ADAPTIVE AI & RRA SYSTEM FOR ROLE PLAYING GAME

Rutvij Joshi[1], Kashyap Patkar[2], Shraddha Sonde[3], Pravin Patil[4]

[1,2,3,4] *Dept. of Information Technology, Padmabhushan Vasantdada Patil Pratishthans College of Engineering,, Mumbai, Maharashtra, India*

[4] *Professor, Dept. of Information Technology, Padmabhushan Vasantdada Patil Pratishthans College of Engineering,, Mumbai, Maharashtra, India*

---***---

Abstract - *Video games provide a sense of immersion and interaction that no other form of media does. But often that immersion breaks when elements in the game start repeating itself, which causes loss of interest. This paper reviews the techniques already implemented for countering the repetitive behavior in games. Also, our research deals in researching techniques that efficiently hide the repetitive elements of the game, thus creating an illusion of uniqueness whilst maintaining player immersion and pace of the game.*

Key Words: AI, Video Games, RRA, Immersion, Repetition Reduction, JSON

1. INTRODUCTION

Over the past two decades, how games are perceived has been completely redefined. Instead of being used as a means of recreation in spare time, video games have evolved to encompass massive worlds in which players like to spend hundreds of hours. With every new generation, the complexity, depth, and level of detail in games has increased. Also, the increase in processing power, the quality of graphics, and the increased storage capacity of devices have led to the acceptance of games as mainstream media.

The first set of video games can be traced back to the late ’50s, yet they didn’t get the limelight till the early ’70s. ‘Pong’ was one of the first commercial arcade video games, released in 1972 it was a game that allowed two players to play 2-dimensional table-tennis. It had one governing rule, i.e. control the paddle to pass the ball to the opponent, missing the ball gave the opponent one point. The goal was to score 11 points before the opponent does. Most games in those days revolved around one or two defining rules, while this gave games replay value, it also made games overly repetitive [6].

By the early ’80s games had started becoming more complex in gameplay as well as narrative. Companies like SEGA, Atari, and Nintendo released new powerful home consoles which were affordable by the general populous, this brought in a new array of games to the market. These games now included multiple rules and mechanics, thus making games more dynamic. By the early 90’s role-playing, player freedom and exploration had started getting more emphasis, also games started having more mature content thus attracting a larger audience of varying ages. Storytelling started getting more attention as it gave players a sense of contributing for a larger cause, thus immersing the player even deeper in the game. Games like Metal Gear (1987), Legend of Zelda (1986), Final Fantasy (1987), and The Elder Scrolls Arena (1997) ushered a new way in which games are perceived. Video games became even more diverse when new genres such as FPS, RTS, and MMO emerged due to the transition from raster graphics to 3D graphics [6][7].

Over the years the content in games has increased to allow a more dynamic and varied experience for the player, the graphics have evolved so much so that they are on the verge of achieving photorealism. This improvement in graphics and an increase in the content has resulted in increasing the overall size of games, and even though games have grown in size and content, one thing that has persisted is repetitiveness. One of the goals of our work is to realize when repetition breaks player immersion. Further, we want to design a system that will reduce the number of repetitive gameplay elements and the frequency at which repetition occurs. Also, we want to create a technique that will help in modifying the repetitive elements so that they become dynamic and more immersive whilst maintaining factors such as pace, effectiveness, and player interest.

2. LITERATURE SURVEY

‘Navigational techniques to improve usability and user experience in RPG games’ a paper by Christopher Carter, Qasim Mehdi, and Thomas Hartley study how a mini-map navigation system can be beneficial to improve the gameplay and keeping the user immersed for a longer period which ultimately increasing the longevity of the game. It focuses on how RPG Action-Adventure Games can be beneficial by the use of Mini Maps. The authors focus on improving the end-user experience and the overall gameplay. All of these factors would ultimately result in the longevity of the game [2].

In ‘Developing Games with Elementary Adaptive AI in Unity’ the authors Apostolos Meliones and Ioannis Plas use a Non-Playable Character (NPC) AI System that can perceive changes in the environment and adapt accordingly. The objective of this paper was to create AI enemies in games that can perceive the changes in the environment and adapt their strategies accordingly. It also shows how such a system can be implemented by creating a demo game to test their idea in [3].

In the paper 'Estimation of player's preference for cooperative RPGs using multi-strategy Monte-Carlo method' the author Naoyuki SATO, Kokolo IKEDA, Takayuki WADA design a system where AI Players estimate the goals and sub-goal preferences of the human player and act according to these preferences. They modeled the preferences of sub-goals as a function and decided the most likely parameters by a multi-strategy Monte-Carlo method, by referring to the past actions selected by the team-mate human player. Also, the paper evaluated the proposed method through two series of experiments, one by using artificial players with various sub-goal preferences and another one by using human players. The experiments showed that the proposed method can estimate their preferences after a few games, and can decrease the dissatisfaction of human players [4].

Ramy Taher Makram Wassef, Awad Khalil in 'Increasing Game Immersion through Randomizing Game Characters Appearance' studied how "Character and Object Repetition" in scenes work and comparison of alternative solutions to it. The paper studied various solutions previously discussed in many papers, and how none of these papers mentioned the use of such technique efficiently for 3d models in real-time [5].

3. EXISTING TECHNIQUES

3.1 VISUAL REFERENCE

Visual references such as the mini-map navigation system, in-game markers and pointers sometimes improve the gameplay and keep the user immersed for a longer period, which ultimately increases the longevity of the game, but if not implemented correctly they can also prove to be dull and repetitive. On the other hand, to achieve immersion some games avoid the use of navigation guides and maps and want players to organically discover and traverse the game as they would in real life. Depending on the type of experience that the game is trying to deliver visual referencing strategies and methods can be defined to suit the game immersion.

3.2 GOAL ESTIMATION & PLAYER BASED DECISION

Game AI estimates the goals and sub-goal preferences of the human player and acts according to these preferences. This allows for a more subtle approach to help the decision making of player and guide them through a game. A primary example of this is seen in sports-based games such as FIFA, PES, WWE 2K series, and various other sports-based video games. Here the AI tries to identify how the player will react before the player reacts. Thus providing more challenge & competition to the player.

NPC AI Systems can also perceive changes in the player and adapt accordingly. This will give a more non-linear approach but again limits the dependency factors. This allows for a more non-scripted and dynamic action to be performed. Where NPCs have a list of possible actions to be performed and rather than scripted routine or specific patterns, NPC adapts and changes their actions based on player movement. Even though this system provides a more dynamic approach to NPC, it is still limited as players can only perform pre-defined actions which again reduce the dynamic and non-repetitive nature of the AI, as players can easily identify how NPC behaves and identify and map player actions with NPC actions.

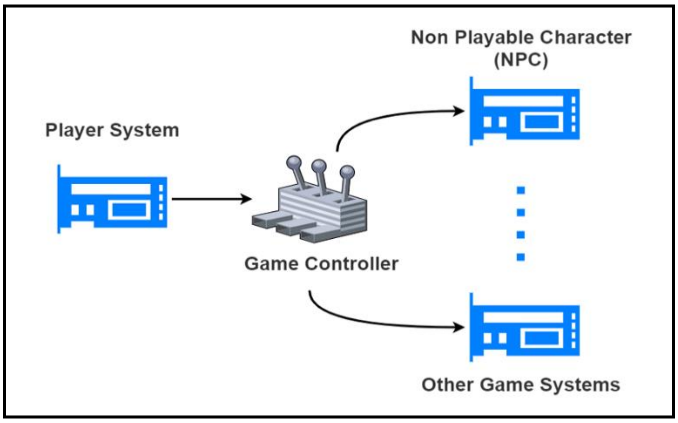


Fig -1: NPC decisions Based on Player Actions

4. ADAPTIVE AI

An adaptive AI approach uses Inter-System Analysis by NPCs where each NPC will share information with other related NPC's to modify their patterns and paths to make them more unique. This will together provide a more vivid experience. This provides more input factors to the NPC and more factors result in more dynamic patterns and increased rate of pattern changing. Inter System analysis allows Developers to implement modules where NPC can analyze the surrounding virtual environment and change their patterns at runtime and to better suit their need of the game. Instead of just reacting to the player's actions, NPC will accept other NPC actions as input and react to their actions. NPC's won't be restricted to a singular input, i.e. player actions but will also consider other NPC actions, giving a sense of a living realistic world.

Another technique used to improve NPC behaviour is Fuzzy (Non-Binary) Decisions in NPC Choices thus making the game experience more dynamic. Fuzzy Logic can also help in getting better reactions by NPC to the actions performed by the player. Instead of having a scripted/pre-set actions that the NPC's will follow, NPC's will abide by s set of rules and use those to make decisions on their own. This makes the game world feel more believable and thus provides a better life-like environment.

These methods however effective are limited by the actions that developers give to a specific NPC and thus after a while can start repeating itself and it may be less perceptive but it will be noticeable and may break player immersion.

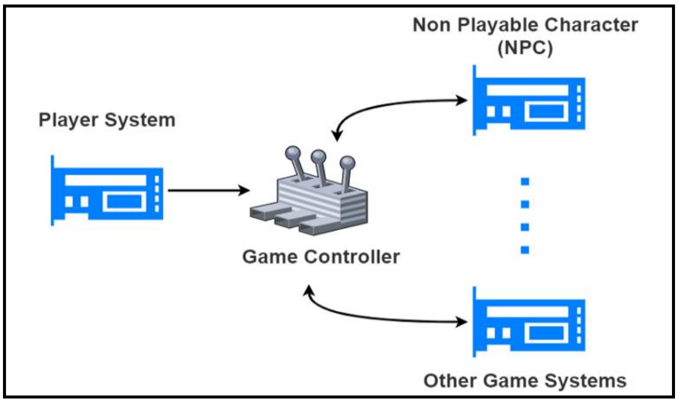


Fig -2: NPC decisions based on Player and other NPC Actions

5. RRA SYSTEM

Now after seeing various available solutions we understand that it is not enough to provide Adaptive AI and Analysis Modules. We must be able to modify and add to the new content of the NPCs and other game systems itself at runtime. We propose this Repetition Reduction Algorithm (RRA) System to reduce repetition, modify the content of the game at runtime, and enhance the immersion and gameplay of the player and provide a better more immersive gameplay. It focuses on the Complexity Adjustment using moment to moment Analysis. It adjusts the difficulty and harshness of the game world as the player plays to better meet player capabilities and increase or decrease the game's difficulty depending on the player's ability. The second algorithm remaps and modifies content and allows dynamic content and modified paths and decision options to the player and NPCs. This causes the change in the path of the player if the same path is re-used over and over multiple times. This will be done in the background without the player realizing that the game is redirecting the player to take alternative actions.

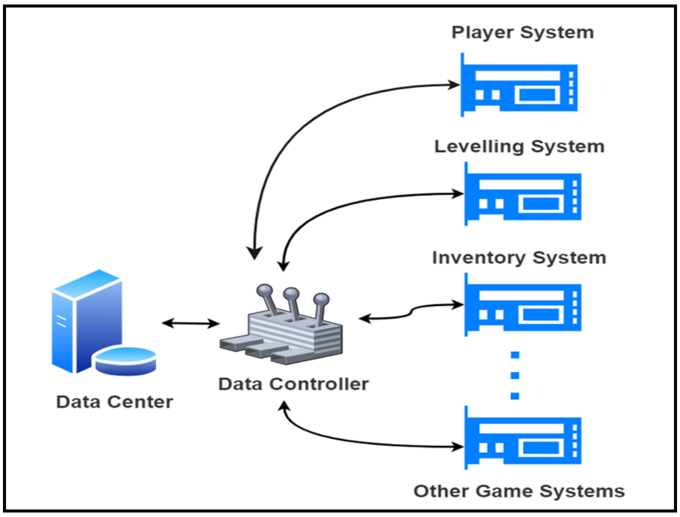


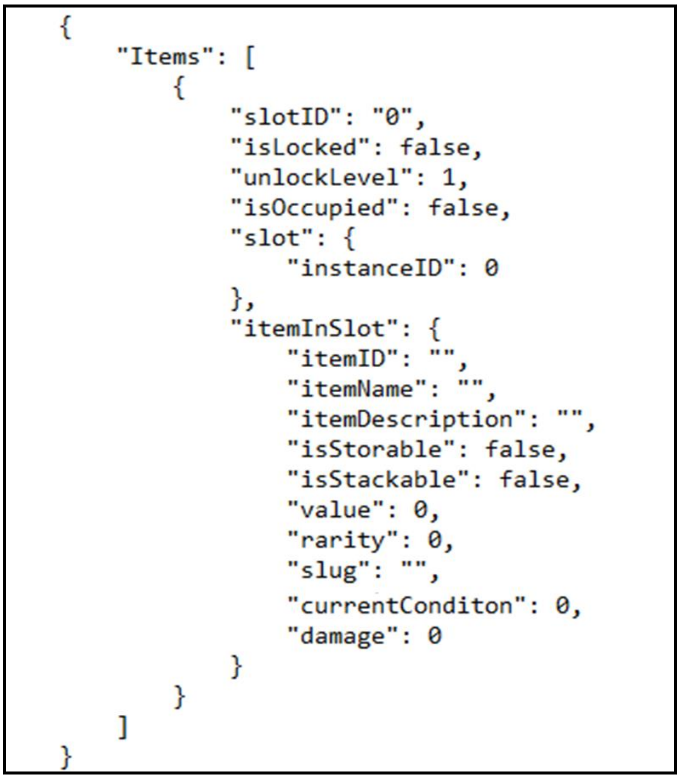
Fig -3: Data Center based Decision Making

To use the RRA Systems we need to identify when repetition takes place. This repetition can be by the player or by the virtual elements of the game. So to track the events that happen in the game we will introduce tracking agents. These agents will monitor various aspects of the game and will keep track of the events that are happening in the game. When these agents when observing a specific event take place which qualifies to modify the content, they will notify the event triggering system to use the RRA system to try and block the repetitive elements in the game and modify the content within the protocols and boundaries defined by the game.

To make the content modifiable by the in-game object, it is necessary for the content readily available to these objects in a single structure. Generally, when designing games every game object maintains its own content but this can cause issues of inconsistent data format. Hence we will store all data within a single entity, which we will call 'Data Center'. This data center will contain all modifiable data as well as modification rules and methods.

5.1 JSON BASED DATA CENTER

In order to be able to modify the data of the game it is essential to maintain same structure for all the data. This must be done so that a single system can be designed which can control all the data of the game. So we designed a structure similar to a database which consists of various files which represent data tables and each file contains various records of different types. We term this storehouse of data as 'Data Center'. Another factor to consider is that the data should be accessible and in easily interpretable format for fast performance. So after some consideration JSON seems to be the best file format in this instance.



```
{
    "Items": [
        {
            "slotID": "0",
            "isLocked": false,
            "unlockLevel": 1,
            "isOccupied": false,
            "slot": {
                "instanceID": 0
            },
            "itemInSlot": {
                "itemID": "",
                "itemName": "",
                "itemDescription": "",
                "isStorable": false,
                "isStackable": false,
                "value": 0,
                "rarity": 0,
                "slug": "",
                "currentConditon": 0,
                "damage": 0
            }
        }
    ]
}
```

Fig -4: Example of JSON Content Structure in Data Centre

The data from JSON files will be extracted at runtime and stored in dynamic list of objects which can then be modified and on quitting the game these objects will again push the updated data into the JSON files; to be reused when game is resumed.

5.2 WORKING OF RRA SYSTEM IN A ROLE PLAYING GAME

When the JSON files are loaded the system will create objects at runtime to make data modifiable and accessible to the game components.

1. All JSON will be parsed into the system into the system.
2. For each record an object of that record type will be created and the values of the parsed record will be stored in these objects.
3. These objects can then be modified and accessed by the game components.
4. RRA System will track these objects and allow objects to be modified by the Game Components within the modification protocols set by the developers.
5. RRA System will also allow new objects to be created at runtime within the purview of the game rules.
6. Before exiting the game all objects will sequentially be converted into JSON Records and updated or added into the respective files.
7. This process will easily allow for game to create new data dynamically at runtime.

This is our idea of how dynamic data can be achieved in the Game. The problem however it will face is that this method only allows text based entities to be created. Game Assets such as 3D and 2D graphic and audio files will still have to be reused. This system will however provide an improvement in the gameplay for the user as the game will keep producing objects with new values, for instance as player levels up the weapons and enemies will level up with the player, thus adding challenge and reason for player to continue playing.

3. CONCLUSIONS

This paper proposes a system that is estimated to increase the total playtime of players by 1.25 to 1.3 times. This will provide an automated and adaptive solution to increase and modify the content of the game. This will help in designing automatic systems that will adjust the gameplay in the backend without breaking player immersion. It will give more dynamic and varied experience to the player without having to add more content to the game. Overall it will also improve the replay value of the game and generating more dynamic outcomes instead of the static binary decisions scripted into the code of the game.

REFERENCES

[1] Dr. Stefan Greuter, Dr. Adam Nash, Game Asset Repetition, 2014 https://dl.acm.org/citation.cfm?id=2677782

[2] Christopher Carter, Qasim Mehdi, and Thomas Hartley, Navigational techniques to improve usability and user experience in RPG games, 2012, https://ieeexplore.ieee.org/document/6314557

[3] Apostolos Meliones, Ioannis Plas, Developing Games with Elementary Adaptive AI in Unity, 2017, https://ieeexplore.ieee.org/document/8324230

[4] Naoyuki SATO, Kokolo IKEDA, Takayuki WADA, Estimation of player's preference for cooperative RPGs using multi-strategy Monte-Carlo method, 2015, https://ieeexplore.ieee.org/document/7317935

[5] Ramy Taher Makram Wassef, Awad Khalil, Increasing Game Immersion Through Randomizing Game Characters Appearance, 2011, https://ieeexplore.ieee.org/document/6000332

[6] https://www.history.com/topics/inventions/history-of-video-games

[7] https://techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/

[8] Campbell, Hank, What Was The First Computer Game? Science 2.0. ION Publications, 2015.