Development Project 55-608850 Pursuit - Procedural Dungeon Generation in 2D Development

Author: Jamal Ahmed
Student ID: C1051264
Supervisor: Temitope Adeosun, Jaya Tangiraka
Word Count: 3120

Abstract

This academic report showcases the in depth design, development and evaluation of a 2d dungeon crawler called “Pursuit”. The game utilizes procedural generation to increase player engagement and player replay-ability. At the heart of the games structure, random walk algorithm implemented with Unity2D framework, this aids in generating diverse and difficult game levels. The study uses Bloom’s Taxonomy framework for structuring and showcasing the literature review. The project also implements agile methodologies tailored to solo development, making sure there is a iterative and adaptable design process. Structured testing paired with questionnaires provide insight to player satisfaction and locate any key issues or usability errors. The key finding highlight the games success in keeping players interested and the theory of its technical implementation.

Table of Contents

Introduction

Problem Statement

The primary challenge that is going to be addressed in the development of “Pursuit”, involves producing a dynamically engaging 2D dungeon crawler. This will be achieved by using procedural generation, dynamic difficulty adjustment and applying best practices. “Pursuit” seeks to overcome common mistakes and design flaws – such as repetitive and monotonous level design, as this can diminish and lose player interest quickly.

Summary of Main Findings

The report specifically explores the implementation and evaluation of random walk algorithm within Unity2D game engine. Utilizing these tools and methods to create challenging game levels. This academic report discusses the theoretical and practical application of the procedural generation using the random walk algorithm, the development using agile game development, ethical considerations, evaluation of “Pursuit through structured testing and finally in-depth analysis to determine the success of the project; further insight into future improvement.

Literature Review

This literature review examines academic and practitioner literature focusing on game design and development. With a further inspection on procedural generation within 2D development and similar game mechanics. The review is built upon structured selection, analysis and synthesis that support the development of “Pursuit”.

This review is also built around Bloom’s Taxonomy framework. This framework was recommended by my supervisor to categorize the levels of abstraction and complexity in the reviewed research. Bloom’s Taxonomy helps produce critical examination and analysis of documents, organizing and guiding my findings into a spectrum. This ranges from basic understanding to complex procedures such as analysis, synthesis, evaluation and application. The frameworks methodological approach, aids this review to showcase structured understanding of procedural generation in 2D development.

Search and Selection

The focus was to produce a comprehensive academic and practitioner sourced literature, that accounted key components of game design and development. With a main focus on procedural generation algorithms, player engagement, and the use of the Unity2D framework.

Search Strategy

Most of the research was done through academic databases such as Google Scholar. In addition, key industry publications and technical blogs were examined to incorporate contemporary practices and innovations in game development.

The conducted searches used a combination of keywords and phrases that relate to the project. This includes "procedural generation", "Unity2D game development", "dynamic dungeon layout", "player engagement in video games”, "algorithm efficiency in game design" and similar prompts.

The selection criteria focused on research done in the last 15 years. This ensured that the papers, books or articles are relevant to current technology and methodologies. Of-course preference was given to papers and articles that discussed procedural generation in 2d development. Documents with more citations were also prioritised.

Selection Process

Titles and abstracts were initially screened for relevance to the selection criteria and project scope. Mainly focusing on procedural generation in Unity2D , if not, in general 2D game development. Then full texts of selected documents where read and reviewed. I then filtered the documents based on its contribution to the design, development and understanding to the project scope.

Finally the selected literature were then studied comprehensively and in-depth. Through analysis and comparison, insightful understanding within procedural generation, 2d development and similar game mechanics. Papers have also been further chosen based on their methodological rigour and their relevance to the design, development and scope of the project.

Analysis of Sources

2D Procedural Generation Algorithms

Summary and Main Findings

Koesnaedi & Istiono's (2022) paper showcases the implementation of the Drunkard's Walk algorithm in generating random levels for a rogue-like game. The design and development process of the game named "Monster" Cave using Unity2D game engine. Also Player satisfaction measurement using GUESS assessment, and suggestions for future improvements in the game design. Similarly Goandy's (2020) paper discusses the importance of creating engaging content in video games, particularly through the use of the rogue-like genre and Procedural Content Generation. Just like Koesnaedi & Istiono (2022), Goandy (2020) presents a successful implementation of the Drunkard Walk algorithm in a 2D Top-Down Shooting game, resulting in a good level of player satisfaction. The study finds that this algorithm, while simple, effectively contributes to creating a challenging and dynamic game environment that adapts to different play styles.

Almgren's et al. (2014) paper took a slightly different angle to Koesnaedi & Istiono (2022) and Goandy (2020) to produce a rogue-like genre. While similarly they also focused on procedurally generated levels and importance and benefits of procedural content generation in game development. What was different was the implementation of Procedural plot generation this created varied and unique content efficiently. The authors also expressed their satisfaction with their work and mention future work ideas for further development.

Methodologies

Again comparing Koesnaedi & Istiono's (2022) and Goandy (2020) methodologies they are very similar. Koesnaedi & Istiono's (2022) and Goandy (2020) build the game with the Drunkard Walk algorithm. They also both use Unity2d with C#, testing with players, measuring player satisfaction with GUESS, data evaluation, and documentation. Specifically Goandy (2020) conducts a player satisfaction survey using GUESS, testing with a minimum of 30 players, and administering a 51-question Likert Scale questionnaire. Koesnaedi & Istiono (2022) also includes a literature review and game design aspects.

On the other hand Almgren's et al. (2014) methodologies used in the study involved agile development techniques, specifically a variation of the Scrum methodology. The team was divided into smaller groups

working on different aspects of the game each week. Java was chosen as the programming language due to team familiarity, and the Lightweight Java Game Library was used for graphics. Git and GitHub were used for version control, which worked well for an open-source project.

Analysis

All three papers drive home the point that procedural generation results in gameplay improvement, though the methodologies and game genres are different. The common point among all is this emphasis on increasing player engagement and replay-ability in games via non-repetitive and challenging game worlds. Valuable and relevant to this discussion is the suitability of a modified random walk-type algorithm (similar to the Drunkard’s Walk) for the project "Pursuit". This shows that algorithms presented here is a crucial component to the development of complex and engaging game environments; a perfect fit for increasing variability and complexity of "Pursuit" gameplay.

Next Step

The above discussion sets a very good basis in and by itself for the justification of the Drunkard's Walk-like algorithm in "Pursuit". The research will continue by synthesizing how those support and sharpen the proposed solution; particularly, how it influences the design and user experience of the game. In this way, there will be further bridging of theoretical research to practical application in game development.

Dynamic Difficulty Adjustment

Summary and Main Findings

View of Dynamic Difficulty Adjustment (2024) – this paper investigates how dynamic difficulty adjustment is incorporated into games with procedural level generation by using the Markov Decision Process. The special focus is directed toward understanding how dynamic difficulty adjustment, based on real-time input from the player, could be utilized to advance the engagement and retention of a player. It was found that, when applied correctly, the use of a Markov decision process to guide procedural level generation can significantly improve player satisfaction through the maintenance of optimal challenge levels.

Contrary to this Hivana's et al. (2013) paper focuses on multi-criteria decision-making on cloud services, it gives basic concepts for constraint satisfaction programming, which can readily be applied in dynamic difficulty adjustment in games. The authors are able to discuss this in terms of how constraint satisfaction can be effectively applied in tailoring recommendations or changes to specification criteria, where in terms of gaming, this would be the criteria that specify the tailoring of game parameter adjustment to the skill level and preference of the player.

Methodologies

View of Dynamic Difficulty Adjustment's (2024) methodology is highly quantitative in nature, as the approach has to consider the gameplay performance metrics and incorporates them into a Markov decision process. However Hivana's et al. (2013) methodology adopted to a gaming context with constraint satisfaction programming uses predefined and real-time data collected from the player and game to dynamically adjust the game parameters.

Both *View of Dynamic Difficulty Adjustment* (2024) and Hivana et al. (2013) methods involves intensive player testing for validation in order to confirm effectiveness of the dynamic adjustments toward maxing player satisfaction and retention.

Analysis

Both papers underline the fact that the best methods to increase player engagement involve the use of sophisticated algorithms that offer more unique game experiences. However, relevance to the chosen solution,

motivated on both theoretical backing and practical experience, is Hivana’s et al. (2013) method. The use of constraint satisfaction programming for dynamic difficulty adjustment in “Pursuit”. The Approach is expected to enhance the adaptive responsiveness of the game. This also goes hand in hand to create a ever unique gameplay experience.

Next Step

this analysis reinforces and justifies the implementation of dynamic difficulty adjustment into “Pursuit” using constraint satisfaction programming. Personally I have found constraint satisfaction programming to be very effective in the past.

Development Methodology

Summary and Main Findings

Kortmann & Harteveld (2009) explain in their paper how the adoption of agile methodologies within game development mimic software engineering practices. Their paper also stresses the importance of Agile's iterative cycles, feedback and adaptability are beneficial in a project that undergoes frequent requirement changes. Similarly, *Game* (2023) is a document developed regarding the adoption of agile in game development with a significant interest in rapid iterations, flexible planning, and constant feedback.

Methodologies

Both Kortmann & Harteveld (2009) and *Game* (2023) discuss agile methodologies that are supposed to be used during game development to progressively improve the quality of the game. Kortmann & Harteveld's (2009) paper outlines quick and short development cycles and regular meetings to enable quick adaptation. *Game* (2023) sets scrum practices in place. Such as sprint weeks, daily stand-ups, sprint reviews and much more. The agile game methodology is designed to make development process easier by being effective in assisting the dynamic changing nature of game development.

Analysis

After analysing both papers, Kortmann & Harteveld (2009) and *Game* (2023) showcase the strong points of agile development. The positives mentioned many times before provide the managing complexity required to tackle the design and development of "Pursuit".

Next Step

These insights confirm the suitability of the Agile methodology for the "Pursuit" development. Next steps include formalizing Agile processes as a solo developer for continuous improvement. This approach ensures the project adheres to agile game development best practices, optimizing development efforts to enhance player experience.

Synthesis of Insights

The research conducted across various domains in game development - 2D procedural generation algorithms, dynamic difficulty adjustments and development methodologies – provide a strong foundation and blueprint for the structure of “Pursuit”. The findings also allow me to deliver a robust gaming experience that aligns with current technologies.

Procedural Generation algorithm

The insight from Koesnaedi & Istiono (2022) and Goandy (2020) guide me to use random walk algorithm within Unity2D, which is a platform that I am both experienced in and familiar with. This algorithm will be important in generating varied and complex room layouts by initiating random rooms. The algorithm will produce rooms by using multiple random paths from the centre of the room. Rooms will be linked together using connected rectangular corridors, however these corridors will be randomly placed; rooms will then be placed at ends and/or at intersections. This altered method to Koesnaedi & Istiono (2022) and Goandy (2020) capitalizes on my prior experience and also utilizes the procedural content generation, with in turn enhances and adds replay-ability to “Pursuit”.

Dynamic Difficulty Adjustment

Using the principles found from both *View of Dynamic Difficulty Adjustment* (2024) and Hivana et al. (2013) showcase that dynamic difficulty adjustment will increase player retention and satisfaction because it will adapt continuously to above the players level of expertise. The process that I have chosen will be similar to Hivana et al. (2013), where constraint satisfaction programming will dynamically adjust player upgrades, enemy strength and game challenges by considering the players initial stats and current real time stats.

Agile Game Development

Kortmann & Harteveld (2009) and *Game* (2023) will aid in shaping the development strategy for “Pursuit”. As a solo developer, the agile framework will be altered to adapt to individual development. I will implement methods such as regular iteration, breaking down tasks into smaller chunks, feedback cycles and adaptable planning. This paired with project management strategies from Cadle & Yeates (2008), which

highlight efficient management techniques for managing projects of any size. This combined methodology of agile development technology and PRINCE2 project management technology will ensure that the development process remains highly adaptable and responsive. These advantages aid in overcoming new challenges.

Summary

In summary the integration of these tools, technologies and methodologies provide insights into the design and development of “Pursuit”. Not only do these insights provide best practices but also newest technologies and innovation. Finally, by combining the procedural generation algorithm, agile practices, adaptive difficulty adjustment, project development and other technologies (such as Git with GitHub for version control) set a clear pathway for designing and developing “Pursuit”. This approach guarantees that the project will be innovative and deliver on the aim of increasing player satisfaction.

Legal, Social and Ethical Issues

During the development of “Pursuit”, legal, social and ethical issues was considered. Mainly in relation to intellectual property rights and copyright infringement. While developing, free to use assets were used to avoid any issues. Also, although concerns were considered for showcasing certain game elements (such as a fireball), they were found to be deemed too minor to provide any influence to the projects integrity.

Evaluation of the Finished Product

Evaluation strategy

The evaluation strategy of “Pursuit” was quite simple. Firstly I used industry best standards to collect and analyse user experience. Specifically conducted remote play-testing was a the primary choice for evaluation. However post-play questionnaires as follow-up allowed me to create comprehensive evaluation, essentially allowing me to place numbers and themes to findings.

Quality Assessment Process

Play-testing Procedure

The play-testing process involves the player remotely playing “Pursuit”. Participants of the session will fully play test the game, including all different types of skill levels and experience within the gaming scene. This will allow me to make an accurate understanding on the impact and functionality on people from different gaming backgrounds. The players from the tests also will be requested to record their gameplay during each session. This adds extra benefits, as players interact with the game, unexpected events could appear; because I have recorded their gameplay, it will be easy for me to recreate the scenario.

Questionnaire

Once gameplay is finished, the player is asked to populate a questionnaire via Google Forms. This tool captures their feedback on the complexity and views on “Pursuit”, specifically the level design, effectiveness of featured game mechanics and overall engagement. The survey includes both quantitative scales and open-ended questions to gather detailed player feedback.

Analysis of Player Feedback

Procedural Generation Feedback

Using further analysis and inspection player feedback will shed light on how the random walk algorithm impacts or impactless it is towards their gameplay experience. Particularly regarding dungeon layouts and variation. This also helps gauge, to what extent is the gameplay going to

be different and how many gameplay loops until the experiences become similar. This feedback is crucial to the development of the procedural generation algorithms success.

Overall Gameplay Satisfaction

The questionnaire highlights the critical features and designs that the player hates and enjoys. Here the design and development can shift to looking into improving and changing future deliverables to improve the gameplay. This covers the feedback received on gameplay, game mechanics, level design, UI/UX design and any other intuitive features of the game.

Reflection and Improvement

During the feedback and evaluation phase, several key improvements were identified. Using 5 testers to under go the session, as Myers's et al. (2012) states that 5 users can uncover 80% of usability errors, we optimized our testing to functional testing (using visual recording to repeat those errors again). This strategy proved efficient and effective, allowing us to quickly adjust and improve any errors that occurred.

Key areas of improvement

Optimization of procedural generation

feedback showed that the procedural generation indicated the occasional repetitiveness in dungeon layouts. The solution to this was to create more varied variables such as enemies and props to be more intuitive. And also look into different algorithms for different or higher levels, such as Binary space partitioning.

Interface Enhancements

Players suggested that UI/UX elements and similar features could be more intuitive. The solution to this would be providing a options section to alter key binds, audio and other features.

Lack of more Game mechanics

Players complained at the players lack of game mechanics, as only a ability system and basic game mechanics were only implemented. The

solution to this would be providing maybe a inventory system paired with enemy drop system.

Questionnaire Overview and Responses

Questions Included in the Questionnaire

Many questions are not stated here (such as “what character do you think is strongest?” - gameplay balancing question) as they are not relevant the aim of the project.

1) How would you rate the experience playing “Pursuit”? (scale 1-10)
2) Did you find “Pursuit” level design to be challenging?(Yes/No)
3) What aspect of the game did you like the most?
4) Where there any section that felt repetitive or frequent?
5) What feature would you think would improve “Pursuit”?

Themes, Numbers and Responses

1) Overall satisfaction was rated highly at 76%, this indicates a successful initial deployment
2) Majority of players found the gameplay to be easy, which lead to many play-throughs. A very small minority found the game to be difficult.
3) Many players enjoyed the initial encounters with gameplay aspects, we could assume more content/ variables could increase player satisfaction further.
4) Due the ‘easy’ nature of the game many play-throughs were established, an effect of this was repetitiveness. There are many solutions to this, one being increase the difficulty of the gameplay; another would be increase the variables and/or content.
5) Majority suggested improvements in user interface and design, a couple asked for more game mechanics. Lastly the minority indicated for harder level design.

Note on Exclusion of Dynamic Difficulty Adjustment

While initially discussed, dynamic difficulty adjustment was not fully implemented due to its complexity, recourse requirements and large scope. However, the foundations, such as the ease of game design features like quick creation of game objects and variables. Dynamic difficulty adjustment can be implemented later.

Conclusion

This academic report include research, design, development and evaluation of the procedural generation algorithm, player engagement and other game mechanics. Key finding reveal that the random walk algorithm successfully enhances level design and player engagement. Overall the user satisfaction was 76% suggesting the fact the there was correct game design and development when building “Pursuit”

References

Almgren, S., Anttila, L., Oskarsson, D., Sörensson, F., & Tiensuu, S. (2014). *Astrogue: A Roguelike Using Procedural Content Generation for Levels and Plots in a Computer Game*. https://odr.chalmers.se/server/api/core/bitstreams/b70c5ec5-8d75-460d-81ca-0d1dce1688c9/content

Cadle, J., & Yeates, D. (2008). *Project Management for Information Systems Project Management fi fth edition*. https://www.greatertzaneen.gov.za/documents/news/Project%20management%20for%20information%20system%205th.pdf

Game, A. (2023). *Agile Game Development: Build, Play, Repeat, 2nd Edition*. O'Reilly Online Learning. https://learning.oreilly.com/library/view/agile-game-development/9780136204831/?sso_link=yes&sso_link_from=sheffield-hallam-univ

Goandy, H. (2020). No Escape: A 2D Top-Down Shooting Roguelike Game Embedded with Drunkard Walk Algorithm. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(2), 1045–1049. https://doi.org/10.30534/ijatcse/2020/22922020

Hivana, A., Medeiros De Macedo, Da Silva, C., & Ferreira, L. (2013). *Simpósio Brasileiro de Pesquisa Operacional A Comparative Analysis of Multicriteria Approaches for Cloud Services Selection*. https://www.din.uem.br/sbpo/sbpo2013/pdf/arq0347.pdf

Koesnaedi, A., & Istiono, W. (2022). Implementation Drunkard's Walk Algorithm to Generate Random Level in Roguelike Games. *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, *5*(2), 97–103. https://ijmrap.com/wp-content/uploads/2022/07/IJMRAP-V5N2P27Y22.pdf

Kortmann, R., & Harteveld, C. (2009). *Agile game development: lessons learned from software engineering*. https://web.northeastern.edu/casperharteveld/pubs/Agile_ISAGA2009.pdf

Myers, G. J., Badgett, T., & Sandler, C. (2012). The Art of Software Testing. *Wiley EBooks*. https://doi.org/10.1002/9781119202486

View of Dynamic Difficulty Adjustment via Procedural Level Generation Guided by a Markov Decision Process for Platformers and Roguelikes. (2024). Aaai.org. https://ojs.aaai.org/index.php/AIIDE/article/view/27540/27313