

DEVELOPMENT OF A TOWER-DEFENSE GAME WITH PROCEDURALLY
GENERATED STAGES

A Thesis
Presented to the Faculty of the
Computer Studies Department
College of Science
Technological University of the Philippines
Ayala Boulevard, Manila

In Partial Fulfillment of the
Requirements for the Degree
Bachelor of Science in Computer Science

By:

KEITH C. ROSETE
DAN EIAIEL F. EBOÑA
CYRENE ROSE C. MANIBALE
MAURICE H. LAGAS

June 2024

INTRODUCTION

The study aims to develop a Tower Defense Game with procedurally generated stage generation with dynamic enemy AI pathfinding. The results of the study can serve as a basis for future research in the field of Procedurally Generated Games especially through the means of Tower Defense. The game will be created using the Unity game engine and the use of Visual Studio Code, 6.6 and C#. The Programming Language used in the development of the game is C# #utilized by Unity Game Engine. The game will only be developed for desktop computers and laptops. It will be an offline single player game that generates a new map whenever a new game is started through procedural content generation. The game will not focus on the shortest path possible. The core gameplay mechanic of a tower defense game is to protect the base or specified area from waves of enemies that progressively become more challenging in each round. In every playthrough, the player is only given a limited amount of HP. Enemies will also have various HP counts. If the player has at least 1 remaining HP when the final wave ends, the player will be shown a victory screen.

METHOD

The game's replay value will be increased by the development of a pathfinding algorithm that will provide a variety of paths with varying degrees of complexity and difficulty. The Planning phase begins by Identifying the Objectives and Defining Scopes. The Game Grid, Perlin Noise, and Spawn Point scripts are used to generate the procedural generation in the project's tower defense game. The main menu has the Mode, Settings, Gallery, and Exit buttons. The Index Use Case is shown in Figure 2. The Tower Defense game will be built using the Unity Game Engine. The game starts at the Screen which prompts the player to click to go to the Main Menu. The main menu includes the Main menu, Gallery, the Settings, and information about Slimes and Tower. There are several tower types built by the player which can be built to protect the base. The soundtracks are publicly available for commercial and free of use for commercial and non-commercial purposes. The player can also choose the resolution for the screen resolution. The game starts with an opening sequence that can be skipped by pressing any key. Maintainability testing is performed to guarantee that the game can be adjusted to meet requirements. The core tower defense game mechanics follow the conventional elements of players defending against waves of enemies by strategically placing defensive towers. The game system is determined through ISO/IEC 25010 quality metrics to evaluate the developed program. The player is able to build towers and there are enemies for the towers to target. The Gallery button allows the player to view the Enemy Type and Tower Types. The game's premise is shown in Figure 5 Penzilla's Basic GUI Bundle. The three pieces of music used in this project are obtained from the DOVA-SYNDROME Official Youtube Channel. The audio sound effects used in the development of Slimes Raid Cookie TD were uploaded by YouTube channels SoundX, Sandra Michelle, and 2MirrorsDialogue. The game can be set up to run on several different platforms, or it can be used to create additional objects or systems. 54. The Low-Level Design (LLD) for the implementation of the High-Level design (HLD) numerous components and phases on a specific level. 54. The High-level Design is the design of the game's various components. 54. Some of the components are created to actively target the player's bases

Low-level design types. 54+. The High Level Design is created to target the players? bases Low Level Design types.

RESULTS

There are two buttons the user can click on, the first on the left restarts the game starting a new playthrough in the same game mode while the other returns the screen to the Main Menu. The player can select the Play button in order to start the game, change the settings of the game through the Options button, check the gallery through the magnifying glass icon, or exit the game. Once the player clicks the Tower button, the player will be directed to the Tower Gallery Screen. Players are given a limited amount of resources in the form of in-game currency to strategize in defeating waves of enemies and surviving. The evaluation process utilized the ISO 25010 software quality metrics to evaluate the developed program to meet the standard criteria for assessing the quality of the software. The game may not work on Windows versions that are no longer being supported by Microsoft. The player can place and remove towers on the nodes within the grid nodes in order to defend the base from waves of monsters. The Tower Gallery Screen provides information about every Tower gameplay. Clicking on the Checkmark restarts the game while clicking the X Mark resumes the game. Slimes Raid Cookie TD is a 2D Pixel Art Tower Defense game that takes inspiration from old school tower defense games and modern procedurally generated games. The Normal game mode ends upon clearing 20 waves of enemies. The game offers varied gameplay challenges from procedurally generated stages. The AI of the monsters are able to explore and exploit multiple paths. The project is developed using C# and the Unity Game Engine of Test Cases. The functional suitability test execution is performed in Table 8. The final chapter presents the project description, project structure, project test results, and project evaluation results of the study.

DISCUSSION

Slimes Raid Cookie TD received an overall weighted mean of 3.13 that can be interpreted as a ?Very Acceptable? rating based on the criteria of ISO 25010 on Functional Suitability and Maintainability. This section of the paper presents the findings of the evaluation, including a comprehensive analysis of the scores obtained from each criterion. The report also includes the conclusions and recommendations for future improvements of the project. The study has achieved the following insights: The game effectively aligns with the requirements and expectations of its users. The implementation of Ant Colony Optimization on the pathfinding of enemy AI introduces a dynamic level of randomness.