

Payback - A Revenge Arcade Game

Adityakumar Yadav
Department of Information Technology
K.J Somaiya College of Engineering
Mumbai, India

Abhishek Doshi
Department of Information Technology
K.J Somaiya College of Engineering
Mumbai, India

Aditya Thadeshwar
Department of Information Technology
K.J Somaiya College of Engineering
Mumbai, India

Prof. Ravindra Divekar
Department of Information Technology
K.J Somaiya College of Engineering
Mumbai, India

Abstract— Today gaming is a sixty-billion-dollars-a-year industry which pushes the performance limits of today's computers. Games have been written since the days when DOS ruled the IBM PCs. And yet, old arcade games and 2D DOS games are still enjoyed by a vast number of game enthusiasts.

Our proposed game is a story based 2D arcade style top down shooter game with a retro DOS style, yet vibrant and immersive interface. The user will find that this game is very easy to learn; yet it gets trickier as we move from one level to another. It is not only action-packed; it also has a captivating storyline and background music that the end user will thoroughly enjoy.

Keywords—Game Development, 2D, top-down, shooter, Gamemaker Studio 2, Artificial Intelligence (AI), GML, GUI, mp-grids.

I. INTRODUCTION

Computer games have become very popular in children and adolescents' life. Everywhere in technology-rich environments equipped with laptops, smart phones, game consoles like mobile devices and other digital devices games are used. Game development technology has matured and became more advanced than before in recent years.

The proposed game will be a single-player 2D top down shooter game made in Gamemaker studio 2 using Gamemaker language (GML). Player will be able to move around in a given level and shoot enemies. Enemies will have basic AI of moving around in the level and shoot player. It will consist of multiple levels supporting an interactive storyline. It will be vibrant in looks with immersive story and metro look. We are trying to propose a game which have a retro-style look along with an adventurous gameplay using latest technology game engine i.e. Gamemaker Studio 2, so that it gets the player more involved in the game. The proposed game i.e. Payback is a story focused 2D top-down shooter game for PC's where the player will get the essence of arcade gaming style along with thrilling gameplay.

II. LITERATURE SURVEY

Christopher M. Kanode and Hisham M. Haddad [1] talks about unique methods of using Software Engineering methodologies to develop games by combining the work of teams covering multiple disciplines. This research paper is mainly focused on various scenarios and challenges in game development and

how they can be overcome by using traditional software engineering practices.

Rido Ramadan and Yani Widyani [2] addresses various challenges faced by the game developers during its lifecycle. This paper presents a new game development life cycle model and provides guidelines for successfully delivering a good quality game. The proposed GDLC takes on the iterative approach to enable a higher degree of flexibility towards changes during the development process.

Junfeng Qu, Yinglei Song and Yong Wei [3] analyzed various game design patterns like state patterns, observer pattern, mediator pattern, builder pattern, etc. used in game development. Since game development and programming involves rapid modification, so there is a need to make sure the correctness of a program by evaluating its behavior and which can be done using various game design patterns. Design Patterns can also be used to create sprites, different behavior for different sprites with strategy, state management, communication and collision detection.

III. ABOUT GAME ENGINE

The proposed game was made using Gamemaker Studio 2. The motive behind choosing Gamemaker Studio 2 was: 1] Multiplatform Workflow: Using a single development workflow it allows you to export your game directly to Windows Desktop, Mac OS X, Ubuntu, HTML, Android, PS4 and Xbox One. 2] Provides extensive library of events and actions to sculpt the game you desire. 3] GML being, based on C programming language makes it easy to learn. 4] It offers Chain View which helps to easily navigate between resources without losing track. 5] Gamemaker allows you to structure your game to work with a client/server model, dealing with all of the network management behind the scenes, while allowing you to focus on the game itself. 6] It helps to save and backup the project history from anywhere in the world with their Git Integration. 7] It has powerful debugger for tracking errors and bugs which eventually saves both time and efforts. So, the main aim was to extract some of these features in the proposed game.

```

if(mouse_check_button_pressed(mb_right))
{
    if(sprite_index != sprPWalkKnife)
    {
        sprite_index = sprPWalkKnife;
        mask_index = sprPlayerCollisionMask;
    }
    else
    {
        sprite_index = sprPWalkMP5;
        mask_index = sprPlayerCollisionMask;
    }
}

```

Sample code snippet to switch between knife and gun sprite.

IV. METHODOLOGY

The research method used for this project was exploratory in nature, as the motive was to try to explore the domain of game designing and development. This method was very much flexible and do not involve testing of any hypotheses. Every game has their own design and system architecture. The architecture chosen for this game was Data Flow Architecture. Since the gameplay was designed such that the player moves ahead only after killing all the enemies and obtaining sufficient points for the current level. So, the data flow follows a sequential pattern. Moreover, the data i.e. scores gets carried from the previous level to the next upcoming level so the architecture perfectly fits into the system. Literature survey conducted was fruitful as it helped to take into account several design patterns for the proposed game. For e.g. Prototype Pattern [4]: As the game renders large number of enemies at different locations having varying strengths and attacks, this pattern provides solution to create an abstract Enemy Class and then different subclasses can be built based on the requirements of power and capabilities.

V. GAME DESIGN AND IMPLEMENTATION

Payback is a story focused 2D top-down shooter game for Windows. The main aim is to provide the user the essence of arcade gaming style along with thrilling gameplay. The proposed game is like mini-RPG (Role Playing Game) where the main user will be playing a role of Cop and acting according to the desired set of events happening around him. The game is divided into three chapters with increasing level of difficulties and uncertainties. After completion of every chapter or level the player will be provided with hidden message/hint about the next location. The mission of this game is to rescue the daughter from the hands of enemies without getting killed and try to find out the purpose behind his revenge. In order to progress in the game the player has to kill all the enemies present in the hideouts or godowns and try to collect the hidden message so that he can know the next secret location and advance through the levels ultimately reaching the main boss location where he finds his kidnapped daughter and rescues her by killing the main boss.

VI. LEVEL DESIGNING

Level Designing is both an artistic and technical process and it was done with the help of Room Editor provided by

Gamemaker 2. The proposed game 'Payback' is divided into three chapters wherein each chapter consists of an average of five rooms. Each room is designed and decorated according to the required aspect and storyline of the game. The Chain-View feature of Gamemaker 2 comes in picture here, as it makes easy to identify which room is connected where. Each room is occupied by the desired objects, sprites and pre-designed events that will happen on interaction with the objects. Since the game is more of an action-drama it involves the use of Artificial Intelligence to have more realistic approach. The AI of enemies vary slightly, causing them to occasionally move unpredictably and making it difficult for the player to plan a perfect approach. To make up for this, Payback allows the player to restart each stage as soon as they die, allowing them to quickly fine-tune their approach over several attempts. The game is designed such that only after finishing all the enemies of a room the player is allowed to make transition to another room. Appropriate textures and audios are being used to produce an immersive player experience at each level.

VII. USING AI FOR PATH FINDING

Enemies will use Artificial Intelligence in path-finding to detect and shoot the player accordingly. This can be achieved in Gamemaker Studio 2 using mp-grids. An mp-grid is a "motion planning grid", and all it does is section up a room into individual grid "squares", and each of these squares can then be flagged as "occupied" or not. This grid is then used by another mp function to create a unique path resource that will go around those squares flagged as "occupied" and go through those ones that are not. you would then assign this path to an instance and (to the player) it will look like the instance is displaying "intelligence" as it neatly avoids obstacles while following the path. Another important function that is often used in decision making and AI, is angle_difference. This function determines the smallest relative angle between two directions, a value between -180 and 180 degrees. For example, by measuring the angle between the direction where an enemy is facing and the direction of the target object, one can determine if the target is within the "cone of vision" of the enemy or not and if the object lies within the cone of vision of the enemy the enemy will try to attack the player object.

Cone of vision:

TABLE I. EXAMPLE OF CONE OF VISION

cone_left/ cone_right	image_angle	Cone of Vision
75 degrees	50	25 to 125

```

cone_left_angle = image_angle + cone_left;
if(cone_left > 359) cone_left -= 359;

```

```

cone_right_angle = image_angle + cone_right;
if(cone_right < 0) cone_right += 359;

```

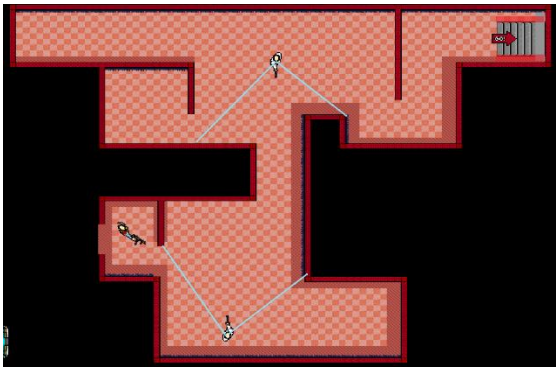


Fig. 1. Example of Cone of Vision used by enemies.

VIII. GRAPHICAL USER INTERFACE

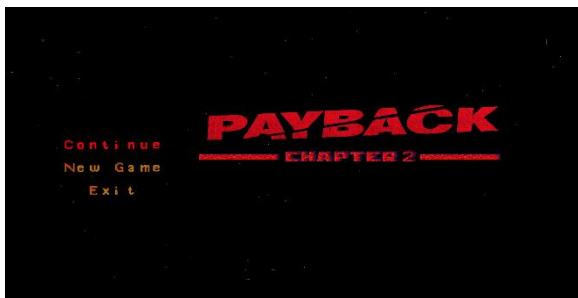


Fig. 2. Game's Menu Screen



Fig. 3. Example of Game Cutscene

Character	Top view	Role
		Protagonist Name: Leo Role: Cop, Kate's Father Abilities: Shoot, switch weapons, collect ammo, medkits and knife.
		Name: Kate Role: Leo's Daughter
		Name: Alex Role: Leo's Friend
		Antagonist Name: Carl Role: Gang Leader, Kidnapper Abilities: Shoot
Enemies		

Fig. 4. Characters involved in the game

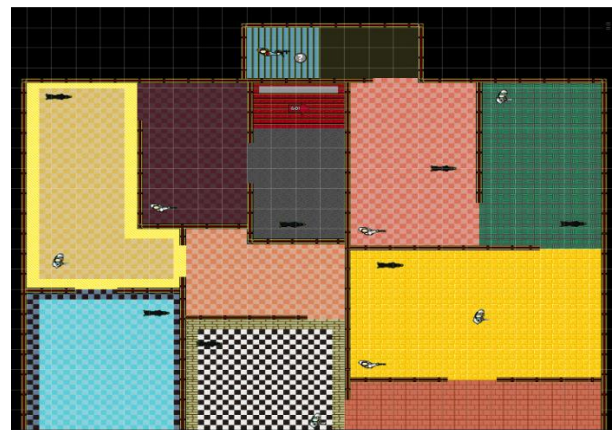


Fig. 5. Different Room Level Designs.

IX. CONCLUSION AND FUTURE ENHANCEMENTS

The proposed game has attempted to provide the essence of arcade gaming style along with thrilling gameplay to the end users. GameMaker 2 played a crucial role in the development and deployment of this game. Future Enhancements can include a couple of different stories and angles. Also, at the end of each chapter, the game can grade the player's performance based on factors such a speed, variation and damage, with high scores which will help to unlock new weapons and additional masks to change player's avatar. More attacks such as punches, weapon throws and door slams can added to increase the difficulty level of the game.

REFERENCES

- [1] Christopher M. Kanode and Hisham M. Haddad, "Software Engineering Challenges in Game Development" in Sixth International Conference on Information Technology: New Generations, 2009, Las Vegas, NV, USA.
- [2] Rido Ramadan and Yani Widayani, "Game development life cycle guidelines" in International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013, Bali, Indonesia.
- [3] Junfeng Qu, Yinglei Song and Yong Wei, "Design patterns applied for game design patterns" in 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing , SNPD 2016, Shanghai, China.
- [4] B. Kreimeier, "The Case For Game Design Patterns", 2002, [online] http://www.gamasutra.com/view/feature/132649/the_case_for_game_design_patterns.php.