International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

# 2D PLATFORMER GAME IN UNITY ENGINE

**Tejas Bhosale[1], Sahil Kulkarni[2], Shreya N. Patankar[3]**

*[1,2] Students, Computer Department, Mumbai University, Datta Meghe College of Engineering, Sector-3, Airoli, Navi Mumbai*
*[3] Assistant professor, Computer Department, Mumbai University, Datta Meghe College of Engineering, Sector-3, Airoli, Navi Mumbai*

---

**ABSTRACT -** *A platform game is a subgenre of video game and can be of two types such as a puzzle game or an action game. The main concept of a 2D platformer is that the player will control a character or avatar and guide him through a series of platforms which can be grounded or elevated depending on the level design. The mission of such games is always to collect certain items (coins, rubies etc.) and avoid or fight the enemy characters. In this game we are creating a beautiful cloud city environment which will consists of grounded as well as air suspended platforms and a system to collect coins and multiple potions which will change the character or avatars behavior. Through the development of this game we are trying to demonstrate the combination of creative thinking and problem solving by programming an exciting experience, for this we are going to use the plethora of functions and features provided by the Unity3D engine and some image and audio editing software. This game will be supported on multiple platforms such as the latest versions of Windows, MacOS and Linux.*

## 1. INTRODUCTION

Game development is the process of using graphics libraries, game engines and image editing software to display images on the screen and let the player manipulate the images on the screen to create experiences and tell the player a story.

There are two types of 2D games: top-down camera approach and 2D front of the scene camera approach. In top-down camera approach the camera is placed in the ceiling position so that the player is viewing the images from the top so the manipulations are done in all four directions i.e. north, south, east, west. The player has full view of the whole level. the. In 2D front of the scene camera approach the camera is placed in front of the scene. A scene is the placement of all game attributes to create a scenario. In this approach the player only sees that much part of the screen which can be captured by the camera at any given time. The concept of this type of camera placement is that the level is gradually introduced to the player by moving the camera on the X axis in both the directions. Furthermore we will be using procedural generation to create variety in level design and loot placement. Loot is the coins and potions that will be used to give the player an incentive to explore as much of the environment as possible. In this game the player can control the avatar using traditional input devices like keyboard or a dedicated game controller.

### 1.1 Features of the application:

**1.Interactive UI:** The user should see the home screen, which will have an option for starting a new game, continuing your previous game, and a settings tab. The settings tab will let the user adjust the display and sound settings.

**2.Freedom of Control:** The user will get the option to choose the input device of their preference for e.g. the player can choose a keyboard layout setup or a Controller setup.

**3.Exploration and item collection**: Interactive levels with multiple ways to reach the destination, explorative minds will be rewarded with treasures and health potions for exploring every nook and corner.

**4.Procedural Generation:** The player can find secret levels which are completely procedurally generated, I.e. the levels are made iteratively based on increasing difficult.

## 2. LITERATURE SURVEY

2.1 “Effective Triangulation based Pathfinding.”, Douglas Jon Demyen

In this paper, author describes a technique for extracting features such as dead ends, corridors, and decision points from an environment represented using a constrained Delaunay triangulation. They make use of triangulation to represent the environment. The approach taken in this thesis uses the properties of a constrained triangulation (one formed around the aforementioned line segment barriers) to identify dead ends, corridors, and decision points in a connected area of the environment. The result of this process is a graph where the nodes represent where a decision must be made as to which side of an obstacle to go. The final result is that the pathfinding task is simplified to the point where the search algorithm need only decide to which side of each obstacle to go, while all features of the environment are retained.

2.2 “The Guerrilla Guide to Game Code”, Jorrit Rouwé

There are a lot of articles about particular aspects of a game. However very little is written on the subject such as the binding structure that ties all aspects of the game together.

choosing a structure to build your game on is important. The paper suggest use of a Model-View-Controller mechanism for development purpose. The Model-View-Controller mechanism fits nicely with other often used techniques like making a deterministic game. Model-View-Controller architecture is also really good at is online multiplayer. A good structure will reduce risk and increase the efficiency of the team.

2.3 "Model Driven Game Development: 2D Platform Game Prototyping", Montero Reyno and José Á. Carsí Cubel.

To deal with increasing complexity of game development, this paper proposes to apply Model-Driven Development (MDD) methodology to game development in order to rise the level of abstraction of game development, enhancing productivity and shifting efforts from game programming to game conceptual specification, allowing the automatization of the coding tasks in game development. As an example, they present a game design prototype tool to prototype 2D platform games for PC, with automatic code generation from UML models.

2.4 "The art and science of level design", Cliff Blezinsky

This article deals with defining the role of the "Level Designer" and how his role is crucial behind the success of the development process as well as the commercial success of the game. A Level Designer should have knowledge about Art as well as the science behind game development so that a balance can be maintained between the visuals and the functionality of the game. The limitations of the hardware and the platform on which the game is built should also be considered. These responsibilities are explored and explained in this paper.

| Sr No | Paper Title | Authors | Advantages | Disadvantages |
|---|---|---|---|---|
| 1 | Effective Triangulation based Pathfinding | Douglas Jon Demyen | It simplifies the pathfinding task to the point where the search algorithm need only decide to which side of each obstacle to go. | The concept is useful but a little difficult to implement |
| 2 | The Guerrilla Guide to Game Code | Jorrit Rouwé | It suggest Model-View-Controller mechanism for development purpose which reduces risk and increase overall efficiency. | The main drawback is that there will be some code and memory overhead to support an Entity and Entity Representation class for every game object. |
| 3 | Model Driven Game Development: 2D Platform Game Prototyping | Montero Reyno and José Á. Carsí Cubel | The model-driven approach enhances productivity in game development and reusability of the software artifacts | The UML models used to specify 2D platform games are closer to software engineers than to game developers |
| 4 | The art and science of level design | Cliff Blezinsky | It makes us understand who level designer is, his/her responsibilities and role in development process | Role of level designer keeps changing and one has to adapt to it. |

**Following applications have been implemented:**

**1. Super meat boy:** Super meat boy is an independent video game designed by Edmund McMillen and Tommy Refenes and was developed by Team Meat. This game comes under the category of puzzle platformer, the aim of the game is to go through multiple platformer levels with increasing difficulty.

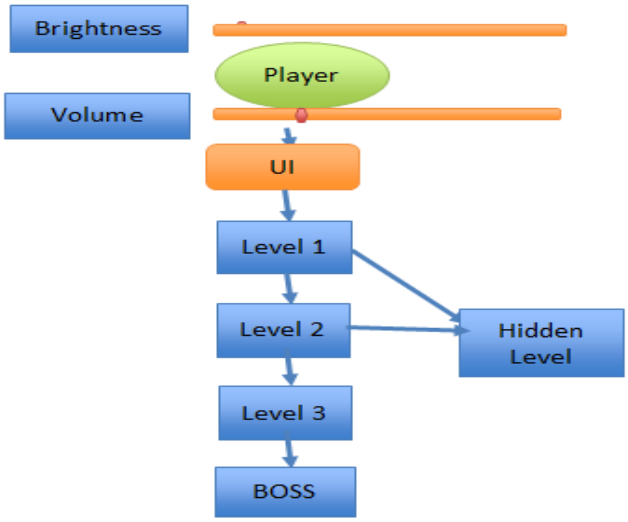
**2. FEZ:** Fez is an independent puzzle platformer game developed by Polytron Corporation. The game revolves around the story of a main character and it is unveiled through part by part as the game progresses.

**3. Spelunky**: Spelunky is an open source indie platformer video game created by Derek Yu. This game uses procedurally generated levels for platform designs and enemy and collectibles position. This game shows how procedurally generation helps in creating variety to give different experience to different players.

## 3. IMPLEMENTATION

The user interfaces are planned as follows:

### 1.Main Screen:

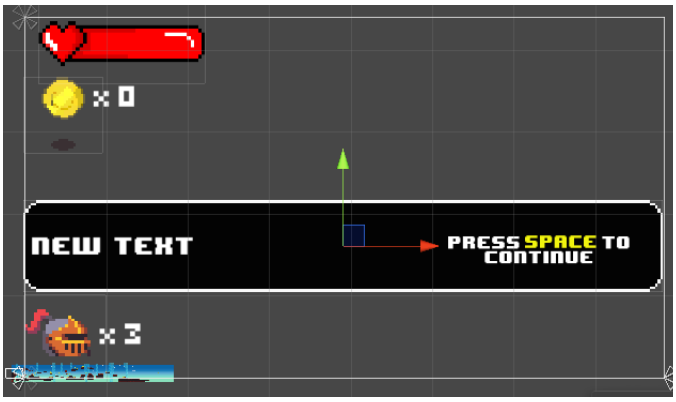



### 2.Settings Screen:

This Screen will have attributes and sliders to adjust the values.


### 3.In Game UI Screen:

The In-Game UI will show the text-based dialogue box, the game attributes such as health bar, number of lives, and coins collected.




### 4.Custom Brush:

We programmed a drawing brush to draw better level design. Before this the way of designing a level was to drag and drop individual sprites, the custom brush works on the 2D grid system provided in Unity. We added a top, bottom, right and left component of a sprite, so when we draw with that brush it automatically completes the image by adding the remaining components of the sprite. This reduces the time taken to design and create a level by a lot.

### 5.Trigger Animation:

Based on the concept of Wumpus world example we have devised a way to trigger the enemy animations only when the player is in the vicinity of the enemy. This is done by using a On Trigger Enter box. On Trigger Enter is a function provided by Unity Engine which can trigger certain actions when the player enters this box. This way whenever the player enters the box the enemy simulates the actions of recognizing the player.

### 6. Hidden Procedural Cave Generation

It makes sense of game in terms of tile system. It will leave it boundary tiles open so that the level is passable. Then for the remaining tiles, it maintains list of all possible vector positions. Now it will randomly select a position from the available position list and spawn an element there ( it can be anything like coins, walls, enemy, etc) The swapned location is then removed from the available vectors list. The process is repeated for all the elements that need to be spawned and this entire process is repeated to generate new cave with difficulty based on level number.

## 4.TECHNOLOGY USED

**Unity3D Engine**: This is the main component based on which the whole project is based. It is the ultimate game development platform. The all-in-one editor offers the latest and best tools for developing, launching 2D games. The unity3D engine also provides SDK's and development environments for Window's, MacOS and Linux systems.

**Sprite Workflows**: Unity Engine's sprite workflow editor and features were used to scale and slice the texture sheets to get every individual frame to be in the perfect scale.

**Animation and Animator**: Sliced Images are grouped together to add animation to the avatar the frames per second feature allowed us to get the perfect cycle range for our animations.

**Object Oriented Programming in C#**: The C# language was used to make all the components work together and communicate with each other. The C# language allows to attach individual behavioral scripts to each game component, this way information can be shared between each component and the UI to make the game as interactive and feature full as possible.

**Photoshop:** Photoshop is the predominant photo editing and image manipulation software, this application was used to create pixel art images which forms the main art assets of our game.

**Audacity:** Audacity is an open source audio editing application that was used to make sound effects and add a depth to the game.

## 5. CONCLUSION AND FUTURE SCOPE:

This game will take the player on a journey of adventure and exploration through multiple levels and scenarios. This system can be expanded in the feature by adding more levels and enemy types, develop better AI for the enemy and NPC's (non-playable characters). Future versions of Unity3D engine will bring more features and opportunities to increase the efficiency of the code and add more features.

## 6. REFERENCES

[1] "The Guerrilla Guide to Game Code", Jorrit Rouwé

[2] "Model Driven Game Development: 2D Platform Game Prototyping", Montero Reyno and José Á.. Carsí Cubel

[3] "The art and science of level design", Cliff Blezinsky

[4] "Efficient Triangulation-Based Pathfinding" Douglas Jon Demyen