

THE DEVELOPMENT OF 3D MULTIPLAYER MOBILE RACING GAMES BASED ON 3D PHOTO SATELLITE MAP

Ahmad Hoirul Basori¹ - Riyanarto Sarno² - Surya Widyanto³

¹Department of Informatics, Faculty of Information Technology,
Institut Teknologi Sepuluh Nopember Surabaya, Indonesia (ITS)

Email: hoirul@its-sby.edu

Abstract –The growth in mobile technology cause new improvement in some mobile devices, many cellular vendors try to equipped their mobile devices with new application, such as: office application, small games, and even 3D games. New framework that has been built can be used for 3D visualization and simulation.

This paper discuss about the development of multiplayer racing game application based on 3D mobile technology for windows pocket PC platform. The new technology on direct 3D mobile will be used for handling the games and also used racing track, which has built from photo satellite and then built the 3D map. To handling multiplayer, systems have combined the application with Dead reckoning algorithm. This algorithm quite easy and very useful to handle a lot of packet that sent through wireless media (WIFI). On testing phase, there is one testing that has been done for comparing suitable devices for this game application.

Keywords: Direct3D Mobile, Multiplayer Game, Dead Reckoning, 3D Map.

1. INTRODUCTION

3D Programming on mobile platform is new technology that can be applied to some mobile devices. But not every mobile device can run 3D application; it needs some addition framework and API, such as: java JSR-184 for hand phone or communicator and for *Personal Data Assistant* (PDA) can use Managed Direct3D Mobile component [11] on .NET Compact Framework v2.0 or use OpenGL SE component on C++.

In this paper will be discussed about how to apply 3D programming on mobile devices using Managed Direct3D Mobile on .NET Compact Framework v2.0, and using *Wireless Local Area Network* (WLAN) as communication media on this multiplayer game.

In addition this paper also compares the application performance based on two devices, in order to get result which one the pocket PC devices that suitable for this game.

2. MANAGED DIRECT3D MOBILE PROGRAMMING

Managed Direct3D Mobile is addition component on .NET Compact Framework v2.0.

Thus, it needs windows mobile operating system 5 and .NET Combact Framework v2.0 [5] to run those applications.

Managed Direct3D Mobile is component that derived from DirectX 9 which has ran on desktop platform then it simplified , so it can run on mobile devices. The component from Direct X desktop Version that has derived only Direct 3D, and Direct 3D mobile more similar to Direct 3D on DirectX 8, because there are some facilities that has been erased such as: *vertex shader* and *pixel shader* [11].

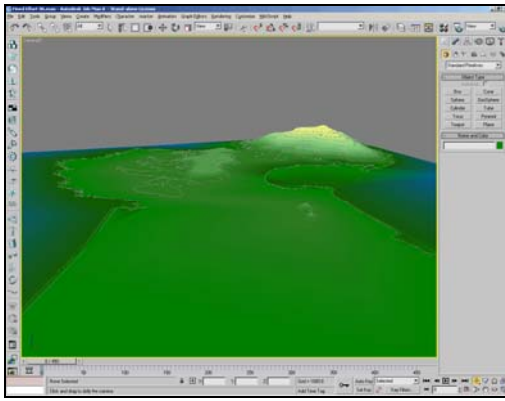
On several devices like PDA with ARM processor, Managed Direct3D Mobile will show different act, because those processor don't have ability to process *floating-point* [11]. The consumption of memory will become a problem when developing application based on Direct3D Mobile. This is because of small memory and system also save the program on RAM not on hard disk or memory card. Thus, same object will need a different memory need when used on mobile devices besides when used on Pc desktop.

Basically, The method of software development using Direct 3D mobile have a lot of similarity but the developer must be extra cautious when applied the application into mobile devices that still have limitation on graphics visualization and processor speed. One more thing that must be considered when developing application based on Direct3D Mobile, we must used PDA device that connected by cradle into our PC , because the PDA emulator still not enough because it's lack on speed and also the memory so small [16].

3. 3 DIMENSION MAP CONSTRUCTION

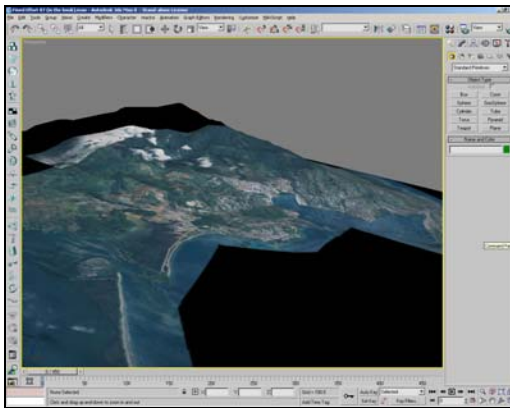
There are a lot of tools that can be utilized for developing 3D objects based on Direct3D Mobile such as: 3D Studio Max, Maya, Blender and etc. On this paper, 3D maps will be built using raw material from vector data and satellite photo's. Modeling process using 3 dimension 3D studio Max tools [15] and also using MeshX[14].

First Step, Import 3D data vector into 3D studio max application then data converted into 3D model until performs a shape like landscape, see Picture 1.



Picture 1: 3D Map Model

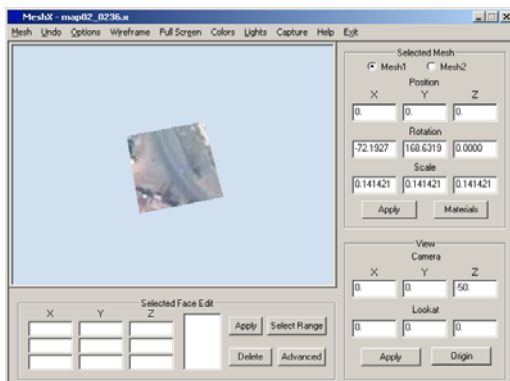
The second step is apply some texture from raster map that given by photo satellite, The photo's used for texturing and material model for 3 dimension map [15]. (see picture 2)



Picturere 2: 3D map model with texture and material

Third, simplify the model until can be loaded in mobile device that has limited memory. Fourth, Exporting 3D model dimension that already simplified using 3D Studio Max *export plugin* based on DirectX Exporter to convert into (.x) extension.

Fifth, the next process is simplifying texture that already on direct x file that have been processed by MeshX application [14](see picture 3). The result from this application still has DirectX *desktop* format.



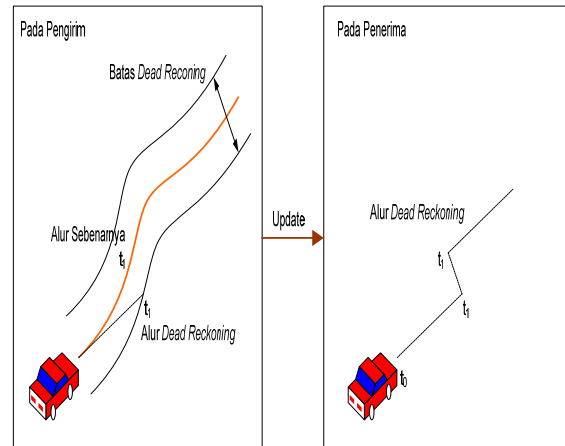
Picture 3: MeshX application

Last step, process conversion from DirectX desktop format into file that has DirectX Mobile format, this format has extension. Md.3.dm using MeshConverter that comes from Microsoft MSDN [11]. This file will be used on 3 dimensions programming on mobile devices based on direct 3D mobile framework.

4. DEAD RECKONING TECHNIQUE ON MULTIPLAYER GAME

Dead reckoning technique [3] is a technique that comes from department of defends United States of America. This technique has been used for war simulation using computer network. It's also can be used for decreasing network bandwidth needed and erasing effects from *network latency*.

Dead reckoning is computation replication which used by all parts that has participate on a games. In this case, object simulation that used is: car object complete with its complete attribute such as: position, speed, direction, etc. The main core on dead reckoning algorithm is same distribution of resource that will be used by all players. It's also can be used for predict all movement's and action that done by object's on a games application.. (See picture 4)



Picture 4: Dead Reckoning Technique

Below, The samples for dead reckoning techniques on recipient for the first loop on the game:

```
while(true)
{
    // menerima paket baru
    if (socket1.packetAvailable())
    {
        socket1.recieve(&packet1);
        info1.convertRawPacket (&packet1);
        if
        (ObjectList.member(info1.objectID()))
        ObjectList.update(info1,ctime())
    }
}
```

```

while(true)
{
// update posisi objek berdasarkan input
keyboard
MyObject.CalcNewPosition(keyboard1.readInput());

// hitung posisi Dead Reckoning
MyObject.CalcDRPosition(ctime() +
lastTimeSent);

// kirimkan koreksi posisi apabila batas
DR terlampaui
if (distance(MyObject.CurrentPos(),
MyObject.DRPos()) > thres)
{

packet1.convertToPacket(MyObject.CurrentPos());
socket1.send(packet1);
MyObject.SyncDR();
lastTimeSent = ctime();
}
}
}

```

5. THE DEVELOPMENT MULTIPLAYER GAME ON PDA

Game development has been made to explore 3D programming using Direct3D Mobile and also to communicate data among PDA using *Wireless LAN*. Game application development that has done is implemented by C# programming language on Microsoft Visual Studio .NET 2005 environment. The mobile devices that used for developing game on mobile environment are listed as below:

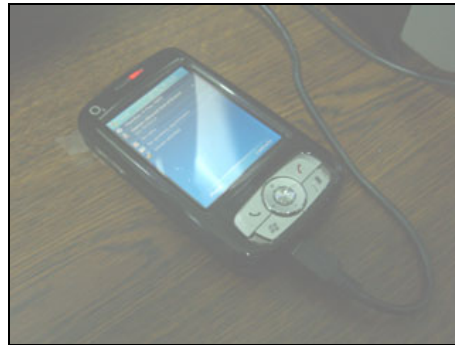
- **PDA HP iPaq hx4700** (picture 5)
 Processor : Intel xScale PXA270-624MHz
 Memory RAM : 64 MB
 Memory ROM : 128 MB
 Storage : 512 MB
 Display : VGA 640x480, 65536 Color
 Wireless LAN : IEEE 802.11b, 11 Mbps



Picture 5: PDA HP iPaq hx4700

- **PDA O₂ XDA Atom** (Picture 6)
 Processor : Intel xScale PXA270-416MHz
 Memory RAM : 64 MB
 Memory ROM : 128 MB
 Storage : 1 GB

Display : QVGA 320x240,
 262000 Color
 Wireless LAN : IEEE 802.11b, 11 Mbps



Picture 6: PDA O₂ XDA Atom

6. TESTING PHASE ON GAME APPLICATION BASED ON PDA DEVICES

• Multiplayer Game Scenario

This scenario has been made to evaluate game application on multiplayer mode. First test, will be started from first interface as main menu, choosing role as client or server, choosing the limit of maximum player, choosing map that want to be used, choosing game mode until appear the result of game.

On this scenario, the test has been done using two devices: PDA HP iPaq hx4700 as server and XDA O₂ Atom as *client* to make game run on multiplayer mode. The result is game can be run on this mode and two PDA can synchronize each other using dead reckoning techniques, like shown on picture 7 and picture 8.



Picture 7: Game Interface on Server Side



Picture 8: Game Interface on Client Side

- Comparison On Application Performance

On this scenario, testing is used to know how far game application performance will be making some differences when run on different devices. Game mode that will be played is single player mode, and then application will be run into different mode by turns. Performance measurement has done by count how many object per frame every second or it called *framerate/frame per second* (fps).

This performance measurement is processed when game is started by players and until the game finish or object car reached finish boundary. There are two devices that used on this application: PDA HP iPaq hx4700 and XDA O₂ Atom. Testing and evaluation on XDA O₂ Atom can be shown as below:



Picture 9: Testing and evaluation result on O₂ XDA Atom

The next step is testing and evaluation that has been done on HP iPaq hx4700:



Picture 10: Testing and evaluation result on HP iPaq hx4700

This testing and evaluation on this scenario has shown that application performance on PDA O₂ XDA Atom better than application performance on PDA HP iPaq hx4700 with frame rate 5.56 fps on the beginning of game and 3.85 fps at the end of game, while on PDA HP iPaq hx4700 only produce framerate 3.13 fps on the beginning of game and 2.78 fps at the end of game.

This is because of PDA O₂ XDA Atom has lower screen resolution than PDA HP iPaq hx4700. even though processor speed on PDA HP iPaq hx4700 faster than PDA O₂ XDA Atom, but screen resolution also make some big impact on application performance.

7. CONCLUSION

There are some statement that can be concluded from several testing and evaluation on software and performance before like listed below:

- According to testing and evaluation that has been done before, screen resolution give more effect on application performance rather than processor speed.
- Conversion process from 2D map into 3D map need some step until it can be loaded from mobile devices like simplifying process and segmentation until it can be used on mobile devices.
- Dead reckoning on this game still not produce excellent result, this is shown by severed object because of lack synchronization.
- For further games on the future, it will be better if consider 3D map from spatial data for better visualization.

8. REFERENCES

- [1] [HOI-03] Basori, Hoirul. Tutorial Rational Rose, IlmuKomputer.com, 2003
- [2] [DOU-03] Boling, Douglas. Programming Microsoft Windows CE .NET, Third Edition, Microsoft Press, 2003
- [3] [GAM-97] Dead Reckoning: Latency Hiding for Networked Games
http://www.gamasutra.com/features/19970919/aronson_01.htm
- [4] [DEI-05] Deitel, H. M. Visual C#® 2005: How to Program, Prentice Hall, 2005

- [5] [WIK-06] Wikipedia, English, 2006. "History of .NET Framework"
http://en.wikipedia.org/wiki/.NET_framework
- [6] [WIK-06] Wikipedia, English, 2006. "Multithreading".
<http://en.wikipedia.org/wiki/Multithreading>
- [7] [BOO-05] Introduction to Multiplayer Game Programming
<http://trac.bookofhook.com/bookofhook/trac.cgi/wiki/IntroductioToMultiplayerGameProgramming>
- [8] [STE-03] Makofsky, Steve. Pocket PC Network Programming, Addison Wesley, 2003
- [9] [FRA-03] McPherson, Frank. How to Do Everything with Your Pocket PC, Third Edition, McGraw-Hill, California, 2003
- [10] [TOM-03] Meigs, Tom. Ultimate Game Design: Building Game Worlds, McGraw Hill, California, 2003.
- [11] [MSD-05] Microsoft Developer Network (MSDN) Library for Visual Studio 2005
<http://msdn.microsoft.com>
- [12] [SUH-02] Suhendar, A, Visual Modeling Menggunakan UML dan Rational Rose, Informatika Bandung, Bandung, 2002
- [13] [GOT-03] The .NET Compact Framework QuickStart Tutorial
<http://samples.gotdotnet.com/quickstart/CompactFramework/doc/>
- [14] [MES-05] MeshX by Bud Baker, 2005.
<http://www.spinnerbaker.com>
- [15] [ONN-05] 3D Modelling Tutorial by Onno van Braam 2005
<http://www.onnovanbraam.com/index.php?tutorials/pm-1-intro/>
- [16] [WOR-05] cfWorldWind - 3D World Application, 2005
<http://www.brains-n-brawn.com/cfWorldWind>
- [17] [AND-05] Panda DirectX Exporter for 3D Studio Max by Andy Tather
<http://www.andytather.co.uk>