World Maps Conversions for 3D Mobile Games Working in Real-time Environment

Maciej Kopczynski
Faculty of Computer Science
Bialystok University of Technology
15-351 Bialystok, Poland *m.kopczynski@pb.edu.pl*

Abstract

Research shows results and describes solution and techniques allowing for conversion of highly detailed real world maps into simpler maps which can be used in real-time 3D mobile games focusing mainly on medium and low computing power devices. This approach is especially interesting for mobile 3D apps developers working with real world maps that can be implemented in games like tycoons, strategies or geolocation type. The obtained results show possibility of achieving good compromise between level of details extracted from original maps and high performance of 3D graphics generated on various mobile devices.

Keywords: map, 3D game, conversion, optimization, mobile device.

1. Introduction

Location-**B**ased **M**obile **G**ames (**LBMG**) are a subtype of broad-based games, where real world maps data and geographic location is fundamental to the game design process. In these games, mobile devices such as smartphones and tablets are used both to play and, in many cases, to determine the player's position. Example of such game is [12]. The creation and development of LBMG is more complicated than traditional games. The need for a smooth and constant flow of information from the real world maps to the virtual world maps, as well as high FPS rate of game, is a fundamental function of LBMG that affects gameplay.

As part of this work, an optimization solution was proposed that addresses this problem through functionalities in the preprocessing, analysis and visualization of large input real world GIS data sets focused on mobile devices. Hypothesis of this research is that transformation of such data by reducing details level, leads to a significant reduction in the CPU and GPU processing time on mobile devices, especially on biggest group of those devices which are low and medium computing-power types.

In the literature one can find mainly descriptions of concepts or partial process optimizations for transformation such type of data into optimized and ready-to-use form. Approach to open geospatial data integration in 3D game engine for urban digital twin applications was presented in [1] and, for industry, in [2]. Description of GIS-based educational game using low-cost virtual tour experience can be found in [3]. Example of location-based framework for mobile games is described in [4], [7] and [5]. Planetary-scale geospatial analysis platform based on Google Earth engine was proposed in [6]. Main assumptions concerning gamification and space in video games in general are described in [8] and [9]. Existing approach for maps optimization and performance analysis in games can be found in [10] and [11].

Market products like ArcGIS [14], OpenMapTiles [15], Mapbox [16] and Bing Maps [17] offer services and game SDKs that provide map data with varying levels of detail, depending on the zoom level. However, products that offer raster tiles cannot be converted into a 3D game environment because the maps are pre-rendered in 2D, making it impossible to extract detailed information from flat image data.

The paper is organized as follows. In Section 2 some information about the basic definitions are presented. The Section 3 focuses on description of solution architecture, while Section 4 is devoted to presentation of the experimental results.

2. Basic definitions and requirements

For the purpose of measuring obtained results related to maps transformations and mobile device application performance, some definitions have to be introduced. Each geographical region will have the following values measured:

- R_{points} - output number of total points defining shape of the road,
- E_{dist} - total combined road distance,
- T_{gen} - time required to generate the area,
- T_{load} - time required to load and process the region in the client environment,
- T_{frame} - time required to render the resulting geometry in the client environment,
- R_{verts} - output number of total vertices of road network mesh,
- R_{tris} - output number of triangles forming a final road network mesh.

3. Solution description

Test application was created in .NET technology using C# language and Microsoft Visual Studio 2022 on the PC side. Mobile device running Android operating system part was created in Unity 2021 LTS.

Fig. 1 presents general architecture of solution.

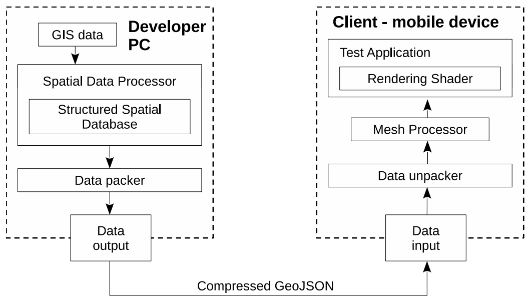

Fig. 1. Solution architecture in general.

System consists of two parts - test application running on mobile device (client side) and real-world map processing application running on PC (developer side). Transformed data is packed alongside game files and embedded into the client application using compressed GeoJSON format [13]. Main functional parts of developer PC's application are *GIS Data Source* for real-world map data source based on OpenStreetMap, *Structured Spatial Database*, which stores geospatial data using PostgreSQL database with PostGIS extension, *Spatial Data Processor*, which is key module on developer side that handles extracting only the data, that is essential to client environment. In this module, merging of the roads is achieved by transforming each of the lines into a polygonal area encompassing a distance (5km) surrounding given road geometry and simplified using the Douglas-Peucker algorithm with a tolerance of 500 m. *Data packer* is responsible for preparing collected and processed data into game data packages. Main functional parts of mobile device are *Data unpacker* responsible for unpacking the embedded data and preparing it for processing by *Mesh Processor* and further visualisation by *Test application* using *Rendering Shader*.

4. Experimental results

Presented results were obtained using a PC equipped with an 64 GB RAM DDR4 running at 3200 MHz and 6-core AMD Ryzen 5600X processor. Database engine was PostgreSQL 14 with its tablespace physically mapped to SSD drive connected via NVMe at PCIe Gen3 speed. Mobile devices used in tests were Samsung Galaxy S20, Razer Phone and Samsung Galaxy S6. Three different areas were used during obtaining experimental results. Three selected regions represent various geographical parts of the world with different properties - from the vast wilderness of Canada with low road density to the dense urban agglomerations packed on the east coast of the United States.

Parameters for the algorithms running on the PC computer (developer side) were set in a way, where each final map region after processing targeted $10 \leq V_{count} \leq 20$ cities as resulting city complexity targets and road complexity of $E_{dist} > 8000$.

Table 1 presents results obtained on PC computer related to selected regions processing in terms of preparing maps for mobile device test application. *Input* columns group describe complexity of raw data related to selected region, where *Area* presents region size, *Road network* shows total length of previously roads tagged as "motorway", "trunk", "primary", "secondary", and "tertiary" in OpenStreetMap terms and *Settlements* describes total number of settlements tagged as "city", "town" and "village". T_{gen} column presents total time from start of reading raw data to the moment, where final data package for mobile device is ready.

Table 1. Results acquired for real map processing

Region	Input			Output		T_{gen}
	Area	Road network	Settlements	E_{dist}	R_{points}	
	$[km^2]$	$[km]$	$[-]$	$[-]$	$[-]$	$[s]$
Northeastern US	348 076	89 020	1 325	19 003	5 143	207
Texas	708 011	156 635	1 141	28 818	6 749	115
Saskatchewan	504 367	32 319	179	20 017	4 365	14

As can be seen, the generation time varies drastically depending on the complexity of the region in terms of road network and number of cities, but is not significantly dependent on the size of the region itself. This is explained by the fact that the algorithms operate on geometry and graphs, and the Saskatchewan area, despite of its very vast area, do not have much of the roads.

Map data generated by PC computer (developer side) was then sent to the mobile device (client side) and test application with rendering part was used to acquire results in terms of time, what has direct impact on mobile application performance. Measurements related to complexity of generated geometry was also collected. R_{verts} describes total number of vertices of road network mesh, while R_{tris} represents number of triangles forming a final road network mesh.

Table 2 presents results obtained on different mobile devices depending on selected regions using both sets of map data: raw and processed ones. Column T_{load} describes map data loading time measured from the accessing the data, while T_{frame} presents frame rendering time. Column group *Raw map* corresponds to raw map data, which was only filtered by previously mentioned roads and cities tags, while column group *Processed map* contains measurements taken on processed by PC computer map data.

Geometrical results for filtered raw and processed map data is equal to:

- filtered raw map:
  - R_{verts} = 4 402 754 for Northeastern US; 3 690 430 for Texas; 193 471 for Saskatchewan,
  - R_{tris} = 4 213 483 for Northeastern US; 3 355 568 for Texas; 177 887 for Saskatchewan.

Table 2. Results on mobile devices for selected regions

Mobile device	Filtered raw map		Processed map	
	T_{load}	T_{frame}	T_{load}	T_{frame}
	$[ms]$	$[ms]$	$[ms]$	$[ms]$
Northeastern US				
Samsung Galaxy S20	30 350	8.53	106.32	0.03
Razer Phone	49 920	18.44	153.31	0.05
Samsung Galaxy S6	79 072	37.48	288.58	0.11
Texas				
Samsung Galaxy S20	28 222	7.01	88.72	0.02
Razer Phone	44 842	15.08	122.67	0.04
Samsung Galaxy S6	75 786	29.25	235.45	0.12
Saskatchewan				
Samsung Galaxy S20	1 526	0.36	68.67	0.02
Razer Phone	2 120	0.79	110.92	0.05
Samsung Galaxy S6	3 903	1.66	174.14	0.10

- processed map:
  - $R_{verts} =$ 12 142 for Northeastern US; 15 924 for Texas; 9 866 for Saskatchewan,
  - $R_{tris} =$ 10 592 for Northeastern US; 14 022 for Texas; 8 926 for Saskatchewan.

Timing and geometrical results shows, that each area was transformed into a similarly performing renderable package that allows the game to achieve similar performance targets across multiple devices with different computing power, including graphical capabilities. This is essential to modern game design because players expect stable performance, no matter what are their gameplay choices.

5. Conclusions

Performed research shows, that creating efficient and versatile methods for processing real world data focusing on game development industry, especially for mobile devices is possible. Proposed solution helps in speeding up creating new geolocation games, as well aids creating smooth gameplay, because it does not require huge knowledge in the field of GIS data acquisition, GIS data storage and processing and visualization of large GIS input data sets in real time in graphic engines of games on mobile devices.

Crucial challenge in solution development was targeting fixed ranges of desired city counts and total length of roads in order to achieve consistency. Both gameplay design and performance target scale nearly linearly with the length of roads and amount of cities. This relationship is then reflected in the mesh vertex count, which can be a bottleneck in a limited-resource environment of mobile GPU deployment targets. Another challenge is transforming everything into small packages that can be easily expanded and handled in isolated fashion. The conversion of all data into GeoJSON as an intermediate format facilitates the distribution, inspection, and further expansion of game assets.

Further research will focus on further development and optimization of proposed methods, especially in the field of selecting settlements, creating more general road mesh for bigger cities, adding mechanisms for creating parallel road considering requirements for different road categories, as well as testing more mobile devices with bigger regions.

Acknowledgements

The work was supported by the grant WZ/WI-IIT/3/2023 from Bialystok University of Technology. Research results are based on the project "Geo Game Service – a scalable service that provides useful data about geolocation information in asynchronous multiplayer games" financed by National Center of Research and Development.

References

1. Rantanen, T., Julin, A., Virtanen, J.-P., Hyyppä, H., Vaaja M.T.: Open Geospatial Data Integration in Game Engine for Urban Digital Twin Applications. ISPRS International Journal of Geo-Information 12(8), 310 (2023)
2. Schleich, B., Anwer, N., Mathieu, L., Wartzack, S.: Shaping the digital twin for design and production engineering. CIRP Ann. 2017, vol. 66, pp. 141—144 (2017)
3. Varinlioglu, G., Sepehr, V.A., Eshaghi, S., Balaban, O., Nagakura, T.: GIS-Based Educational Game Through Low-Cost Virtual Tour Experience – Khan Game. In: Proc. of the 27th CAADRIA Conference, Sydney, 9-15 April 2022, pp. 69–78 (2022)
4. Ionescu, G., Valmaseda, J.M.D., Deriaz, M.: GeoGuild: Location-Based Framework for Mobile Games. In: International Conference on Cloud and Green Computing, pp. 261–265 (2013)
5. Predescu, A., Mocanu, M., Chiru, C.: A case study of mobile games design with a real-world component based on Google Maps and Unity. 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, pp. 1-6 (2021)
6. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R.: Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sens. Environ. 202, 18–27 (2017)
7. Lee, A., Chang, Y.S., Jang, I.: Planetary-Scale Geospatial Open Platform Based on the Unity3D Environment. Sensors 20 (20), (2020)
8. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From Game Design Elements to Gamefulness: Defining Gamification. In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, pp. 9–15 (2011)
9. Wolf, M.: 3 Space in the Video Game. The Medium of the Video Game; Wolf, M. (Ed.); University of Texas Press: New York, NY, USA (2021)
10. Abubakar, A., Zeki, A.M., Chiroma, H.: Optimizing Three-Dimensional (3D) Map View on Mobile Devices as Navigation Aids Using Artificial Neural Network. In: International Conference on Advanced Computer Science Applications and Technologies, Kuching, Malaysia, pp. 232–237 (2013)
11. Koh, E., Park, G., Lee, B., Kim, D., Sung, S.: Performance Validation and Comparison of range/INS integrated system in urban navigation environment using Unity3D and PILS. In: Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, USA, 20–23 April 2020; pp. 788–792 (2020)
12. Transportico game homepage, https://play.google.com/store/apps/details?id=com.riftcat.transportico. Last accessed 10 April 2024.
13. RFC 7946 standard homepage, https://datatracker.ietf.org/doc/html/rfc7946. Last accessed 10 April 2024.
14. ArcGIS homepage, https://www.arcgis.com/index.html. Last accessed 24 June 2024.
15. OpenMapTiles homepage, https://openmaptiles.org/. Last accessed 24 June 2024.
16. Mapbox homepage, https://www.mapbox.com/. Last accessed 24 June 2024.
17. Bing Maps homepage, https://www.bing.com/maps. Last accessed 24 June 2024.