track updates, and contribute enhancements through pull requests and issue reporting. In addition to the repository, the application is already deployed and open for use, so users can [directly access the live system](#) without a local installation. GitHub registration is free of charge and is recommended for users who want to report bugs, request features, or participate in development.

3. User features

3.1. User hierarchy

The AMM Game uses role-based access with two core account types and three practical personas in laboratory operation: administrators, experiment managers, and participants (players). Administrators and managers use the management interface to configure and run experiments, while participants use the player interface to join sessions and interact only with their own game and profile data. Access is controlled through secure authentication, and guest/invite-based participant entry is supported when needed. This hierarchy separates operational control from participant activity and supports efficient execution of controlled experimental sessions.

3.2. Administrator

Administrators have full access to all AMM Game capabilities, including functions that are also available in limited form to other user groups. They can create and edit experiments, recruit and assign players, control experiment lifecycle states (e.g., draft, started, running, ended), configure rounds and market parameters, and monitor live activity and transactions during sessions. Administrators can also manage user accounts and availability, maintain platform-level settings, and export experimental data for analysis. In the following sections, only a selected subset of administrator features is presented in detail.

3.2.1 Setting up an experiment

To register a new experiment in AMM Game, administrators define the core configuration and operational metadata required for execution and analysis. This typically includes the experiment name, a short description, number of players, number of groups, round structure, and AMM-related parameters such as balances and pool settings. The platform also supports reusable templates, allowing administrators to standardize experiment designs and quickly instantiate new runs with consistent settings. During preparation, administrators can edit experiment status and lifecycle stages (for example draft, started, running, ended), which provides controlled progression from setup to live session.

3.2.2 Assigning pool members

Administrators assign participants to experiments through the management interface, including recruitment of active users and allocation to groups. AMM Game stores player-to-experiment and group assignments explicitly, making selection and allocation reproducible and auditable after sessions. Deactivated participant accounts are retained for historical documentation of prior experiment activity.

3.2.3 Initiating invitations

Participant recruitment for an experiment is initiated actively by an administrator through the management interface. Administrators can control when recruitment starts, monitor how many slots are filled, and stop recruitment at any point. The system supports controlled onboarding via active-user selection and invitation-based entry, reducing uncontrolled first-come-first-served effects and enabling more balanced session composition. Once required capacity is reached, recruitment can be closed and the experiment can proceed to the next lifecycle stage.

3.3. Experimenter

Experimenters and administrators work through a dedicated management dashboard that provides direct access to experiment creation, recruitment, lifecycle control, and live monitoring. Experiments can be managed across stages (for example draft, started, running, ended), and participant status can be tracked in real time during active sessions. Assignment and grouping actions are recorded in the system, supporting transparent sampling and post-session review. The platform also supports structured handling of experiment artifacts and outputs, facilitating collaboration and reproducibility.

3.4. Participant

Participants access the platform through personal authentication or controlled guest/invite entry, depending on study configuration. After login, they can view their active experiment context, join assigned sessions, and interact with the AMM interface according to their permissions. Participant-facing pages also provide access to profile and experiment-related information, while core experiment-control functions remain restricted to management roles. This separation preserves usability for participants while protecting study integrity and administrative control.

4. Summary

We hope that the AMM Game will become a meaningful contribution to experimental research on market behavior, decision-making, and learning in blockchain-based environments. The platform is designed to improve quality, transparency, and reproducibility by integrating participant management, experiment configuration, recruitment, session execution, and data collection in one system. By documenting key operational steps such as player assignment, experiment progression, and participation status, it supports consistent experimental governance and post-session analysis. Ongoing development is open-source and community-driven through the project's GitHub repository, while the deployed application remains openly accessible for practical use and continuous feedback.