Measuring Inflation within Virtual Economies using Deep Reinforcement Learning

Conor Stephens[1,2] and Chris Exton[1,2]
[1] *Computer Systems and Information Science, University of Limerick, Ireland*
[2] *Lero, Science Foundation Ireland Centre for Software Research, Ireland*

Keywords: Reinforcement, Learning, Games Design, Economies, Multiplayer, Games.

Abstract: This paper proposes a framework for assessing economies within online multiplayer games without the need for extensive player testing and data collection. Players have identified numerous exploits in modern online games to further their collection of resources and items. A recent exploit within a game-economy would be in Animal Crossing New Horizons a multiplayer game released in 2020 which featured bugs that allowed users to generate infinite money (Sudario, 2020); this has impacted the player experience in multiple negative ways such as causing hyperinflation within the economy and scarcity of resources within the particular confines of any game. The framework proposed by this paper can aid game developers and designers when testing their game systems for potential exploits that could lead to issues within the larger game economies. Assessing game systems is possible by leveraging reinforcement learning agents to model player behaviour; this is shown and evaluated in a sample multiplayer game. This research is designed for game designers and developers to show how multi-agent reinforcement learning can help balance game economies. The project source code is open source and available at: https://github.com/Taikatou/economy_research.

1 INTRODUCTION

Video game economies suffer from a massive internal problem when compared to real-world economies. The creation of the currency in-game is tied to player mechanics; an example would be when players receive money when they defeat a monster. Systems or mechanics that result in the creation of economic value is effectively printing money. This core issue has been the cause of rampant inflation within Massively multi-player online games (MMO) (Achterbosch et al., 2008). An early example was in *Asherons Call:* the in-game currency became so inflated that shards were used instead of money. Similarly, the developers of *Gaia Online* would donate $250 to charity if the players discarded 15 trillion Gold (Gold was the in-game currency used for transactions) (Haufe, 2014). Game designers would create a counterweight to prevent problems within the game's internal balance. *Sinkholes* are key tools for game's designers when designing economies within the game's systems that remove money from the economy. Sinkholes are mechanics that take currency out of the games system, sometimes referred to as "Drains" within machination diagrams (Adams and Dormans, 2012). Sinkholes have taken the form of Auction Fees and Consumable Items only available from NPC's (Non-Playable Characters). However, players will endeavour to avoid these taxes and attempt to amass vast amounts of currency and items as quickly as possible.

Game systems with MMO titles have been shown to implode due to un-intended player behaviour. *Ultima Online*, an incredibly successful early MMO released in 1997 was an early example of the issues players could present to game designers. The creator Richard Garriott explained in an interview how players destroyed the virtual ecology within seconds by over-hunting (Hutchinson, 2018). Within the game where the spawning of new monsters was regulated to a specific frequency; when the game went live problems started to surface, resources were being depleted at an alarming rate, destroying the game's ecosystem due to players over hunting herbivores for fun. To solve this problem future games did not restrict the rate that enemies would spawn, MMO games such as World of Warcraft and *Star Wars: The Old Republic* uses sinkholes to manage the imbalance set within their economies.

The framework that we propose assesses inflation within an in-game economy by measuring the price of resources within the game over time. In this framework the economy is simulated by using reinforcement learning agents playing as both the supplier and consumer within a simplified game economy both

Stephens, C. and Exton, C.
Measuring Inflation within Virtual Economies using Deep Reinforcement Learning.
DOI: 10.5220/0010392804440453
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 444-453
ISBN: 978-989-758-484-8

aiming to gain more wealth through the game systems. This research aims to show the power of reinforcement learning for both *Game Balance* (Beyer et al., 2016) ensuring the game is challenging and fair for different types of players and also assessing the game's economy for potential inflation allowing key stakeholders to better understand how changes within the virtual economy can influence player interactions such as trade without releasing the game or extensive user testing. This paper is built upon two clear objectives:

1. Measure the upper limits of inflation within a multiplayer game economy;
2. Provide a tool for simulating different changes to a game's economy for analysis.

2 RELATED WORKS

This work builds upon many different fields of game's design such as automated game design, game balance and game theory as well as concepts from artificial intelligence and simulation such as deep reinforcement learning and Monte Carlo simulations. The history of these fields are briefly described below with relevance to this paper.

2.1 Automated Game Design

Game Generation systems perform automated, intelligent design of games (Nelson and Mateas, 2007), whilst having a lot of similarities with procedural content generation; Game Generation is a sub discipline of Automated Games Design with the latter being a larger collection of tools, techniques and ambitions, such as collaborative design tools and testing frameworks for games designers. Automated Game Design (AGD) strives not to create content and assets for preexisting games systems which comprises of goals, resources and mechanics (Sellers, 2017); AGD strives to create these systems allowing unique experiences and interactions to be generated.

2.1.1 History of Game Design Systems

AGD was created to develop rulesets for games given a common structure such as chess and strategy board games. The first example of such a system was Metagame (Pell, 1992). This was partly due to breakthroughs in Artificial Intelligence (AI). It was also partly due to the extensive history of these games and the recognition they received in academia, highlighting the importance of such a genre and further exploration of the different design possibilities contained within the well established framework and brought public acknowledgement to the suitability and eligibility of the pursuit of further creation and exploration within this space.

After the mid 2000's the study, development and research into modern video games ballooned in correlation to their growing popularity and financial success of the wider games industry. In the beginning of the 2000's proceduralism was becoming successful in games such as *The Marriage* and *Braid* (Treanor et al., 2011) Games with proceduralism as a core mechanism of the narrative are process-intensive. In these games, expression is found primarily in the player's experience as it results from interaction with the game mechanics and dynamics (Hunicke et al., 2004).

This marked the period of an explosion of AGD systems emerging from UC Santa Cruz. The first would be *Untitled System* which is capable of creating mini-games from natural text input. This system was the first AGD system to change the visuals of the game based on the input by having access to an image database. Game-o-matic was the second AGD system that created a buzz in short succession (Treanor et al., 2012). *Game-O-Matic* is a videogame authoring tool and generator that creates games that represent ideas. Through using a simple concept map input system, Game-O-Matic is able to assemble simple arcade style game mechanics into videogames that represent the ideas represented in the concept map. Game-O-Matic gave a higher quality experience providing a more interactive and engaging process when seeding ideas (seeding refers to a number or vector used to initialize a random number generator, within Game-O-Matic. The seed refers to the terms used to describe the desired game).

2.2 Automated Game Testing

Developers and researchers have been experimenting with machine learning to test their games. This testing has a lot of different use cases and contexts depending on the department spearheading the initiative. Quality assurance departments may consider employing trained bots/learning agents to test the game levels and stories to detect bugs. Examples include: missing collision within the level geometry; finding holes in the narrative events that can hard/soft lock players from progressing; *modl.ai* is an excellent example of a company which offers this service to games developers and publishers. Dev-Ops and the monitization groups of the game company could employ A/B testing to assess the quality of tutorials, in-game-currency and menu systems on different user groups. Game designers may also use artificial intelligence and simulations as a way of assessing the quality metrics such as fairness (of level design) or session-length of the game they are developing and by extension testing, such efforts are made easier by using modern tool such as *Unity Simulations* a cloud based service provided by Unity 3D game engine. Unity Simulation is an excellent resource for testing game balance automatically with different parameters.

2.2.1 Play Testing

Play testing is an integral component of user testing games to ensure a fun and engaging experience for players, designed to review design decisions within interactive media such as games and to measure the aesthetic experience of the game compared to the designer's intentions (Fullerton, 2008). Play testing is a lengthy process both in terms of complexity, cost and time. Several approaches have been presented over the past few years to automate some steps of the process (Stahlke and Mirza-Babaei, 2018). These solutions aim to allow faster, more accurate iterations of designs and greater efficiency during the professional game development cycle. An example of this in practice is *King's* automated play testing of their match-3 "Crush Saga" games where they trained agents to mimic human behaviour and using these agents within simulations, designers can test new level designs (Gudmundsson et al., 2018).

2.3 Game Balance

Game Designers rely on a variety of practices to ensure fair and fun experiences for players by analysing mechanics and systems. In order for designers to create these experiences, the designer assesses the fairness, challenge, meaningful choices and randomness of the different events and challenges within the game. Several terms have emerged over the years to define and explain aesthetics within common genres, one such term is 'Game Feel' (Swink, 2009) a practice of making the mechanics i.e. the "moving parts" of a game more impactful by adding effects such as:

- Screenshake;
- Particle Effects;
- Post processing.

An oversimplification of game balance can be defined as "do the players feel the game is fair" (Becker and Görlich, 2019). To successfully balance a game a variety of considerations and tolerances depending on the game, genre and audience must be considered.

2.3.1 Cost Curve Analysis

Game Balance was traditionally achieved using analytical methods such as cost-curve analysis, spreadsheet and Excel macros. These tools have evolved into machination diagrams, a more visual and interactive framework to test internal game balance (Adams and Dormans, 2012). Each of these methods follows a reductive process that allows the designers to see the impact when values are changed within the games' internal systems. Cost-curve analysis as an example associates every mechanics, systems and items within the game with a benefit and a cost (Carpenter, 2003). Traditionally, the cost increases in line with the benefit of a game mechanic or resource. This relationship between each item forms the shape of the curve. During the balancing process a designer can see if any individual mechanic character or item deviates too much from the curve. Small imbalances are allowed, as choosing the most efficient or effective item creates an interesting decision for the player. However game mechanics that are obviously too high on the cost curve are traditionally identified for a redesign or a small nerf. This approach works well with traditional arcade games and role playing games. However, this approach is less effective in multiplayer games, that have high dependence on intransitive relationships, which can be described as Rock-Paper Scissors mechanics.

2.3.2 Balancing Collectable Card Game Decks

New research into balancing decks of cards within the collectable card game Hearthstone (de Mesentier Silva et al., 2019) showed how genetic algorithms combined with simulated play can show how many viable options there are for players and how to identify critical compositions and decks that would be used if a card within the deck changes. This research shows how fair games are created with genetic algorithms by changing the available mechanics given to both players within a complicated problem domain (over 2000 cards to evaluate). It also opens up several possibilities of testing the probability of victory for any two players based on their deck contents and identifying key cards that decks rely on. The key research identified from this paper is how to evaluate the health of the games META game, (Which describes how the game is played strategically by players at any certain time) by using entropy within the viable decks used by Artificial Intelligence.

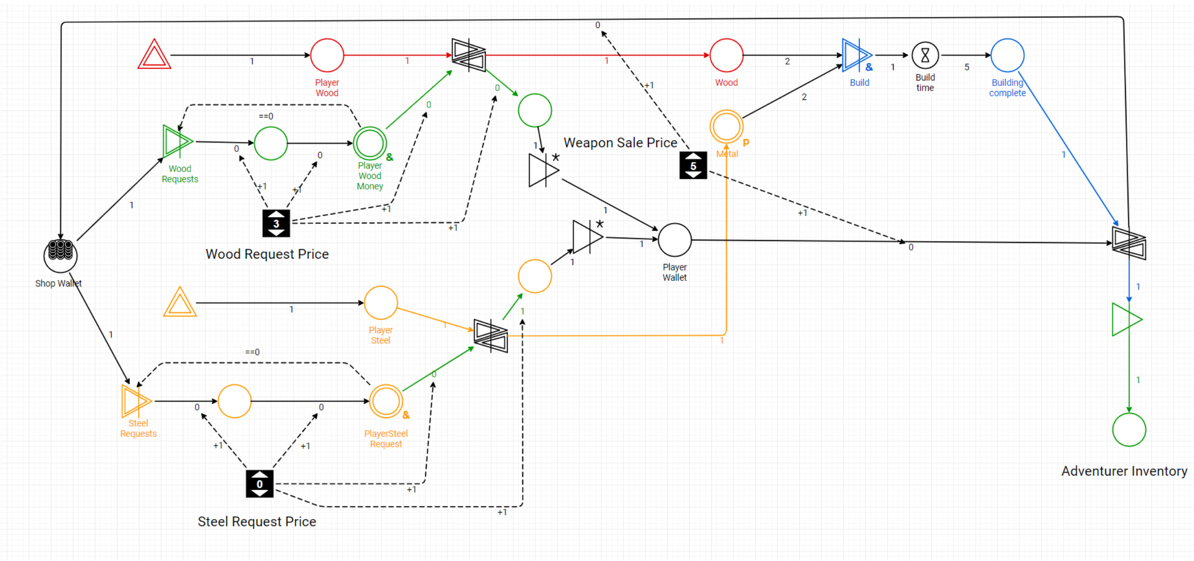

Figure 1: Crafting Machinations Diagram.

3 HISTORY OF ECONOMIES IN GAMES

MMOs feature some of the most impressive economic interactions between agents found in video games. Lots of research has been done to explore virtual economies (Castronova, 2002). This research identified some necessary mechanics that are required to allow fluctuations in value for virtual currencies. These mechanics which include the creation of the currency using in-game mechanics can be formulated as: 'Be directly connected to real-world currency'. An early example of this tie between physical and game currency can be found in *Entropia Universe*, a free-to-play MMO released in 2003 where 1 USD is tied to 100 PED (Project Entropia Dollars) on a fixed exchange rate (Kieger, 2010). There are ethical concerns for this mechanic regarding how the currency is stored and how it can be used for gambling as a key game mechanic.

Game economies include an extensive range of possibilities ranging from Mana in *Magic The Gathering* to Gil in *Final Fantasy.* When designers talk about balancing game economies, they generally mean balancing mechanics that feature conversion between two resources. In game economies, game designers control and carefully balance the economy, defining how players obtain the currency and by how much of each currency key resources cost in the game. This paper discusses game economies that are significantly affected by supply and demand between multiple players. To facilitate this specific in-game mechanics, a first requirement would be allowing players to sell items within the world, the second would be having different requirements for different types of players incentivising trading and cooperation.

MMO's markets and currencies feature inflation because the players influence both the supply and demand of items. This is very different from traditional single-player games where prices of items in shops are a key balance consideration. Having agents in the world can have severe design implications regarding player progression and difficulty achieving particular quests and challenges. Within single player games the prices of items can be set to allow sufficient challenge without giving the player an un-surmountable hurdle to cross, multiplayer games can feature supply and demand which can affect the players ability to progress within the game world. Other techniques found in 'free to play' games are illiquidity (Players have to convert one currency into another before they can make certain purchases).

Due to the infinite nature of enemies and game play found within MMO games; item scarcity is hard to predict within any specific item unless it is made available to only the smallest groups of players. Many games create prestige and scarcity by allowing players to own physical space (for example land and property) in *Second Life* or spaceships and space-stations (as it is done in *Eve Online*). This economy is different from the game play based economy where property is innately a finite resource that can be traded endlessly. It also has different types of values based on location and amenities.

Larger Massive Multiplayer Online (MMO)

games such as *RuneScape* and *World of Warcraft* (WoW) have tackled the issue of hyperinflation throughout the games lifetime. MMO's have experimented with integrating real-world financial policies to curb inflation. These solutions include creating a reserve currency to put a minimum value in the in-game currency. It is also worth implementing illiquidity in the transnational mechanics within the games auctions and trading systems. These design considerations are slow to develop and it is difficult to measure the effects accurately before the game's release.

4 TESTING ECONOMIES

Artificial Intelligence techniques have provided some means for testing and evaluating economic policy and fiscal design. Salesforce showed how using a learning environment; different tax policies can be tested and assessed on specific economies (Zheng et al., 2020). The AI economist is capable of using reward signals from the learning environment such as the positive affirmation of earning money and the punishment of being taxed to create incentives for the agents to relax and enjoy their day and to work within their society and contribute. This work highlights the great potential in finding optimal tax rates and policies to generate the most income for the government without creating disincentives for the agents working when they could be resting.

Games developers have traditionally used modelling techniques to test economies within games (Adams and Dormans, 2012). Machinations is a graph-based programming language created by *Joris Dormans* with a syntax that supports the flow of resources between players. A modern equivalent is the fantastic Machinations.io web app that modernises the previous machinations toolset. Machinations allow the game designers to model the creation flow and destruction of resources within the economy. It also allows resources to be exchanged and converted within a visual interactable simulation.

5 METHODS

This research aims to measure inflation within a game economy. This is achieved by using a game economy as a learning environment for reinforcement learning agents and tracking their trades within an accelerated simulation of the game world. Industry standard tools are used for the game engine and sample code to make this work more accessible for game development professionals. The process to achieve this can be broken into 3 steps:

1. Implement a simple MMO economy using Unity game engine;
2. Train learning agents to play the game allowing them to create, sell and win resources from the game world;
3. Assess the game economy by deploying the trained learning agents within the game world and record the price of transactions.

This framework assesses sample virtual economies within a simulated MMO game using learning agents to simulate the needs and desires of different players. The sample economy is a two way economy between players that focuses on **crafting** and **selling** items and players concentrate on **adventuring** and consuming items. Both sets of agents within this learning environment are dependent on cooperation and bartering between each other to progress in the game. Both crafting and adventuring agents are rewarded with the money they are making. The implementation for the agent's policy was possible using ml-agents an open-source plugin for unity (Juliani et al., 2018) that allows for the development of reinforcement learning agents within Unity.

5.1 Economy Design

Traditional MMO economies are facilitated with supply and demand provided by the game's players. The demand within the in-game economy is achieved by having players needing different types of weapons and resources to achieve their short term objectives within the game. Examples include how Warriors and Mages in *World of Warcraft* would desire different equipment, allowing both to sell unwanted or useless equipment to each other. This example project features a similar dynamic between adventurer and craftsman agents. Craftsman agents spend money to receive resources to make equipment which they sell for a profit. Adventurer agents in an opposite approach use the equipment to collect resources that they sell to craftsman agents. Both agents have to cooperate to prosper within this economy, allowing a wide variety of interactions and influences. The supply is achieved by having mechanics within the game that creates and changes different resources; these include loot drops from battles, and crafting resources into more valuable weapons, that will sell at a higher price to buyers.

The game's internal economy was tested as a machination diagram, as shown in Figure 1. The economy is based on Craftsman and Adventurer agents agreeing on the price of resources and items to facilitate trade. This agreement within the system can be seen in the two types of *Registers* within the machination diagram. The first register type is for the Craftsman agent that allows them to set the price of the request. Moreover, it is an interactive pool that Adventurer agents can use to agree on the price. This interaction locks negotiations for both parties. The second type of register within this diagram is the price for the crafted weapon. This register allows the craftsman to change the price of the weapon even when they have no stock of the weapon. When an Adventurer agent interacts with the Trade node, this exchanges the item in the shop with the Adventurer agent's money. The values chosen for the different outcomes were based on play testing the economy design with different parameters to ensure neither the Adventurer nor the Craftsman runs out of money, which could bring the simulation and economy to a standstill. This was possible using Monte Carlo techniques to simulate players using the Machinations.io (Raychaudhuri, 2008). The economy as shown in Figure 1 was simulated over 1000 time-steps within the environment over 20 iterations for each configuration of the items specific statistics. An example of the balancing process can be seen in Figure 5 which shows the steady upwards direction of the economy and the constant progression and purchasing throughout the simulation. A Monte Carlo search for games design is a popular strategy for exploring the search space of strategies and game states (Zook et al., 2019) (Zook and Riedl, 2019).

5.2 Adventure Agents

Each Adventurer agent can access three unique game systems where they can interact with other agents. The first game system is the request system, similar to a guild in *World Of Warcraft*; Adventurers can see all active resources that Crafting agents requested and the reward when the request is complete. Adventurer agents can choose up to five individual requests to fulfil. The second ability is the Adventurer ability; agents are able to travel within the virtual world. There are four different environments ***Forest, Mountain, Sea and Volcano*** each with varying types of enemies to encounter. When the agent enters an area, a random battle occurs which is designed similarly to Pokemon battles; the authors used Brackeys turn-based combat framework for the implementation of this system. Players can choose to ***Attack, Heal or Flee*** during combat. If either the players' or enemies' health points drop to zero the game is over. If the player was victorious, a loot drop takes place with

Figure 2: Battle Interface.

items and probabilities based on the enemy that the agent defeated. Specific loot values for different enemy types can be found in Table 1. The user interface for this battle mode can be found in Figure 2. The main challenge of the battle system is to manage the player's health points between battles and battling enemies that have sufficient chance to drop resources needed for each agent's accepted requests.

The third ability of the Adventurer agent is the shopping ability agents have access to. All the Crafting agents have shops and Adventurer Agents can purchase items from them. Each weapon is placed within the Adventurer's inventory and a weapon is equipped if it is the strongest weapon in the agents inventory.. Better equipment allows the Adventurers to encounter stronger enemies and receive more loot. If the agent's health drops to zero, it loses the battle and suffers a loss of 5 units of gold from the Adventurer agents wallet.

5.3 Crafting Agents

Each Crafting agent has three different abilities. The first is the ability to input requests for materials which include ***Wood, Metal, Gems and Dragon Scales***. Within the request system, each agent can change the number of each request's resource they require and the price they are willing to pay for it. This system is similar to a guild request found in many traditional MMO's. From this marketplace, Adventurer agents will take requests which they can fulfil through battling and exploration. The second ability each Crafting agent has is crafting; each agent can craft a variety of weapons using the resources they have received. The third and final ability of the Crafting agent is how the agents interact with their shopfront. Agents can put items in their shopfronts to replenish stock. They also have the ability to increase and decrease the price of each item. There are five unique weapons that a Craftsman could create with different crafting

Table 1: Enemy Loot Drops.

Area	Name	Health	Damage	Wood	Metal	Gem	Scale
Forest	Owl	10	2	1			
Forest	Buffalo	10	5	2	1		
Mountain	Bear	15	5	2	2		
Mountain	Gorilla	20	5	2	2	1	
Sea	Narwhale	15	6	1	1	2	
Sea	Crocodile	22	4		2	2	
Volcano	Snake	20	5			2	1
Volcano	Dragon	25	6		2	1	2

requirements; this can be seen in Table 2.

5.4 Agent Sensors

Each agent is trained using Soft Actor-Critic policies using multi-agent reinforcement learning; this is possible by using Unity's excellent ML-Agents package (Juliani et al., 2018). ML-Agents allows developers and designers to train learning agents within Unity game environments. The knowledge is achieved by using *Self Play* within the environment (Silver et al., 2017). Self Play improves the learning process by having multiple agents within the environment use previous versions of the agent's policy. Self Play stabilises the agents learning by allowing a positively trending reward within the learning environment. This was implemented within the learning environment by having half of the Adventurer and Craftsman agents infer their decisions from previous versions of the policy. Both Crafting and Adventurer learning agents switch between active abilities and receive signals from the statistics and environment. These include the price of items within the shop, the health and damage data during battles and the content within the agents' inventories. Agents have discrete actions with different contexts depending on the agent's currently enabled ability.

5.5 Economy Sinks

Both the adventurer, crafting and virtual environment for the game were designed to sustain a healthy supply-demand curve to regulate the prices between the different weapons and resources. However, the base environment has a traditional feature of online game economies that have been the cause of previous instances of hyperinflation. Such hyperinflation is caused by the infinite monsters within the adventuring system, potentially allowing the cost of resources to plummet for the Crafting agents. The game is designed to prevent hyperinflation by having excess rewards removed from the in-game economy. This is achieved by having each Adventurer agent discard items if they do not have a current request for that item. This mechanic is similar mechanic to how *Monster Hunter's* request system works by having unique issues for the early application preventing the player collecting all potentially early game items within their first journey into the wild area.

5.6 Training

The reward signal for both Adventurer and Craftsman agents was to incentivise the agents to coordinate with each other. This was achieved by rewarding the Craftsman agent for selling items with the reward signal being the percentage profit made during a sale. The learning agent is rewarded when it earns money. This is achieved when an agent completes a resource request. The Adventurer agent is penalised with a penalty of -0.1 to punish the agent for its failure. Adventurer agents have a similar reward signal; every time they earn money, they receive a proportional reward signal ranging between 0-0.8 based on the value of money they have gained. Adventurers earn money when they complete a resource request. Both agent's policies are updated using *Soft-Actor Critic* a learning algorithm for reinforcement learning (Haarnoja et al., 2018). Soft Actor-Critic allows higher levels of entropy within the system, enabling further exploration of the stochastic action distribution.

Each episode during training is started by resetting every agent's inventory and wallet, giving them the starting value of 100 gold units and no weapons. When the **Ultimate Sword** is both crafted by the Craftsman agent and sold to an Adventurer agent, the episode is determined to be over, as there is no future content that agents have yet to experience. Each agent's neural network has 128 units in each of its two hidden layers between input and output neurons. We previously explored having intrinsic rewards during the training process however we observed no significant increase in performance within the learning environment; with a significant increase to the computation cost whilst training each agent's policy. This reward signal was achieved by supplying a small cu-

Table 2: Weapon Stats & Crafting Requirements.

Item Name	Damage	Durability	Wood Req.	Metal Req.	Gem Req.	Scale Req.
Beginner	7	5	2	2		
Intermediate	9	6	3	3		
Advanced	10	7	2	2	1	
Epic	12	7	2	2	2	
Master	13	8	3	3	3	
Ultimate	15	10	3	3	3	2

riosity reward to encourage exploration of the possible scenarios within the learning environment.

5.7 Data Collection

After successful training of both the Craftsman and Adventurer agents' policies, the trained models were then used within the learning environment to generate and collect data. Data was collected by recording sale prices of weapons within the simulation. This was recorded as Comma Separated Value data alongside the time within the simulation that the exchange took place.

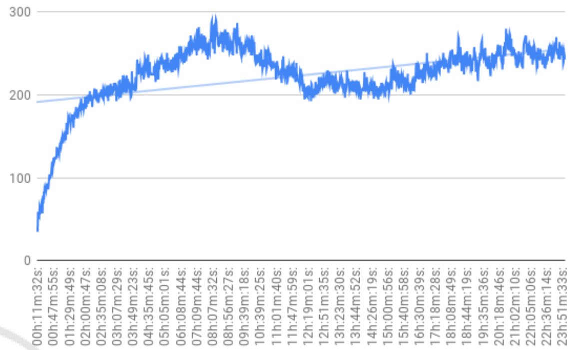

Figure 3: Sale value of Beginner Swords.

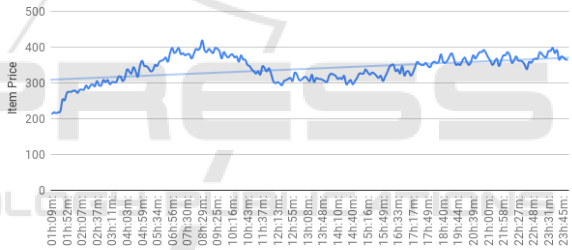

Figure 4: Sale value of Advanced Swords.

6 RESULTS

The sale price of Beginner and Advanced swords are shown in Figure 3 and Figure 4 respectively. The trend lines in both diagrams show a clear upwards trajectory within the virtual economy but their shapes provide a more interesting picture of this game's economy. Between the 5th and 10th hour of simulation the price increases of both weapons. This could be seen as a higher level of demand for the item from the Adventurer agents. This is quickly followed by a decrease in price representing a small dip within the game's economy, potentially relieving the burden on the Adventurer agents within the economy and having the burden of more competitive prices affecting the Shop agents reward signal.

7 DISCUSSION & FUTURE WORK

Economies within games are set to grow in both importance and scale (Toyama et al., 2019) (Tomić, 2017), which makes robust testing of the systems before releasing them to the public a vital precaution to ensure consistent profitability of the game throughout its life-cycle. Testing virtual economies has the potential of allowing companies to protect their assets whilst earning higher and more consistent revenue during the products life cycle. Other techniques to test game economies such as machinations and analysis of player data allows developers to assess the health of the game's economy. However, these approaches are possible only during prototyping and after the release of the game. Our approach offers a middle ground that allows key stakeholders of the game to test the same economic systems found within the game. Moreover, it allows player motivation to be modelled in the form of the reward signal found within our learning environment.

Future work within this research should explore the effectiveness of this approach in different types of economies with other starting parameters. Comparing different economies would make it possible to evaluate the effect of inflation within economies containing positive or negative feedback loops. *Monopoly* with a very high level of positive feedback within the game has a history of exploding inflation and competition as a core premise. Comparing this economy to a more serious game with a negative feedback loop such as *This War of Mine* could highlight the exact difference in flow between these two very different experiences.

The authors propose using this framework in a more extensive online game with multiple mechanics for gaining money and resources. This could potentially allow game designers to assess the impact of different play styles and systems on the games internal economy. This research could be extended by training another neural network whose responsibility is to view transactions between agents in order to predict inflation within the economy during the simulation. This neural network could be used within the game to prevent fraud or hyperinflation in a way similar to how credit card companies prevent fraud in their systems.

ACKNOWLEDGEMENTS

We would like to thank Lero, the Science Foundation Ireland centre for Software Research for their continued help and support.

REFERENCES

Achterbosch, L., Pierce, R., and Simmons, G. (2008). Massively multiplayer online role-playing games: The past, present, and future. *Comput. Entertain.*, 5(4).

Adams, E. and Dormans, J. (2012). *Game Mechanics: Advanced Game Design*. New Riders Publishing, USA, 1st edition.

Becker, A. and Görlich, D. (2019). Game balancing – a semantical analysis.

Beyer, M., Agureikin, A., Anokhin, A., Laenger, C., Nolte, F., Winterberg, J., Renka, M., Rieger, M., Pflanzl, N., Preuss, M., and Volz, V. (2016). An integrated process for game balancing.

Carpenter, A. (2003). Applying risk analysis to playbalance rpgs.

Castronova, E. (2002). On Virtual Economies. *Game Studies*, 3.

de Mesentier Silva, F., Canaan, R., Lee, S., Fontaine, M. C., Togelius, J., and Hoover, A. K. (2019). Evolving the hearthstone meta. *CoRR*, abs/1907.01623.

Fullerton, T. (2008). Game Design Workshop. A Playcentric Approach to Creating Innovative Games. pages 248–276.

Gudmundsson, S., Eisen, P., Poromaa, E., Nodet, A., Purmonen, S., Kozakowski, B., Meurling, R., and Cao, L. (2018). Human-like playtesting with deep learning. pages 1–8.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870, Stockholmsmässan, Stockholm Sweden. PMLR.

Haufe, V. (2014). Hyperinflation in gaia online — know your meme. https://knowyourmeme.com/memes/events/hyperinflation-in-gaia-online. (Accessed on 08/05/2020).

Hunicke, R., Leblanc, M., and Zubek, R. (2004). Mda: A formal approach to game design and game research. *AAAI Workshop - Technical Report*, 1.

Hutchinson, L. (2018). War Stories: Lord British created an ecology for Ultima Online but no one saw it.

Juliani, A., Berges, V., Vckay, E., Gao, Y., Henry, H., Mattar, M., and Lange, D. (2018). Unity: A general platform for intelligent agents. *CoRR*, abs/1809.02627.

Kieger, S. (2010). An exploration of entrepreneurship in massively multiplayer online role-playing games: Second life and entropia universe. *Journal For Virtual Worlds Research*, 2.

Nelson, M. and Mateas, M. (2007). Towards automated game design. pages 626–637.

Pell, B. (1992). *Metagame in Symmetric Chess-like Games*. Technical report (University of Cambridge. Computer Laboratory). University of Cambridge Computer Laboratory.

Raychaudhuri, S. (2008). Introduction to monte carlo simulation. In *2008 Winter Simulation Conference*, pages 91–100.

Sellers, M. (2017). *Advanced Game Design: A Systems Approach*. Game Design. Pearson Education.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T. P., Simonyan, K., and Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815.

Stahlke, S. N. and Mirza-Babaei, P. (2018). Usertesting Without the User: Opportunities and Challenges of an AI-Driven Approach in Games User Research. *Comput. Entertain.*, 16(2).

Sudario, E. (2020). The broken economy of animal crossing: New horizons — cutiejea. https://www.cutiejea.com/real-life/acnh-economy/. (Accessed on 12/30/2020).

Swink, S. (2009). *Game Feel: A Game Designer's Guide to Virtual Sensation*. Morgan Kaufmann Game Design Books. Taylor & Francis.

Tomić, N. (2017). Effects of micro transactions on video games industry. *Megatrend revija*, 14:239–257.

Toyama, M., Côrtes, M., and Ferratti, G. (2019). Analysis of the microtransaction in the game market: A field approach to the digital games industry.

Treanor, M., Blackford, B., Mateas, M., and Bogost, I. (2012). Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the The Third Workshop on Procedural Content Generation in Games*, PCG'12, page 1–8, New York, NY, USA. Association for Computing Machinery.

Treanor, M., Schweizer, B., Bogost, I., and Mateas, M. (2011). Proceduralist readings: How to find meaning in games with graphical logics.

Zheng, S., Trott, A., Srinivasa, S., Naik, N., Gruesbeck, M., Parkes, D. C., and Socher, R. (2020). The ai economist: Improving equality and productivity with ai-driven tax policies.

Zook, A., Harrison, B., and Riedl, M. O. (2019). Monte-carlo tree search for simulation-based strategy analysis.

Zook, A. and Riedl, M. O. (2019). Learning how design choices impact gameplay behavior. *IEEE Transactions on Games*, 11(1):25–35.

APPENDIX

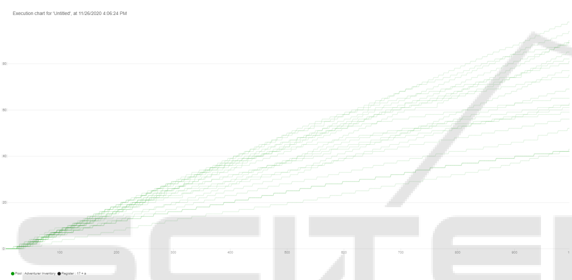

Figure 5: Monte Carlo testing of Economy Design.

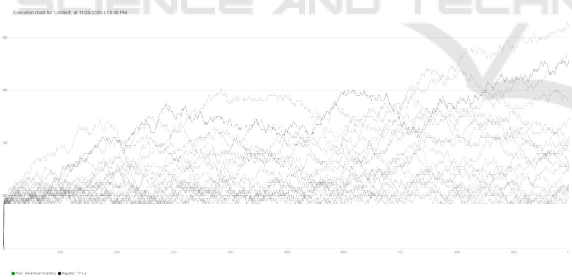

Figure 6: Monte Carlo Item Purchase Frequency.