

Creating a 2D Platform Game

Sergii Shepelenko
University of Tartu
sergii@ut.ee

Abstract

The creation of computer games require a lot of knowledge and experience. Nowadays exists a lot of frameworks that allows developers to create the games in faster and easiest way. This work presents the approach of creating screen scrolling 2D game without using existing platforms (box2d, unity2d). The basic idea how to create game level with Tile Map and how to implement physics like jumping, free falling and running. Make a scroll movement based on player position on the screen.

1. 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 Java.

2. RELATED WORK

This work based on previous research by Stephen Tang and Martin Hanneghan from Liverpool John Moores University.

2.1 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.*

The interaction between these concepts illustrated on figure 1.

2.1.1 Game Structure

The game structure describes the architecture of the game. It divides a game into sections called *game sections*. Each game section consists of a game context, a pedagogic event indicator and an event trigger.

The transition from one game section to another game section (or the flow of game content) is governed by the definition of a trigger or the invocation of a GUI input event

The environment of the game depicts the style of the game content showed to players. It could be presented as simulation or presentation of the game.

Games that is developed for applying in education or serious games have to hold some

pedagogy methods. In the author's model which article I am reviewing, authors connect every game session with at least one pedagogic events from the Gagne's nine events of instructions [2]. These events ordered in sequence are (1) gaining attention, (2) informing learning objectives, (3) recalling prior learning, (4) presenting learning content, (5) providing learning guidance, (6) eliciting performance, (7) provide feedback, (8) assess performance and, (9) enhance retention and transfer[1].

To call the transition among the game sessions and the game events within the game scenario we need to use event triggers. The latter can be categorized as:

- user input is detected by input trigger through hardware interface or graphical user interface (GUI);
- time trigger is substantially a countdown timer that has time interval with frequency of once, to repeat infinitely or finitely;
- proximity trigger has a hotspot, which finds collision of a certain game object, a class of game object or a group of game objects among other classes;
- game mechanics trigger is related to a sequence of game application linked of events, for instance, game event, media event and simulation event.

Every event trigger is related to a command, which shows the transition among game sections or game event activation.

2.1.2 Game Presentation

Presentation of the game is a virtual canvas, which keeps media elements as well as GUI elements to make menu of the game, a game notification or cut-scene to introduce information regarding the game and let them to navigate across the structure of the game.

A classic game menu presentation consists of a virtual canvas, which contains a background image or animation, GUI element that refers to other game sections and maybe a sound loop.

2.1.3 Game Simulation

Game simulation is an instrument which reconstructs scenarios virtually for game-play to

occur. The simulation of a game scenario is defined by complex of principles that determine the interactivity, physical and temporal properties of the virtual world. Simulation of the game has game, game dimension, game tempo and game physics that provide it a form.

Dimension of the game relates to the virtual space where simulation of the game occurs. Therefore, it can be 2D or 3D. 2D restricts the viewing of a game scene from the side view (platformer or side scroller), top view or an angled view, while in 3D game you can observe game scene from all angles.

Game tempo defines measurement of the pace of time.

The physicality of the game is determined by the game physics. It embodies the collision world and environment forces. The collision world will inform the corresponding game object and simulation if a collision has been found and response to that event could be an action from game object or to award a score based on the determined interaction principle. In addition, the type of environment forces act on all game objects should be specified in the game physics.

2.1.4 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 [1].

view. Camera can have fixed position or it can be animated along a path to provide a cinematic display.

2.1.6 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.

2.1.7 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 (keyboard, mouse, gamepad, motionsensor, camera) or GUI.

2.1.8 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.

2.2 Conclusion

In the reviewed article, authors had analyzed the available game design models and emphasized the challenges regarding it – partial representation of a game and the lack of formalism. Obviously, that a new game design model is required specifically for the use in author's game model-driven serious games development framework [3]. Authors of article that is considered as related work for my seminar task have borrowed ideas from available game design model and elaborate a totally new model – the Game Content Model (GCM).

3. ACHIEVED WORK

3.1 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. 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 (see Figure 2).



Fig.2 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 in 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 at Figure 3.

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.

```

22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 23 24 25 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 23 24 25 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 23 24 25 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 21 21 21 4 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21

```

Fig3. 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.

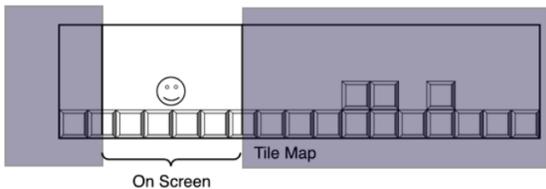


Fig.4

Based on tile map size and gamepanel 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:

```

public TileMap(int tileSize) {
    numRowsToDraw = GamePanel.HEIGHT /
    tileSize + 2;
    numColumnsToDraw = GamePanel.WIDTH
    / tileSize + 2;
}

```

3.2 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.

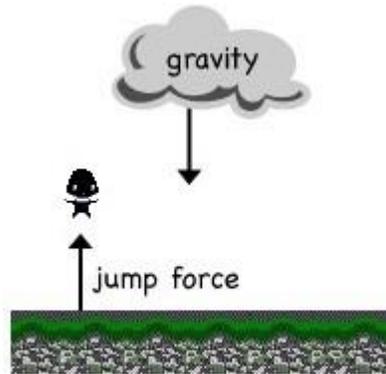


Fig.5 Player physics

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 (see Figure 6).

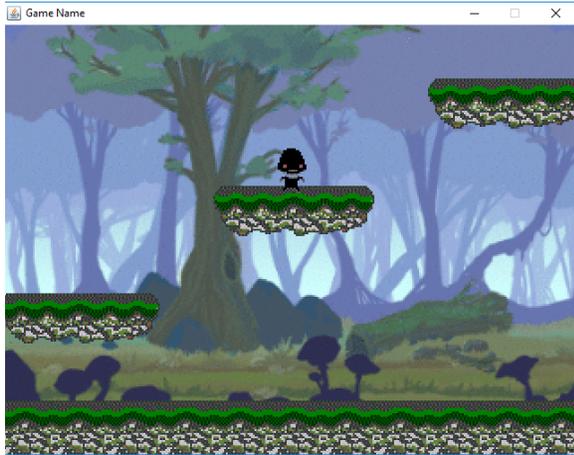


Fig.6 Collision resolution step

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. 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].

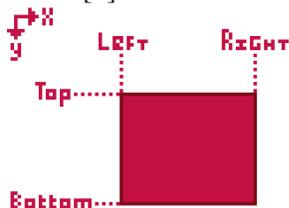


Fig.7

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.

4. 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

5. CONCLUSION

In this paper, I have presented the idea of how to design 2D scroll game. There is an implementation of working physics engine and creation a game world base on tile map approach.

6. REFERENCES

1. Stephen Tang and Martin Hanneghan "Game Content Model: An Ontology for Documenting Serious Game Design"
2. R. M. Gagne, The Conditions of Learning and Theory of Instruction, 2nd ed., New York: Holt, Rinehart & Winston.
3. S. Tang, and M. Hanneghan, —A Model-Driven Framework to Support Development of Serious Games for Game-based Learning, I in 3rd International Conference on Developments in e-Systems Engineering (DESE2010), London, UK, 2010, pp. 95-100.
4. David Brackeen, Bret Barker - Developing Games in Java 1st edition.
5. "2D Tilemap Collision" - <http://jonathanwhiting.com/tutorial/collision/>

