Journal of Advanced Zoology

ISSN**: **0253-7214

*Volume **45** Issue **S-4 Year 2024** Page **38-42***

Enhancing Video Game Quality With Refinement Approaches: A Comprehensive Guide

Ashwini Shivane[1]*

*[1]*Computer Science Department, Indira College of Commerce & Science, Pune, India.*
Email: shivane.ashwini85@gmail.com

***Corresponding Author:** Ashwini Shivane*
**Computer Science Department, Indira College of Commerce & Science, Pune, India.*
Email: shivane.ashwini85@gmail.com

Abstract

The video game industry is expanding rapidly and has great room to grow. There are several steps involved in the creation of a game. The focus of this paper is on various approaches, techniques, and strategies that can be applied at various game development stages. This will result in the rapid, low-risk development of high-quality, personalized products. This strategy will boost player satisfaction in addition to increasing the number of video gamers.

Keywords: CAGR, STRIPS, GOAP, GDD, Game Engine, Game Worlds Graph framework, Cyber Security Game.

I. INTRODUCTION

There are different types of software's used in day to day life to assist, help,facilitate and most important to entertain us.
The video game industry is huge. In fact, it's bigger than the movie and music industries combined, and it's only growing. Although it does not receive the same attention as the film and music industry, there are more than two billion gamers worldwide. This is 26% of the world
The market for video game software worldwide increased at a compound annual growth rate (CAGR) of 14.3% from $228.1 billion in 2022 to $260.83 billion in 2023. At a compound annual growth rate of 13.4%, the video game software market is projected to reach $431.73 billion by 2027.[1]
With this immense opportunity quality gaming practices can meet many needs of gaming companies. We can offer a wide range of personalized services, as well as gaming satisfaction also.
The development of the game is a very creative task & goes through different stages and in these stages we can apply different strategies ,practices to improve quality of it.

II. DIFFERENT STAGES OF GAME DEVELOPMENT

1. Planning
2. Pre-production
3. Production
4. Testing

5. Pre-Launch
6. Launch
7. Post-Launch

The organization of expert workshops provides an opportunity to discuss topics related to best practices in experimental methods and user research, as well as technical advances in the development of innovative games.[2]

1. PLANNING

This is a very important step where you have to predefine the game genre, target audience, development tools, type of graphics, ie. 3D or 2D, technology stack, required budget, etc..

Although there are many different classic design techniques in the academic world, hardly any of them have yet found a real application in the field of commercial video games. Game environments are often very complex and very dynamic.

They are very different from the environments for which classical design methods were originally designed. classical design techniques are based on the following assumptions about the observed system:

a) the system has a finite set of states,
b) the system is fully observable,
c) it is deterministic,
d) it is stationary in the sense that its state can only be changed through the actions of agents known to the designer,
e) the only goals of the agent are achievement goals defined as a goal state, i.e. states to be avoided are not defined,
f) a solution plan is a prescribed finite set of actions,
g) actions have no duration and lead to immediate transitions between states,
h) the planning is done offline, i.e. the designer does not consider all changes in the world state during the design process.

However, most of these assumptions do not apply to games.

Thus, several modifications to these approaches are necessary in game design and other practical design applications.

The Stanford Research Institute Problem Solver (STRIPS) was developed in 1971 at the Stanford Research Institute. Its main purpose was the creation of plans for a robot that should navigate and re-arrange objects in rooms.

STRIPS, implemented in Lisp, belongs to a group of problem solvers that search the space of world states to find a state where the conditions for a given goal are met.

While STRIPS has proven to be a very powerful tool for justifying large search spaces in a classic design environment, F.E.A.R. implemented a modified version of STRIPS to make the designer more controllable, efficient and more suitable for real-time gameplay by abandoning some of the classical design assumptions mentioned in Section II. This resulted in the so-called GOAP (Goal Oriented Action Plan) [3]

2. DESIGN

The game design process can be divided into 3 stages pre-production, production, post production.

1. Pre-production

It involves planning, scheduling, mapping, research and gathering resources for the future production process.

Pre-production is the stage where writers, artists, designers and programmers work together to define the scope of the game. This includes coming up with ideas for gameplay, characters, setting and story.

The creative freedom of the independent sector creates an environment where the community can grow, make mistakes and then learn from them. This in turn promotes talent and offers more to the gaming industry.

Video games are an integral part of the sociocultural landscape and a creative medium of communication between digital producers and users. Because we are creative in our context and culture, the more productions we create, the more our social and personal creative intelligence increases. Independent video game developers can take more risks by doing something different that increases players' interest in creating new content..

AI in gaming enables game developers and studios to collect and analyze data on player behavior. This data can be used to understand how players end up playing, the most popular parts of the game, and what stops users from playing. This information can be used by game developers to enhance gameplay or find legitimate opportunities.

In the not-so-distant past, you couldn't dream of designing a game from scratch without a thorough knowledge of at least one programming language. Now you can simply choose a no-code tool and create a decent game without writing a single line of code..
Some of the tools are mentioned below

- Nuclino — documentation, world building, and planning tool
- Construct 3 — 2D game design software for beginners
- GameMaker Studio 2 — no-code 2D & 3D game design tool
- RPG Maker — JR PG-style 2D game design software
- Godot — free and open-source game engine
- Unity — most popular game engine among smaller studios
- Unreal Engine — A game engine with superior visuals
- ZBrush — the all-in-one-digital sculpting solution

Teams often face difficulties in specifying the core mechanics of the game. They normally write a Game Design Document (GDD), defining the project and its scope, during the pre-production phase. This document is also used to divide tasks and define the artistic designs.[5]
However, writing such document is difficult and requires game development expertise. This document is also rarely updated during the projects' life and the game design visions change regardless of the definitions contained within it.
Unclear game design vision is also caused by the absence of a clear play testing process and by problems in the team, e.g., poor division of tasks and lack of brainstorming. The divergence of creative views between game designers and a publisher is also a common problem
Therefore, time should be taken to properly assign tasks and conduct proper brainstorming sessions leading to improved GDD..In this stage, you have to work on the storyline, levels, rewards, rules, regulations, etc. that will serve users with a delightful experience when playing the game.

Database design
Designing an effective database for a game requires a good understanding of both game development and database design. Here are some important points to note:

1. Define your data: Understand what data needs to be stored. These may include user profiles, game status information, leader boards, in-game content, etc.[4]
2. Choose the right database model: Depending on your game and requirements, you can choose a relational database (such as MySQL or PostgreSQL) if you have complex relationships between entities, a No SQL database (such as MongoDB) for more flexible and salable data storage, or even a combination of both.
3. Normalize your data: In relational databases, normalization is important to avoid redundancy of data and maintain data integrity. But sometimes de normalization can be useful for performance optimization..
4. Indexing: Proper indexing can significantly speed up data retrieval time. Identify the queries your game asks frequently and add indices to those columns.
5. Query optimization: Minimize the number of database calls as much as possible. Also try to keep your queries simple and effective.
6. Scaling: As the game grows, so does the demand on the database. Consider how your database will scale - vertically (more efficient server) and/or horizontally (more servers).
7. Backup and recovery: Make sure you have a solid backup and recovery strategy in case of errors.

Interface design
The importance between the design of the user interface and the gaming experience gives the designers a greater impression that they are thinking about something significant. The effects of game design on user emotions are also very important for user satisfaction with games. So ,interface design plays An important role in game development[7]
Not everyone can play every game, even with the advancements in UI design and the emergence of new gaming genres.
In other words, you can't play the game if you can't use the user interface! Users will choose not to play the game if the controls are too challenging to pick up or use, or if they take away from the fun of the game. This

impact can occasionally be observed when a player gives up in frustration after struggling with the controls more than the game mechanic. Perhaps the largest group of potential players who struggle to interact with U Is are those who are disabled.

The two most popular approaches to game accessibility are either building games with accessibility from the ground up or taking already-inaccessible games and remapping them onto new, accessible U Is..[8]

Architecture design

Game Worlds Graph (GWG) is a framework based on Game World and Connector concepts. It offers a new perspective on the game architecture and reveals some fundamental characteristics of the game architecture.

The GWG framework is not only useful for analyzing existing architectures but also for exploring future architectures. As game architectures have evolved, there have been challenges. Research conducted by researchers and practitioners has identified poor robustness and a lack of flexibility as some of the most significant issues with the client-server architecture pattern.

In contrast, the P2P (peer-to-peer) solution, which researchers primarily proposed as an alternative to the Client-Server, has drawbacks such as the inability to prevent cheating and maintain consistent game states across peers. [9]

3. PRODUCTION

Depending on what needs to work, different GE (Game Engine) programming paradigms can be applied. GE enables the control of artificial intelligence processes (rendering, physics, AI, animation, modeling, network, interface, audio, speech, video, security) as well as graphic effects (3D graphics engines).

Techno genetic evolution of GE from the manual analytic stage, where the engine's functionalities are hard-coded in the source code and the programmer is the "technical individual," to the industrial synthetic stage, where the computational object (in this case, the GE) becomes the "technical individual" proper and the high-level scripting language allows non programmers to easily modify many of its functionalities..[6]

4. TESTING

In the software industry, testing has generally considered a tedious and time-consuming task. As a result, automated testing was widely used to lower costs and raise quality. Employing human testers to manually play game builds at different phases of production is the current practice in the industry for testing games. But test automation is necessary due to the intricacy of video games.

Researchers have looked into human-like, goal-directed, scenario-based, search-based, and model-based automated game testing techniques.[10]

Sports, turn-based, card-based, arcade, simulation, tile matching, MMORPG, puzzle, platform, tower defence, stealth, and GVGAI automated games can all be tested automatically using a search-based method.

A goal-directed technique can be applied to the testing of scene-based, board, tile-matching, and instructional games.

Approach that is human-like: it may be applied to automated testing of adventure, sports, shoot-em-up, puzzle, sandbox, GVGAI, and card games.

scenario-based methodology: applicable to testing Adventure, MMORPG, Sports, and Arcade games

Testing of scene-based, educational, platform, MMORPG, and puzzle games can be done using a model-based method.

These methods allow for the implementation of many AI algorithms [11].

By identifying bugs, improving gameplay, and delivering game analysis in a shorter amount of time and effort, automated game testing can assist in boosting the production of game developers and designers

5. RISK MANAGEMENT

The volatility of the entertainment software industry may necessitate informal project management practices in order to allow for rapid and flexible adaptations and responses to the market.[12][14]

careful risk management is still required to ensure projects meet their success targets.

A technique known as "Cyber Security Game" has been included into software to quantify cyber security risks and use that metric to calculate the best way to apply security measures at any given investment level. Cyber Security Game reduces a system's operational risk to enhance its performance in the competitive cyber world of today.[13]

III. CONCLUSION

This paper has proposed a new approach of defining challenge for video game development. Looking at challenges at different stages of development through its difficulty, proposed different strategies, techniques, methods that can be used at various stages leading to quality improvement of product.

REFERENCES

1. https://www.openpr.com/news/3366175/video-game-software-market-is-going-to-boom-activision
2. P. Mirza-Babaei, N. Moosajee, R. J. Teather, J. Carette and V. McArthur, "Workshop proposal: Game UI design, evaluation, and user research," 2015 IEEE Games Entertainment Media Conference (GEM), Toronto, ON, Canada, 2015, pp. 1-2, doi: 10.1109/GEM.2015.7377260. keywords: {Games; Conferences; Human computer interaction; Software; Industries; Best practices; game user research; HCI; game UIs; methodology],
3. X. Neufeld, S. Mostaghim, D. L. Sancho-Pradel and S. Brand, "Building a Planner: A Survey of Planning Systems Used in Commercial Video Games," in IEEE Transactions on Games, vol. 11, no. 2, pp. 91-108, June 2019, doi: 10.1109/TG.2017.2782846.
4. ehttps://www.dragonflydb.io/faq/game-database-designhe project.
5. Ramadan, Rido, and Yani Widyani. "Game development life cycle guidelines." 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS). IEEE, 2013
6. Charrieras, D., & Ivanova, N. (2016). Emergence in video game production: Video game engines as technical individuals. Social Science Information, 55(3), 337-356.
7. Johnson, D., & Wiles, J. (2003). Effective affective user interface design in games. Ergonomics, 46(13-14), 1332-1345.
8. Ahmad, I., Hamid, E., Abdullasim, N., & Jaafar, A. (2017). Game interface design: measuring the player's gameplay experience. In Advances in Visual Informatics: 5th International Visual Informatics Conference, IVIC 2017, Bangi, Malaysia, November 28–30, 2017, Proceedings 5 (pp. 500-509). Springer International Publishing.
9. Pearson, L. C. (2020). A machine for playing in: Exploring the videogame as a medium for architectural design. Design Studies, 66, 114-143.
10. Albaghajati, A. M., & Ahmed, M. A. K. (2020). Video game automated testing approaches: An assessment framework. IEEE transactions on games.
11. C. Salge, C. Lipski, T. Mahlmann, and B. Mathiak, "Using genetically optimized artificial intelligence to improve gameplaying fun for strategical games," in Proceedings of the 2008 ACM SIGGRAPH symposium on Video games. ACM, 2008, pp. 7–14.
12. Schmalz, M., Finn, A., & Taylor, H. (2014, January). Risk management in video game development projects. In 2014 47th Hawaii International Conference on System Sciences (pp. 4325-4334). IEEE.
13. Musman, S., & Turner, A. (2018). A game theoretic approach to cyber security risk management. The Journal of Defense Modeling and Simulation, 15(2), 127-146.
14. Lichtenvoort, D. (2014). Gamification: video game element incentives in cross-functional enterprise information system-project finance and risk management. diplom. de.