International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 9, ISSN No. 2455-2143, Pages 124-128
Published Online January 2020 in IJEAST (http://www.ijeast.com)

OFFLINE TIME LIMIT PACKAGE AND GAME DEVELOPMENT

M.S.Vinu,
Assistant Professor, Department of CSE,
VSBCETC, Coimbatore, India

N. Santhosh, S.Sudhan Raj, M.Sriram, M.Siva Senthur
UG Scholar, Department of CSE,
VSBCETC, Coimbatore, India

***Abstract*—The creation of games requires a lot of knowledge and experience. Nowadays there exist a lot of packages that allows developers to create the games in faster and easiest way. The aim of this project is to develop a 2D game for android using C# in unity game engine. The final product is a challenging collaborative game, which contains 10 levels. The main proposal of this project is to implement a package that limits the game playing time in offline games.**

Keywords— Game Development, 2D Game, Time Limit Package, Application Development

I. INTRODUCTION

Nowadays, game is one of the entertainment media that has become a trend. Currently, game is not only used as a medium of entertainment, even some are made for the purpose of learning media with high effectiveness. Two-dimensional games were most frequently developed in the early years of video games with the main reason for this being that the technical limitations of game hardware prevented the ease of creating three-dimensional graphics. When technology developed sufficiently to allow easier and more effective use of 3D graphics there was a temporary decline in 2D gaming.

Many researchers have discussed the possible benefits for using games in education and we believe they may be particularly helpful in improving computer science education. Games provide a way to create and share educational content while also making students feel their computing education is more relevant. The purpose of these notes is to present and build in a logical progression the skills required to build a 2D game and specifically a platform game in C#.

II. GAME OVERVIEW

This is a story based action game were player character is an apple. Player has to kill all enemies and to cross the challenging levels to save his friend. In between the levels there will be many obstacles, message pop up characters and many environmental objects

III. LEVELS IN GAME DEVELOPMENT

- Planning (type of game)
- Designing required for 2D objects
- Designing the structure of levels in game in rough work
- Importing objects
- Animating the character
- Placing the Object
- Designing the interface
- Programming character control
- Programming scene management
- Programming each object
- Build to android

IV. 2D OBJECT DESIGN

2D objects are designed in Photoshop. Each object is designed based upon the layer. Some of the 2D objects are player, enemies, weapons, environmental items, etc. Objects that are designed must be exported in .png or LDF format based upon its need.

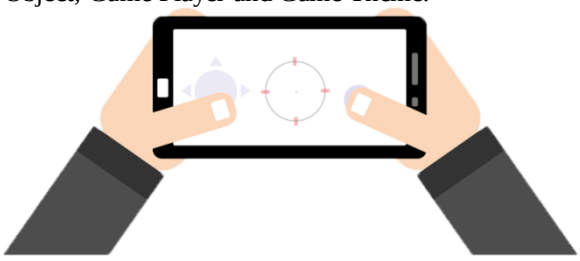
V. ANIMATION

Animation can be done in three ways, Using Multiple Sprites Using Key Frames Attaching bone to the objects. Weapons are interchanged using multiple sprites. Players, enemies, and other 2d environmental objects are animated using key frames with the help of bones.

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 9, ISSN No. 2455-2143, Pages 124-128
Published Online January 2020 in IJEAST (http://www.ijeast.com)

VI. GAME CONTROL MODEL

The game control model presented by authors was developed based on their experience and studies and represents a game ontology from an interactive content viewpoint. They mainly were focused on role-playing and simulation game genres, but the design can also be used in game genres such as action, adventure, sports, vehicle, strategy, etc. The game control model consists from ten interrelated key concepts that best represent the information about game: Game Structure, Game Presentation, Game Simulation, Game Rules, Game Scenario, Game Event, Game Objective, Game Object, Game Player and Game Theme.

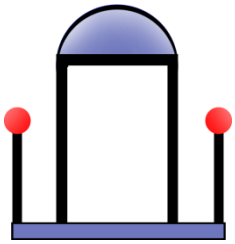


VII. GAME RULE

A game principle states the relation among game world and game objects, also the interaction effect. In the considered model, game principle can be a Game Scoring Rule or Game Interaction Rule. A game scoring rule is merely applicable to game player and it determines what to be awarded to the game player when a scoring condition is satisfied. Game interaction rule dictates the outcome of the interaction from two game objects. Each game interaction rule has an actor which can be a game object, a class of game object or a group of game objects from different classes, a subject which represents a game object, a class of game object, a group of game objects from different classes or the game world and an interaction condition which refer to the state of actor or the state of the game world.

VIII. GAME OBJECTS

Game objects are virtual things that settle the game world and can be developed to have a set of capabilities, for example, making a decision, moving, reacting to environment and player‘s input simulating their presence at the game. An object holds as a set of object attributes, an appearance, intelligence, it can carry out actions and have a description which characterizes the esthetic image of it. Game objects can be classified into 2 groups depending on their behaviors: interactive and non-interactive game objects.



IX. GAME PLAYER

Player is the game user that provides inputs to the game system. Game player is introduced like an entity that has an avatar, an inventory, game attributes, control and records. Game player is introduced by an avatar that is a game object. Avatar has straightforward relations to the game control like input events of game control interface are mapped onto the avatar actions Existence and accomplishment of a player are introduced by data that are game attributes. These data represented as a vital (number of chances to play the game) or a score. Game control gives player the channel to control game objects. Player‘s input is taken through game control interface that can be a hardware interface (touch control, keyboard, mouse, gamepad, motionsensor, camera) or GUI.

X. GAME THEME

Game theme depicts majority of the art requirements connected to the game via significant written text. Visual, aural and even language of the narrative are involved. These are media content on its own that are created by qualified people and correspond to the format of the game technology. This information is determined in the objects and emerge as reference to the file.

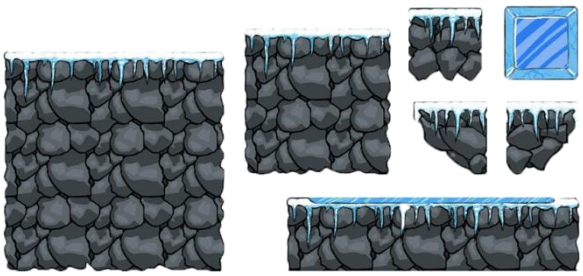
XI. ACHIEVED WORK

12.1 RENDERING VIRTUAL WORLD *(TILE MAP)*

Map of a whole level, or the game map, is typically a few screens wide in a 2D platform game.

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 9, ISSN No. 2455-2143, Pages 124-128
Published Online January 2020 in IJEAST (http://www.ijeast.com)

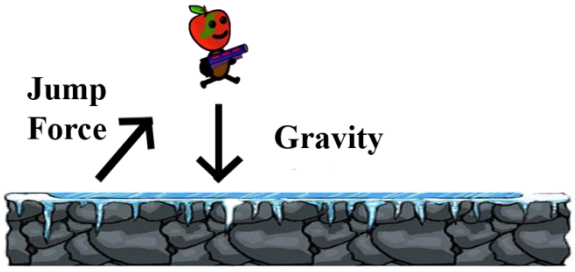
Some maps are 20 screens wide; however others can be 100 or more. Map scrolls when the player walks across the screen. Having one big image for the game map it is not the best solution, it leads to consuming a lot of memory, so in some cases machines would not be able of loading the map. As an alternative of using a big image for the whole map, you can make a tile-based map. Tile based maps divide the map into a grid. Every cell in the grid keeps a little tile image or nothing. The tile map keeps links to which image belongs to each cell in the grid. Thereby, you should have merely several little images for the tiles, and you are able to create the maps as large as you want without concerning too much about memory limitations. The file, which contain the available images for drawing the tile map, present as a sprite sheet. Sprites for Tile Map It consists from two rows: - the tiles in the first row are normal (the player can go through this tile); - the tiles in the second row are blocked. The tile size can be different and it depends of sprites which are used in the game. However it this game the tile size is 30x30 pixels. Every tile from the sprite sheet is stored in tile array accordingly. Tile-based games constantly have greater than one map or level. The small representation of tile map file presented. You will require a simple method to make multiple maps so that just as the player completes one level, the player can afterwards start the next map. Tile Map Only small part of the map is displayed on the screen at a time because tile map is much bigger than the screen. To keep gaming process, the game view scrolls to hold the player in the center of the screen, when he makes any movements. Based on tile map size and game panel size we can specify how many tiles in row and columns we have to draw. However, before you draw the tiles, you need to figure out the position of the map on screen. Start off by keeping the player in the middle of the screen. These parameters specified in constructor of Tile Map.



12.2 PHYSICS ENGINE

Substantial thing that physics engine does:

- Simulate movement (resistive forces such as gravity, also applied forces such as jumping, running, friction).
- Detect and resolve collisions among the player and other objects in the level.
- In physics engine, player will possess the movement-describing variables: moveSpeed, maxSpeed, fallSpeed, maxFallSpeed, jumpStart, stoptJumpSpeed.



- With help of these variables, each movement that you apply to the Player will follow next steps:
  1. Is the jump or move action chosen?
  2. If yes, apply one of the above actions to the player.
  3. Gravity should be applied to the player too.
  4. Determine the resulting speed for the player.
  5. Determine the jump force
  6. Apply the resulting speed to the player and update his location.
  7. Collision among the player and other objects should be checked.
  8. If a collision happens, it should be resolved by moving the player pretty back so that the collision is no longer taking place, or by causing damage to the poor player.

In the game, gravity is always pushing the player down and across the ground, but the collision resolution step will place him back on top of the ground in every frame.

Collision resolution step To determine collision in the game we may consider a rectangle that is defined by 4 values, a left, a right, a top and a bottom.

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 9, ISSN No. 2455-2143, Pages 124-128
Published Online January 2020 in IJEAST (http://www.ijeast.com)

In majority 2D game coordinates left and right are 'x' values, left value lower than the right; top and bottom are 'y' values, top value numerically lower than the bottom[5]. To find collision among two rectangles we may examine if the two rectangles 'x' values (left and right) overlay. Moreover, it can be checked if the rectangles 'y' values (top and bottom) overlay. In case when there is an overlay among both 'x' and 'y' values the two rectangles are overlaying, else they are not.

When we are considering 2D game, a tilemap is usually a grid of even-sized rectangular (mostly square) tiles. Every tile may keep data regarding it's appearance, as well as an indication of if the tile is a wall, open area, or something else. Given we have a way of finding collision among 2 rectangles, finding collision among an object and a tilemap can be as elementary as matching a rectangle that is represent the object and every single wall tile rectangle.

12.3 OFFLINE TIMW LIMIT PACKAGE

To set time limit for offline game, two clocks are built inside the game as a package So that user can access the game only for particular time interval that is set by the developer.

One clock will run in the background of the game another timer will run inside the game. This clock and timer will not visible to the user so that user cant access the clock and timer.

Clock will be set for 24 hour clock format, Once the clock reach 24 hour it will restart timer.Time limit will be set in the timer.

For example if the developer set time limit 3 hours then user can access only for 3 hour in a 24 hour, then user can play the game only after the timer restarts.

XII. FUTURE WORK

To continue this game we need add: - bad players; - create a super power of player (like fireball attack, etc.); - number of player lives; - more obstacles in the level – audio

XIII. CONCLUSION

In this paper, I have presented the idea of how to design and develop 2D game and the idea of offline time limit to set in offline mobile games.

XIV. REFERENCES

1. B Mobasher, Cooley R Jaideep, "A Virtual Exhibition Scheme Based on 3D Game Engine [C]" in , New York:IEEE Press, 1999.
2. C Shahabi, A M Zarkesh, J Adibi et al., The Fight Game Based on Actual Person Countenance, 2001
3. T Yan, M Jacobesn, Lina H Garcia-Mo, "Analysis and Design of 3D Game Engine Based on Design Patterns [C]", Proceedings of the 5th Int'l World Wide Web Conf., 1999.
4. L. Xu, "Research on Mobile Phone Online Game Development Engine [c]" in Hash table,Alaska:Anchorage Press, 2001.
5. J C Bezdek, Design and Realization of 3D Graphic Rendering Engine., Ithaca, NewYork:Cornell University, 1973.
6. M Kamel, S Z Selim, JOURNAL OF LUOHE VOCATIONAL TECHNOLOGY COLLEGE, vol. 27, no. 3, pp.421-428, 2001.
7. L A Zadeh, Design and Implementation of Multithread Simulated 3D Shooting Game, vol. 8, pp. 338-353, 2004.
8. M Kamel, S Z Selim, Texture Mapping and Its Application in 3D Game Engine., vol. 61,pp.177-188, 2006.
9. K S A 1- Sultan, S Z Selim, The Research of Our Country Forestry Enterprise Management Innovation under the Context of E-business, vol. 26, no. 9, pp. 1357-1361, 1993.
10. S Miyamoto, K. Nakayama, Design & Development of an Advanced 3D CAD System

    Based-ACIS Cybernet, vol.16,no. 3, pp. 479-482,1986.
11. J C Bezdek, R Hathaway, "SabinM and TuckerW", Based on game engine 3D graph system research, vol. 17, no. 5, pp. 873-877, 2003.
12. J. Greensmith, U. Aickelin, "Texture Mapping and Its Application in 3D Game Engine", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 49-56.

13. U Aickelin, S Cayzer, "T An Occlusion Culling Algorithm for Moving Objects in 3D Game Scene", 1st InternationalConference on AIS, pp. 141-148, 2002.

14. P Matzinger,"Texture Mapping and Its Application in 3D Game Engine", Annual Reviews in Immunology, no.12, pp. 991-1045, 2009.

15. J. Greensmith, U. Aickelin,S. Cayzer, "Research on Developing 3D Games Based on Morfit 3DEngine", ProceedingsICARIS-2005.