Bounding_Box.pdf

by

Journal of Physics: Conference Series

A Bounding Box Collision Detection Method Based On The Android System For Adventure Game

A Bounding Box Collision Detection Method Based On The Android System For Adventure Game

Wiwien Hadikurniawati[1], Edy Winarno[1], Taufiq Dwi Cahyono[2] and Dahlan Abdullah[3]

[1]Faculty of Information Technology, Universitas Stikubank, Semarang, Indonesia
[2]Department of Electrical Engineering, Faculty of Engineering, Universitas Semarang, Semarang, Indonesia.
[3]Department of Informatics,Universitas Malikussaleh, Aceh Utara, Indonesia

*wiwien@edu.unisbank.ac.id

Abstract. The purpose of this research is to create, design and develop “Star Ship Adventure” game using a bounding box collision detection method. This game is expected to be entertaining and fun to play in leisure time. It will be able to be built on Android device. The application of collision detection in games is used to detect collisions between objects in the game. This game is built using Android SDK (Software Development Kit), Android Development Tools, and Eclipse IDE using the Java programming language. Research results have shown that the player’s plane can read and detect collisions with enemy’s plane objects. The bounding box collision detection technique occurs when a player's plane is able to estimate the distance of an enemy object. When the player moves in the enemy's bounding box range, the detection occurs. As a result, the plane does not penetrate against the enemy.

1. Introduction

Technology development has become rapidly rising and supporting everyday needs. Despite their entertainment focus, Computer Games are at the forefront of computer science.Game technology is growing very rapidly and much studied by researchers because it has the potential of science and commercial. Game is a means of interaction between humans and computers that are assembled in a certain rules to determine the winner [1]. In making the game, the rules need certain algorithms to make up artificial intelligence in the game. The existence of AI (Artificial Intelligence) is very important, because without it the Game becomes very boring and not interesting to play. AI (Artificial Intelligence) is usually awakened from a certain algorithm.

Mobile software development has evolved over time. Google released Android which is an open-source mobile phone operating system with Linux-based platform. It consists of the operating system, middleware, and user interface and application software. The Android SDK provides an extensive set of application programming interfaces (APIs) that is both modern and robust.Android handset core system services are exposed and accessible to all applications. On Android Market, where everyone can download thousands of applications for Android, anyone can upload their programs.Android has emerged as a newmobile development platform, building on past successes and avoiding past failures ofother platforms. Android was designed to empower the developer to write innovativeapplications [2-3].

Several studies have been developed using the Android application. [4] have been using LBS component and LBS architecture contained in the Android operating system.This Application identifies the precise location of human presence system. [5] developed Document Plagiarism Detection Software on Android Smartphone. [6] proposed a reminder system for busy people based on Android. Music player based on Android also developed by [7] on their research.

Game technology can be described as an engine that is used to develop computer-based video games through integrating visual technology, digital technology, simulation technology, intelligent technology, interactive technology, network technology and multi user operating technology [1]. The development of science and technology made people pay attention to mobile, portable and convenient, which makes the mobile game arises at the historic moment. The mobile game based on Android, is gradually popular in the crowd. The game in the research with a good picture and a wide variety of resources has the strong playability and a strong practical value. 2D game is one of those applications on the android platform. Android application are written in Java language. These are compiled into byte codes which is changed to a Dalvik executable file (.dex file) which is further compiled in to android package file (apk file), which can be installed on the Android devices [3].

Several studies of android-based game technology have been widely developed. [8] has developed “Nutrients Hero” game on Android device. This game is expected to be not only entertaining, but also giving knowledge to users. [9] on their study developed an educational game “Merah Putih” on the Android platform that provides information on Indonesian Culture. [10] made a mathematics game on Android. This game focuses on amusement and challenges. “Tales of Mamochi” game with role playing game concept based on Android have developed by [11].

The main objective is to develop a mobile phone game of high quality that will attract a lot of new users and strengthen our marketing position within the mobile phone gaming industry. The game performance objective is to make a game that will be robust, and run with good graphics without any issues. It must also meet the requirements for Android applications and support the standards for correctly installing, starting, running, pausing and exiting the application. The game in the study is called “Star Ship Adventure”, which is an adventure and agile game for android smartphone.

2. Method

The movement on a game refers to algorithms that turn decisions into some kind of motion. All movement algorithms have this same basic form. They take geometric data about their own state and the state of the world, and they come up with a geometric output representing the movement they would like to make. For example, some movement algorithms require very little input: the position of the character and the position of an enemy to chase. Others require a lot of interaction with the game state and the level geometry. A movement algorithm that avoids bumping into walls, for example, needs to have access to the geometry of the wall to check for potential collisions [12].

Collision detection is a fundamental problem in a wide range of fields. Collision detection has been the hot issue in the computer graphics recently. Various collision detection applications are found in robotics, computer graphics, and 3D computer games. Collision detection concerns the detection of collisions betweenobjects in the virtual environment. Primarily employed to stop objects moving through each other and the environment.Collision Detection is everywhere in computer games: betweencharacters and characters, between characters and terrain, etc.Collision detection refers the detection of the intersection of two or more objects.Two forms of collision detection: 1. Continuous: very expensive. Simulate solid objects in reallife. 2. Discrete: objects will end up with penetrating each other [13-15].

A bounding box is simply a box drawn around an object which contains the entire object. It can berepresented as a rectangle with an x and y coordinate, as well as a width and height. A common occurrence of this is in 2D video games where it is used for collision detection, as shown in Figure 1.

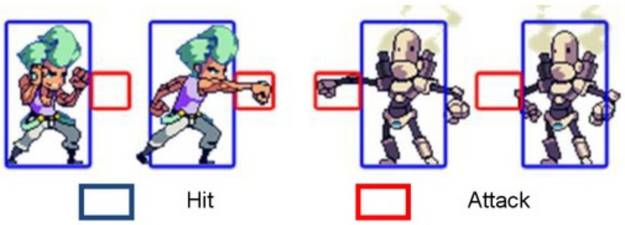

Figure 1. Bounding Box

[13] on their research describes intersection detection between a pair of bounding box. They simplifying geometry entities and assume that the object is axis aligned bounding box.

Rectangle collision detection is a collision detection in a square (bounding box). Every object created in the game has its own bounding box. This method is quite easy because it only needs to compare whether there is an intersection on both objects at a particular coordinate. It is ideal for rectangular objects. To specify the region bounding box collision detection on the object (see fig.2), use the following formula on the eq. 1

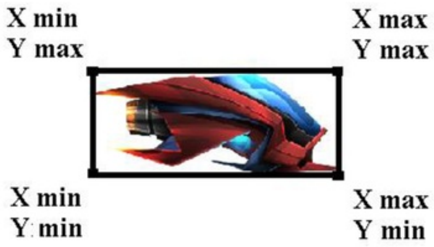

Figure 2. Area of Bounding Box

$$R\ \text{area} = \{ (x,y) \mid \text{minx} \le x \le \text{maxx},\ \text{miny} \le y \le \text{maxy} \} \quad (1)$$

whereas :

R area	=	area of bounding box collision
x, y	=	coordinate x, y
minx, miny	=	the minimum coordinate x, y
maxx, maxy	=	the maximum coordinate x, y

Players on this "Star Ship Adventure" Game will control a plane to explore outer space. The plane will be intercepted by the enemy and the player must avoid the enemy. To control the plane, use the touch of a finger. If the player puts his/her finger on the screen then the speed of the plane increases and the plane will go up. If the player releases his/her finger from the screen then the speed of the plane becomes reduced and the plane will go down. Gameplay mechanism of this game, when the game starts the player will be given 10 shield (like life points). Every time a player hit an enemy, the enemy will instantly be destroyed and the player will lose his/her shield. To win the game the player must complete a predetermined distance. If the player can win the game with the fastest time, he/she will get the new fastest time record and will be displayed in the fastest time menu. The enemy will come randomly on the right side of the screen at different speeds.

Collision detection method in this game is used to detect collisions between objects. The objects in this game are enemy's and player's plane. These objects are represented logically with a rectangular primitive (bounding box). If there is a collision between objects then the bounding box will be mutually exclusive. Next response to the colliding object: 1. enemy's plane will disappear, 2. sound effect will be heard, 3. shield (life point) of the player will decrease. If the player's shield is exhausted (the value of life point is zero) then the game is over.

3. Result and Analysis

3.1 Analysis of Collision Detection Method

Collision detection is a common method used in games for collision detection between objects in the game. These objects have varied shapes, some are square, triangular, polygon, circle or the player's unique shape. For the collision detection process, these objects are generally represented logically with primitive shapes such as squares and circles (in two-dimensional coordinates), cube and sphere shapes (at three-dimensional coordinates). The primitive form that represents this object is called the bounding box or bounding circle. An object touches when these simple representations hit each other.

The object area that collides on the bounding box collision detection will be muted to each other. Testing is done by comparing the maximum value and the minimum value at the coordinates x, y. From equation 1, it can be developed formula for two regional bounding box collision detection. Two bounding box areas will be muted if :

$$AxMin \le BxMax \text{ and } AxMax \ge BxMin \tag{2}$$

$$AyMin \le ByMax \text{ and } AyMax \ge ByMin \tag{3}$$

whereas :

AxMin, AyMin = Minimum coordinate value x, y (player plane area)
AxMax, AyMax = Maximum coordinate value x, y (player plane area)
BxMin, ByMin = Minimum coordinate value x, y (enemy area)
BxMax, ByMax = Maximum coordinate value x, y (enemy area)

The condition due to the occurrence of the slice between two bound squares on the dimension x as shown in Figure 3.

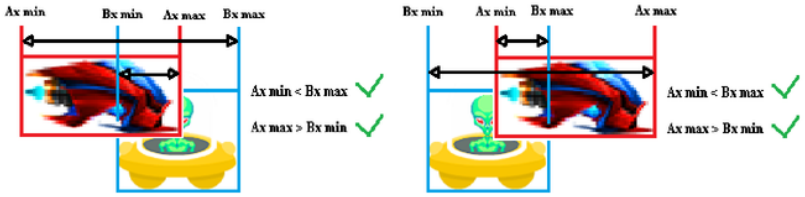

Figure 3. Bound intersect

For conditions that illustrate there is no slice between two bound squares on the dimension x side shown in Figure 4.

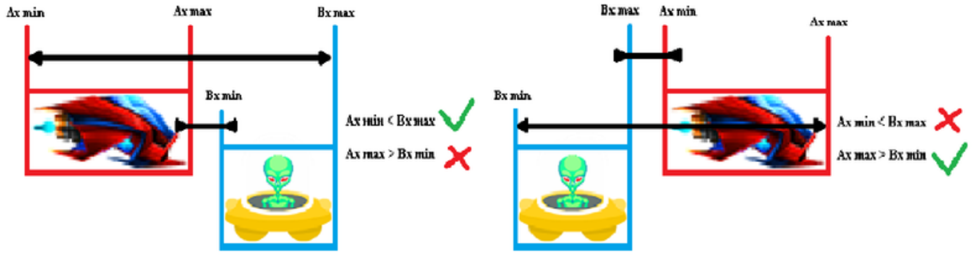

Figure 4. Bound does not intersect

Here is a graph of alternative collision detection conditions, which can be seen in Figure 5.

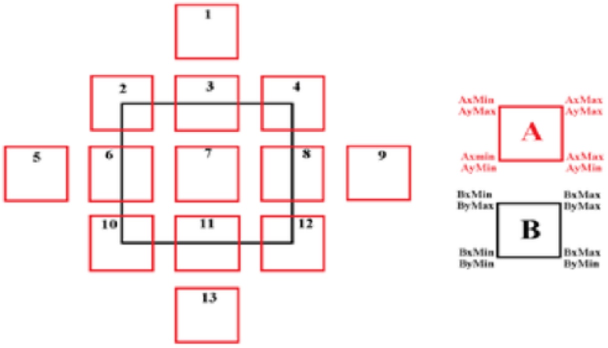

Figure 5. A Box of Alternative Collision Detection Conditions

From the picture above condition, an alternative collision detection condition table is arranged in Table 1.

Table 1. Alternative Collision Detection Conditions

Box Position	Information	Condition (based on eq. 2 and 3)			
		AxMin ≤ BxMax	AxMax ≥ BxMin	AyMin ≤ ByMax	AyMax ≥ Bymin
1	No collision	True	True	False	True
2	Collision	True	True	True	True
3	Collision	True	True	True	True
4	Collision	True	True	True	True
5	No collision	True	False	True	True
6	Collision	True	True	True	True
7	Collision	True	True	True	True
8	Collision	True	True	True	True
9	No collision	False	True	True	True
10	Collision	True	True	True	True
11	Collision	True	True	True	True
12	Collision	True	True	True	True
13	No collision	True	True	True	False

Implementation of collision detection method: 1. All in-game objects (enemy's plane and player's plane) made bounding box, 2. Initialize coordinate bounding box of both objects to detect collisions, 3. Renew the bounding box coordinates on the position of enemy's plane and player's plane. The bounding box has coordinates representing the bitmap line to detect the collision.

The response that will occur when the collision happens, based on the gameplay is the player has a shield and the enemy's plane will disappear after colliding with the player's plane. The trick is to make the enemy disappear and bring it back, as if it were a new enemy. In this step move the enemy's plane to the location of the off state from the left of the screen and change the x coordinates on the enemy object. Game objects can detect collisions and respond when a collision occurs. The way to detect collisions is to compare the player's bounding box with each enemy's bounding box. If a collision is detected, the enemy that collides with the player's plane is removed from the screen, and will reappear on the next frame.

3.2 Game System Testing on Smartphones

Once converted to apk, it will then be installed on the smartphone. Here are the results of the game app display already installed on the smartphone. In the main menu, there are 4 menu options that function differently. They are the fastest time menu, play, about, and exit. The main menu can be seen in Figure 6.

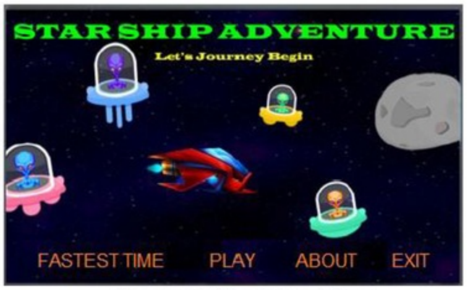

Figure 6. The Main Menu

The game display when the "play" menu is selected, can be seen in Figure 7.

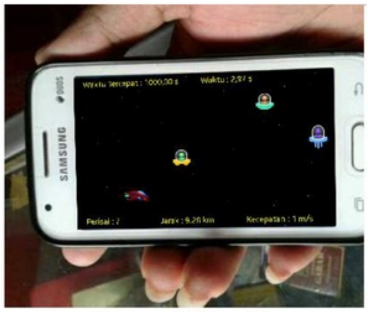

Figure 7. Play Menu

The fastest page display time, containing the fastest time earned by players, can be seen in Figure 8.

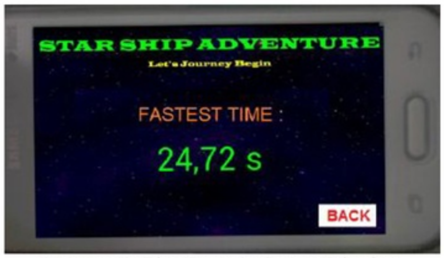

Figure 8. The Fastest Game Display

System test results are summarized in tabular form along with testing procedures, inputs, expected outputs, results obtained, and conclusions, can be seen in Table 2.

Table 2. System Test Results

Modules Tested	Test Procedure	Input	Expected Output	Results Obtained	Conclusion
Main Menu	- Open the app	Pressing on the icon application	Showing "Main Menu"	Showing "Main Menu"	Valid
Playing	- Open the app - On the main menu select start play button	Select on Play button	Showing "Game Play"	Showing "Game Play"	Valid
The Fastest Time	- Open the app - On the main menu select The fastest time button	Select on The fastest time button	Showing "The Fastest Time"	Showing "The Fastest Time"	Valid
Exit	- Open the app - On the main menu select exit button	Select on exit button	Exit from game app	Exit from game app	Valid
Background Music	- Open the app	Pressing on app	- Showing "Main Menu" - Playing Background Music	-Showing "Main Menu" - Playing Background Music	Valid
Voice Effect	- Open the app - On the main menu select Playing button	Select on Playing button	-Showing "Game Play" - Play effect music when there is a collision between objects	-Showing "Game Play" - Play effect music when there is a collision between objects	Valid

4. Conclusion

From experimental research that has been done can be concluded that "Star Ship Adventure" game application uses collision detection method. Game objects are given bounding boxes and predetermined effects on each collision occurring between objects.

This game is built using Android SDK (Software Development Kit), Android Development Tools, and Eclipse IDE using the Java programming language. The results show that the player's plane can estimate and detect collisions with enemy's plane objects. The bounding box collision detection technique occurs when a player's plane is able to read the enemy's distance object. When the player moves in the enemy's bounding box range, detection will occur, so the player's plane does not penetrate against the enemy.

References

[1] H. Guo H, H. Li, G. Chan, M. Skitmore, "Using Game Technologies to Improve The Safety of Construction Plant Technology", *Accident Analysis and Prevention*, *Elsevier*, 48: pp. 204-213, 2012.

[2] S. Holla and M.M, "Android Based Mobile Application Development and its Secure", *International Journal of Computer Trends and Technology*, 3(3), pp. 486-490, 2012.

[3] P. Kaur and S. Sharma, "Google Android a Mobile Platform : A Review", *Proceedings of RAECS UIET, IEEE*, 2014.

[4] E. Winarno, W. Hadikurniawati, R.N. Rosso, "Location Based Service for Presence System Using Haversine Method", *International Conference on Innovative and Creative Information Technology (ICITech), IEEE,* 2017.

[5] N. Nurhayati and B. Busman, "Development of Document Plagiarism Detection Software Using Levensthein Distance Algorithm on Android Smartphone". *5th International Conference on Cyber and IT Service Management (CITSM), IEEE*. 2017.

[6] K. Modi, U. Chauhan, J. Rana, C. Patel, A Rana, V. Patel, "Greeting Reminder Application Based Android", *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2), pp. 339-344, 2013.

[7] P. Yong-Chai, L. Wen-Chao, L. Xiao L, "Development and Research of Music Player Application Based on Android", *International Conference on Communication and Intelligence Information Security, IEEE Computer Society*, 2010.

[8] A. Aulia, F. Tanzil, M.F. Yudhawanto, "The Influence of Game 'Nutrients Hero' For Healthy Diet", *International Conference on Computer Science and Computational Intelligence, Procedia Computer Science 116, ScienceDirect,* pp. 81-86, 2017.

[9] N. Kidi, B. Kanigoro, A.F. Salman, Y.L. Prasetio, I. Lokaadinugroho, A.A. Sukmandhani, "Android Based Indonesian Information Culture Education Game", *International Conference on Computer Science and Computational Intelligence, Procedia Computer Science 116, ScienceDirect*, pp. 99-107, 2017.

[10] S. Chungthanacharoenlert, N. Chockwanich, K. Pasuralertsakul, C. Sukklam, V. Visoottiviseth, "The Golden Sugarcane : A Math Game on Android", *Third ICT International Student Project Conference, IEEE*, 2014.

[11] A. Kurniati, N. Nadia, F. Tanzil, F. Purnomo, "Game Development 'Tales of Mamochi' With Role Playing Game Concept Based on Android", *International Conference Science and Computational Intelligence, Procedia Computer Science 59, Science Direct,* pp. 392-399, 2015.

[12] Q. Xiaoyan, M. Ling, Y. Chengzhu, "Research of Collision Detection Algorithm Based on Hybrid Bounding Box for Complex Environment", *International Conference on Integrated Circuits and Microsystems, IEEE*, 2016.

[13] T. Koziara and N. Bicabic, "Bounding Box Collision Detection", *13th ACME Conference*, 2005.

[14] A.Weipeng and C. Zhiang, "Collision Detection Technology based on Bounding Box of Virtual Reality", *IEEE,* 2010.

[15] X. Feng, X. Shi, J. Chen, "An Algorithm of Collision Detection based on Hybrid Model", *International Conference on Computational Intelligence and Security, IEEE*, 2010.

Bounding_Box.pdf

ORIGINALITY REPORT

15%	10%	11%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	rabida.uhu.es Internet Source	2%
2	ijcttjournal.org Internet Source	2%
3	www.researchgate.net Internet Source	2%
4	Parmjit Kaur, Sumit Sharma. "Google Android a mobile platform: A review", 2014 Recent Advances in Engineering and Computational Sciences (RAECS), 2014 Publication	1%
5	www.atlantis-press.com Internet Source	1%
6	Submitted to Brigham Young University Student Paper	1%
7	Wang Xiao-rong, Wang Meng, Li Chun-gui. "Research on Collision Detection Algorithm Based on AABB", 2009 Fifth International Conference on Natural Computation, 2009	1%

	Publication	
8	Submitted to Manteca High School Student Paper	1%
9	Submitted to Informatics Education Limited Student Paper	1%
10	teacher.nsrl.rochester.edu Internet Source	1%
11	index-of.es Internet Source	1%
12	Norman Kidi, Bayu Kanigoro, Afan Galih Salman, Yen Lina Prasetio, Indrabudhi Lokaadinugroho, Arief Agus Sukmandhani. "Android Based Indonesian Information Culture Education Game", Procedia Computer Science, 2017 Publication	1%
13	Abdul-Aziz, Mohammed A., and Mehdi B. Tahoori. "Soft error reliability aware placement and routing for FPGAs", 2010 IEEE International Test Conference, 2010. Publication	<1%
14	Ding, Xue Jing. "Research on Collision Detection Algorithm Based on Combined Bounding Box", Advanced Materials Research, 2014. Publication	<1%

15	Submitted to International School of Management and Technology Student Paper	<1%
16	repository.unika.ac.id Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		