

# PROCEDURAL GENERATION OF TOWER DEFENSE MAPS WITH MEASURABLE AND CONTROLLABLE DIFFICULTY

---

Master in Informatic and Computing Engineering  
Dissertation Preparation

Carlos Miguel Sousa Vieira (Author)  
Alexandre Miguel Barbosa Valle de Carvalho (Supervisor)

# LIST OF CONTENTS

- Context and motivation
- Hypotesis
- State of the art
  - Tower Defense Games
  - Procedural content generation in games
  - Automatic difficulty evaluation
    - Automated playtest
    - Difficulty estimation
- Solution Perspective
- Solution Workflow
- Workplan
- References

# CONTEXT AND MOTIVATION



THE GAMING INDUSTRY HAS  
BEEN GROWING A LOT OVER  
THE PAST FEW YEARS



PLAYERS DEMAND FOR MORE  
AND BETTER CONTENT



CONTENT PRODUCTION IS  
SLOW AND EXPENSIVE

# HYPOTESIS

To study a solution for automatic content generation and processes to evaluate the difficulty of generated content in order to present the user sequences of maps with consistent increasingly higher degrees of freedom.

This study will be directed to maps for games of the Tower Defense sub-genre.

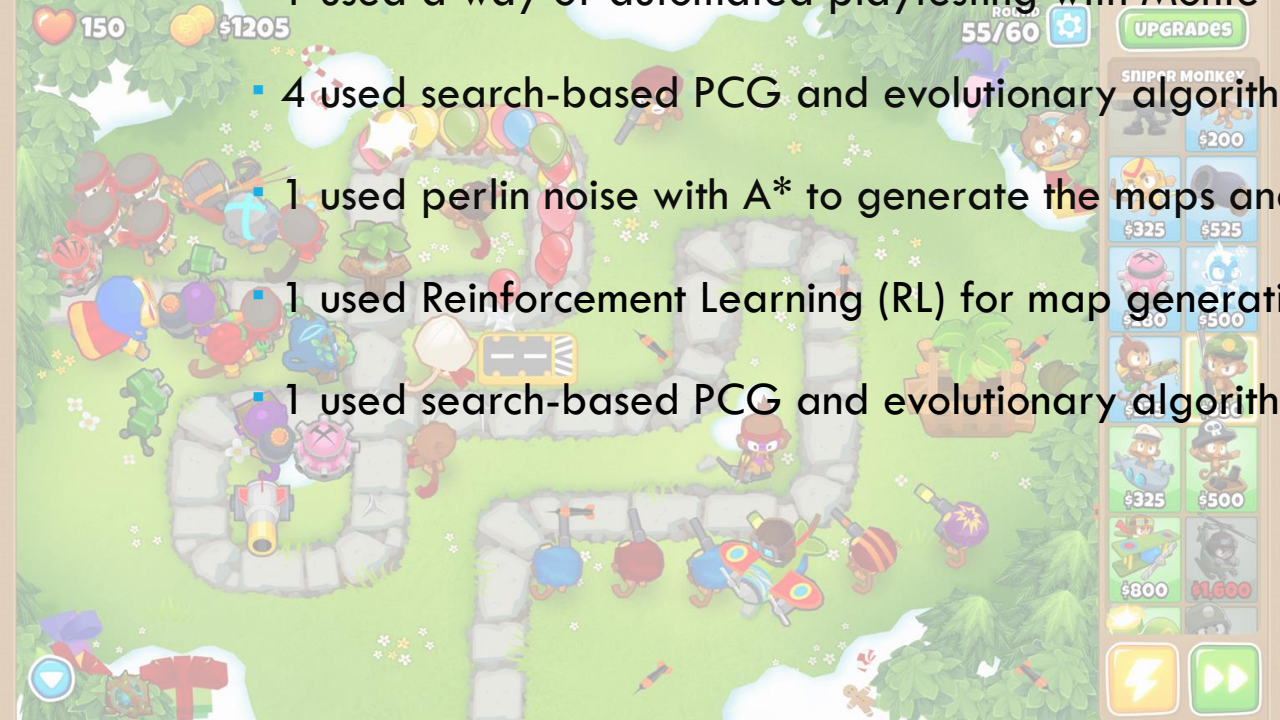


# STATE OF THE ART

# TOWER DEFENSE GAMES

7 surveyed studies show that:

- 1 used a way of automated playtesting with Monte-Carlo Tree Search (MCTS) [6].
- 4 used search-based PCG and evolutionary algorithms to generate and evolve enemy waves [2, 4, 5, 6].
- 1 used perlin noise with A\* to generate the maps and paths [1].
- 1 used Reinforcement Learning (RL) for map generation [3].
- 1 used search-based PCG and evolutionary algorithms for map generation [4].



# PROCEDURAL CONTENT GENERATION IN GAMES

- Fractals and noise [8]
- Constructive generation [8]
- Search-based generation [8]
- Answer-set programming [8]

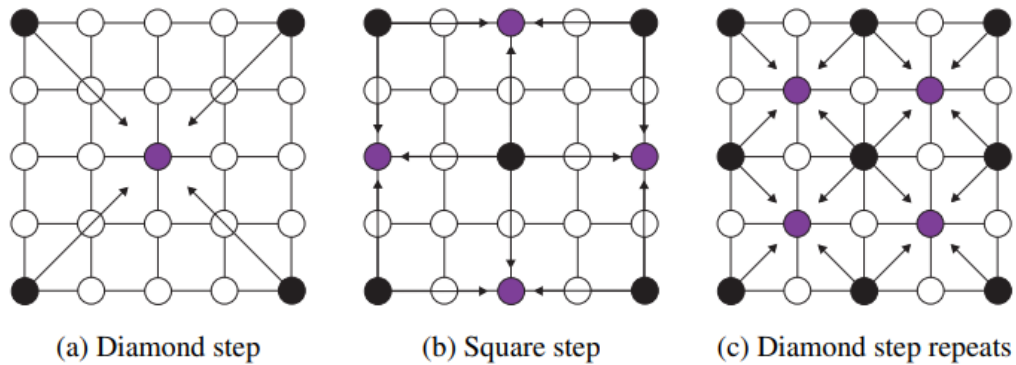


Fig1 – Illustration of the diamond-square algorithm [8].

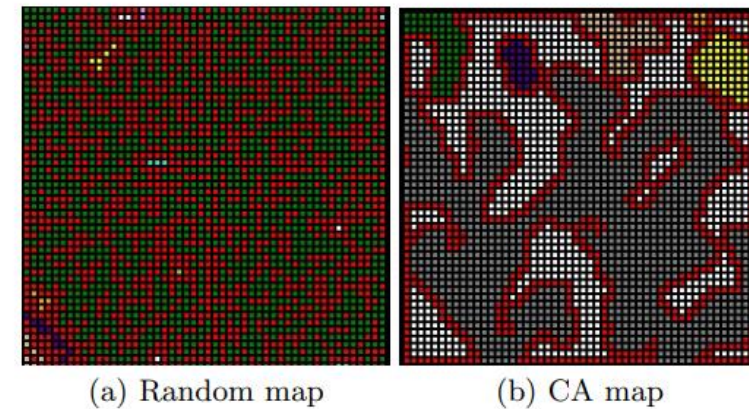


Fig2 – Comparison between a random map and a cellular automata generated with 5 iterations [8].

# AUTOMATED PLAYTEST

## Goals [9]

- Functional correction
- Multiplayer stability
- Performance
- Visual correctness
- Game design correctness
- Game balance and fairness
- Progression and learnability
- Physical correctness

## Approaches [9]

- Search-based
- Goal-directed
- Human-like
- Scenario-based
- Model-based

# AUTOMATED PLAYTEST

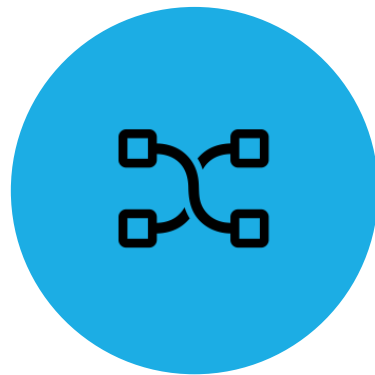
Approach	Implementations	Testing Objectives	Game Genres
Search-Based	<ul style="list-style-type: none"> <li>• <b>Evolutionary Algorithms:</b> [16, 30, 31, 32, 39]</li> <li>• <b>Graph search:</b> <ul style="list-style-type: none"> <li>• Depth first: [41]</li> <li>• A*: [22, 35]</li> <li>• MCTS: [33, 34, 40]</li> <li>• Cicero: [42]</li> </ul> </li> <li>• <b>RRT:</b> [36, 37, 38]</li> </ul>	<ul style="list-style-type: none"> <li>• Functional correctness</li> <li>• Game design correctness</li> <li>• Game balance and fairness</li> <li>• Progression and learnability</li> </ul>	Sports, Turn-based, Cards, Arcade, Simulation, Tile-matching, MMORPG, Puzzle, Platform, Tower defense, Stealth, GVGAI
Goal-Directed	<ul style="list-style-type: none"> <li>• <b>Reinforcement learning:</b> [47, 48, 49, 50]</li> <li>• <b>Restricted heuristics:</b> [51, 52, 53]</li> </ul>	<ul style="list-style-type: none"> <li>• Functional correctness</li> <li>• Game design correctness</li> <li>• Game balance and fairness</li> <li>• Progression and learnability</li> <li>• Physics correctness</li> <li>• Performance</li> </ul>	Scene-based, Board, Tile-matching, Educational
Human-Like	<ul style="list-style-type: none"> <li>• <b>Machine learning:</b> [54, 55, 20, 56, 57]</li> <li>• <b>Restricted heuristics:</b> [58, 59, 60, 61, 62]</li> <li>• <b>Mixed algorithms:</b> <ul style="list-style-type: none"> <li>• MCTS and Stratabots: [63]</li> <li>• MCTS and Evolutionary Algorithms: [64]</li> <li>• MCTS and Machine Learning: [65]</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Functional correctness</li> <li>• Game design correctness</li> <li>• Game balance and fairness</li> <li>• Progression and learnability</li> </ul>	Sports, Shoot 'em up, Tile-matching, Sandbox, Puzzle, GVGAI, Cards, Adventure
Scenario-Based	<ul style="list-style-type: none"> <li>• <b>Record and Replay:</b> [66, 67]</li> <li>• <b>Game simulation:</b> [68, 69, 70]</li> <li>• <b>Visual debugging:</b> [71, 72, 73]</li> </ul>	<ul style="list-style-type: none"> <li>• Functional correctness</li> <li>• Game design correctness</li> <li>• Visual correctness</li> <li>• Multiplayer stability</li> </ul>	Adventure, MMORPG, Sports, Arcade
Model-Based	<ul style="list-style-type: none"> <li>• <b>Petri nets:</b> [74, 75]</li> <li>• <b>Unified Modeling Language:</b> [76, 77]</li> <li>• <b>Game Description Language:</b> [80]</li> <li>• <b>Linear Temporal Logic:</b> [81]</li> <li>• <b>ModelMMORPG:</b> [82, 83]</li> </ul>	<ul style="list-style-type: none"> <li>• Functional correctness</li> <li>• Multiplayer stability</li> </ul>	Educational, Platform, Scene-based, MMORPG, Puzzle

TABLE I  
SUMMARY OF THE STUDY FINDINGS [9]

# DIFFICULTY ESTIMATION



SELECT VARIABLES<sup>[10]</sup>

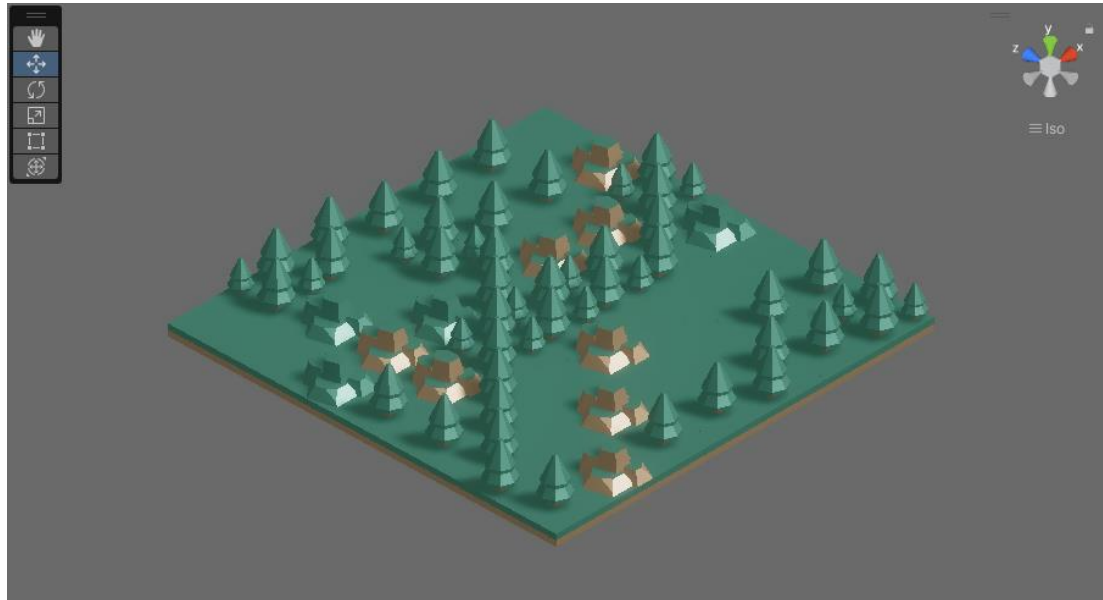


CHOOSE A MODEL<sup>[10]</sup>



DETERMINE  
WEIGHTS<sup>[10]</sup>

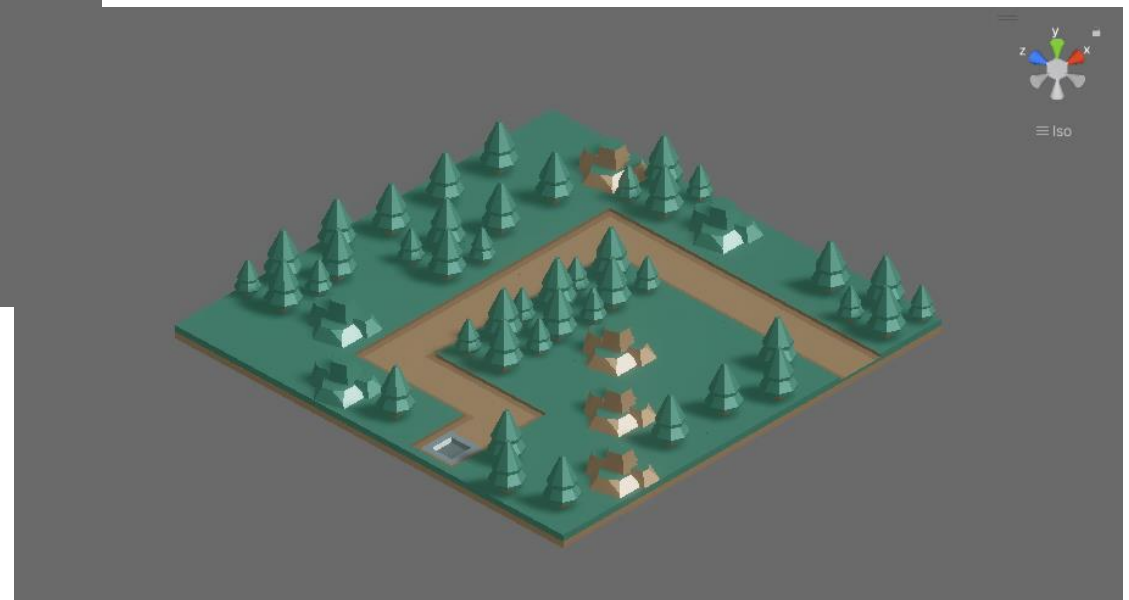
# SOLUTION PERSPECTIVE



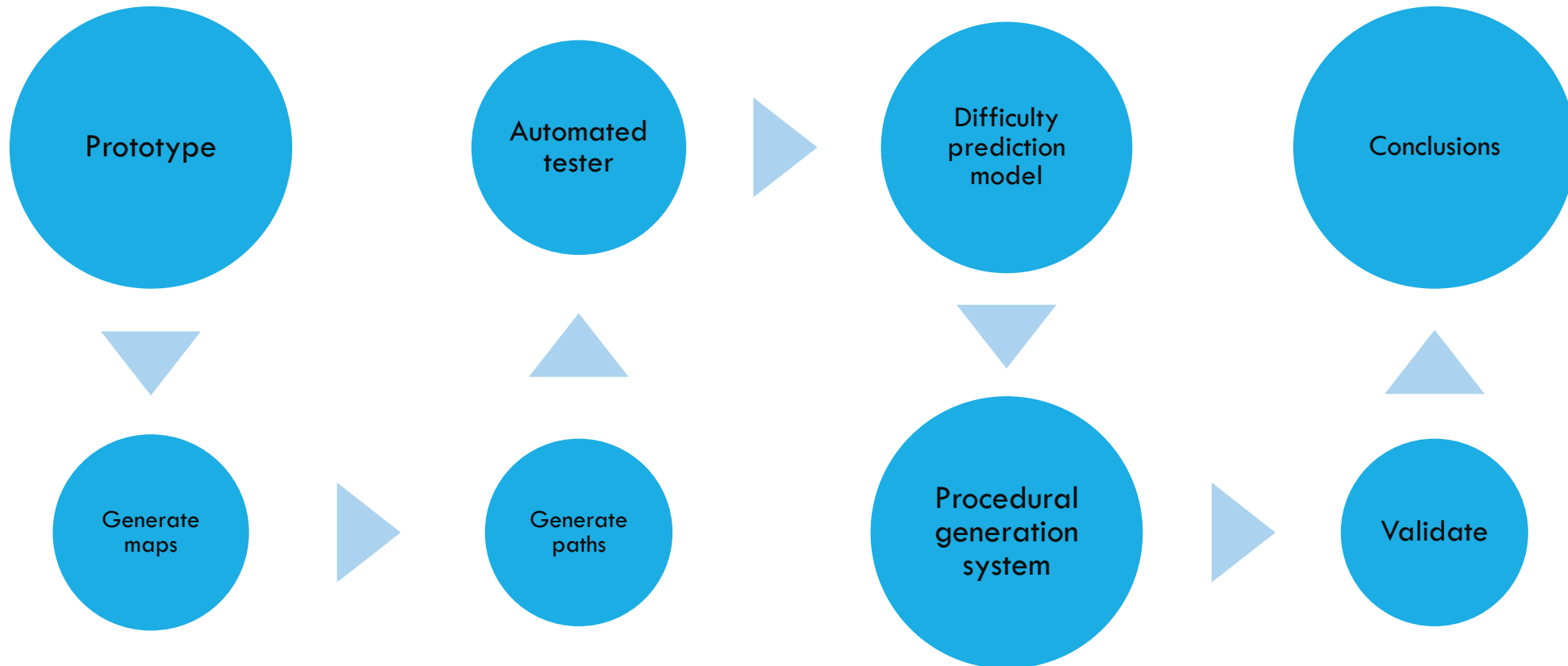
Background terrain generation



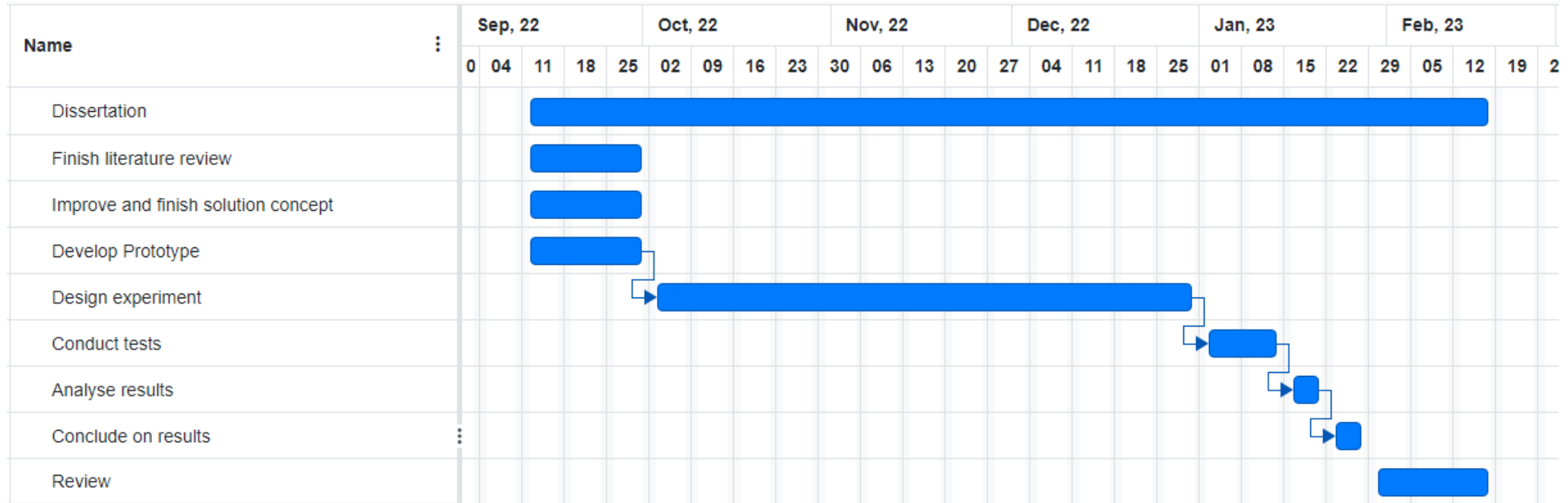
Path generation



# SOLUTION WORKFLOW



# WORKPLAN



# REFERENCES

- [1] – Budiono, M., Liliana, L., & Juwiantho, H. (2022). Meningkatkan Kesulitan Serangan Enemy Dengan Menambahkan Influence Map Pada Metode A\* Pada Procedural Generated Tower Defense Game. *Jurnal Infra*, 10(1), 210-216.
- [2] – Du, Y., Li, J., Hou, X., Lu, H., Liu, S. C., Guo, X., ... & Tang, Q. (2019, March). Automatic level Generation for Tower Defense Games. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 670-676). IEEE.
- [3] – Yueming, X., & Tetsuro, T. (2021). Procedural Content Generation for Tower Defense Games: a Preliminary Experiment with Reinforcement Learning. *ゲームプログラミングワークショップ 2021 論文集*, (2021), 93-97.
- [4] – Kraner, V., Fister, I., & Brezočnik, L. (2021, June). Procedural Content Generation of Custom Tower Defense Game Using Genetic Algorithms. In International Conference “New Technologies, Development and Applications” (pp. 493-503). Springer, Cham.
- [5] – Ariyadi, S. M. S., Purnomo, M. H., & Nugroho, S. M. (2014, September). Creep Offenses Evolution in Tower Defense Games using NSGA-II. In Proceedings of 8th International Conference on Information & Communication Technology and Systems (ICTS) (pp. 153-157).
- [6] – Liu, S., Chaoran, L., Yue, L., Heng, M., Xiao, H., Yiming, S., ... & Qinting, T. (2019, August). Automatic generation of tower defense levels using PCG. In Proceedings of the 14th International Conference on the Foundations of Digital Games (pp. 1-9).
- [7] – Tan, T. G., Yong, Y. N., Chin, K. O., Teo, J., & Alfred, R. (2013, August). Automated evaluation for AI controllers in tower defense game using genetic algorithm. In International Multi-Conference on Artificial Intelligence Technology (pp. 135-146). Springer, Berlin, Heidelberg.
- [8] – Shaker, N., Togelius, J., & Nelson, M. J. (2016). *Procedural content generation in games*. Switzerland: Springer International Publishing.
- [9] – Albaghajati, A. M., & Ahmed, M. A. K. (2020). Video game automated testing approaches: An assessment framework. *IEEE Transactions on Games*.
- [10] – Van Kreveld, M., Löffler, M., & Mutser, P. (2015). Automated puzzle difficulty estimation. In 2015 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 415-422). IEEE.