Development of 3D UDTs for Traffic Monitoring in Unreal 5 Game Engine

David Crampen [1], Yogender Yadav [2,3], Shofia Ishar [3], Sisi Zlatanova [3], Wenwen Tian [4]

[1] Geodetic Institute, RWTH Aachen University, Germany - crampen@gia.rwth-aachen.de
[2] Inter-university Department of Regional & Urban Studies and Planning (DIST), Politecnico di Torino, Italy-
yogender.yadav@polito.it
[3] School of Built Environment, University of New South Wales, Sydney, Australia- (s.ishar, s.zlatanova) @unsw.edu.au
[4] School of Geography and Information Engineering, China University of Geosciences, Wuhan, China, tianww@cug.edu.cn

Keywords: UDTs, Transport Simulations, Game Engines, 3D City Models, 3DCityDB

Abstract

Digital Twins recently, are a much regarded research topic in geospatial engineering, offering great potential for optimization of a variety of interrelated processes in urban environments. A major component of a Digital Twin use case is real-time sensor data input and analysis, which is used as foundation for simulations that lead to decision-support for authorities or even automatic response into the physical environment. The use of game engines as platform for urban Digital Twin use cases is promising due to superior visualization and animation capabilities, that facilitate diverse monitoring use cases for highly dynamic sensor data inputs. This contribution evaluates the fit for purpose of the Unreal Engine 5 game engine for digital twin use cases, by conducting a case study on urban traffic monitoring in the city of Liverpool Australia. The case study proved, that the necessary connections of static data provided via a geodatabase and dynamic real-time data from public portals can be integrated into the engine to establish an interactive Digital Twin use cases for evaluating traffic user interactions.

1. Introduction

In the evolution of urban planning and transportation management, the integration of advanced geospatial technologies has emerged as a promising pathway to address the complexities of urban environments (Zlatanova et al., 2020). Among these emerging geospatial technologies, the concept of Urban Digital Twins (UDT) has gained considerable traction, offering a realistic, dynamic, and immersive representation of the real-world within urban environments (Lei et al., 2023, Diakite et al 2022). The development of a 3D Digital Twin for urban transportation requires combining four different components, a digital representation in form of a 3D city model (Cureton & Hartley, 2023; Dembski et al., 2020, Li et al 2020), acting as static basis for human interaction, when visualized and as reference for dynamic input data. The second component of a UDT is a link from the real-world environment to the digital representation via IoT sensor data, such as information from users of urban infrastructure, which is streamed into the digital model (Cureton & Dunn, 2020). Third, a data analysis component is required, capable of processing the live data as decision-support, which in case of an urban transportation use case could be the identification of high-risk scenarios between different users (Rezaei et al., 2023, Aleksandrov et al 2019). Lastly, the outcome of the analyses conducted must deliver feedback into the physical world, which could be a notification for a user or live actuation of traffic guiding systems to prevent high-risk events.

Game engines are powerful platforms that can be used for the development of dynamic virtual environments like 3D UDT (Rantanen et al., 2023). Popular gaming engines like Unity, Unreal Engine 5, and TwinMotion enable the realistic simulations of cities and traffic flows with detailed structures and interactions with the physical world (Epic Games, 2024; Unity Technologies, 2024). Game engines have been widely employed as the instrumental tools for developing applications using Urban Digital Twins, enhancing public participation and planning processes with detailed and realistic project visualizations (Aleksandrov et al 2021, Aleksandrov et al 2019).

In today's rapidly evolving cities with urban sprawl, there is a need for more efficient and adaptable traffic monitoring systems. However, traditional methods often fall short in providing real-time insights and dynamic solutions to traffic congestion within cities. Hence, the development of 3D UDT twins fills a crucial gap in the existing research methods by offering a cost-effective and visually comprehensive alternative. UDTs have the potential to support urban planning processes efficiently by providing decision-makers with accurate and detailed simulations of traffic scenarios. Though in theory, the ideal Digital Twin use case would rely on full automation of interactions between virtual and physical twin (Chen et al., 2021), this vision is far from reality, especially in highly complex environments like cities. In such cases, it is sensible to choose a less mature Digital Twin configuration relying on human interaction, since the relevance of specific factors for the target use might not be known in advance and subject to the analyses conducted in the employed use case. Therefore, a visualization can be highly beneficial for identification of dependencies between objects and events, by combining the visualization with analytic capabilities of a digital environment.

To minimize the uncertainty of replication of the real world, game engines represent a great opportunity, given they involve advanced physics engines, that allow realistic simulations like Unreal Engine 5. For this Digital Twin configuration game engines might even be the best fit, since they have been designed solely for human interaction in games. Furthermore, the large scale of a UDT use case, requires a high-performance rendering platform to create a real time capable Digital Twin.

The paper investigates the potential of game engines, facilitating the visualisation capability of the Unreal 5 game engine in conjunction with the geospatial data sources. The primary aim of this UDT use case is to visualise how cities are laid out from an infrastructure point of view, how users of urban infrastructure interact, how transportation systems function in an urban environment and how game engines can be leveraged to reach a high-fidelity visualization that allows human monitors to interact with the Digital Twin.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-131-2024 | © Author(s) 2024. CC BY 4.0 License.

The novelty of our research lies in the fusion of state-of-the-art gaming technology with practical urban challenges using geospatial data sources, to establish a visually immersive and interactive platform for monitoring traffic flows. We target the interfaces between the engine, and the typical Digital Twin components, such as the 3D environment, live data input, real time data analysis, database compatibility and decision-support. Specifically, our primary goals are as follows:

i) Evaluation of Unreal Engine as platform for UDT
ii) Efficient data input from a PostgreSQL Geodatabase within Unreal Engine 5
iii) Identification of potential issues regarding Digital Twin implementation in Game Engines
iv) A Digital Twin use case for real-time monitoring of traffic and pedestrian movement flows to aid transportation monitoring and risk management.

2. Study Area and Dataset Acquisition

The study area undertaken for the development of the transport UDT was the city of Liverpool located in Sydney, New South Wales (NSW) region in Australia. The location of Liverpool was selected because of its typical urban traffic patterns, varied urban infrastructure, a relation to modern urban planning and management issues. Furthermore, a large number of data sets were available from previous works (Diakite et al, 2022). The used datasets, resources and implemented interfaces are listed in Table 1.

3. Development Methodology for UDT

For our case study, we chose to implement the use case of traffic monitoring with Unreal Engine 5 (Epic Games, 2024) as a single source of truth (SSOT). The Unreal Engine was chosen over other game engines such as Unity3D engine, due to its rapid development, supporting more and more simulation-based applications with AI support and superior rendering capabilities for 3D environments, coupled with an open-source licence. The application of UDTs for transportation monitoring demands highly dynamic data inputs for accurate vehicle positioning. Game engines have proven to be suitable as platforms for various UDT use cases, being superior to many other solutions, in cases where dynamic data must be visualized efficiently (Rundel & Raffaele, 2023), and specifically to performing simulations (Lee et al. 2022, Aleksandrov et al 2021).

Data Class	*Dataset/ Resource*	*Source*	*Data Format*	*Interface*
Quasi-Static Data	Roads	3D-CityDB	GDB & SHP	Cesium for Unreal
	3D Buildings	3D-CityDB	GDB	Cesium for Unreal
	Inter-sections	3D-CityDB/ NSW open data	CSV	Native
Dyna-mic Data	Live transport data	NSW data portal	GTFS	HTTP/ MQTT
	Pedestrian Movement Data	Fabricated	Shape-file	MQTT

Table 1: Datasets and resources used for the development of UDTs for traffic monitoring in Unreal Game Engine 5.

The geospatial data for the city of Liverpool as shown in Table 1 are stored in 3DCityDB (Yao et al 2018). We denote this data as quasi-static, since updating of the city model, in cases where parts of the city have changed over time is out of scope for this case study, while still being a very important aspect of Digital Twinning with use cases running over long periods of time. It must be clarified that all data within a Digital Twin environment is practically dynamic data. The distinguishing factor however, is the length of the update cycles for the respective data domain. In that way, one can distinguish between data, that is updated within very short time frames and data where update cycles are relatively long, such as the 3D environment. A major limitation for UDT use cases in general is the cost of repetitive data

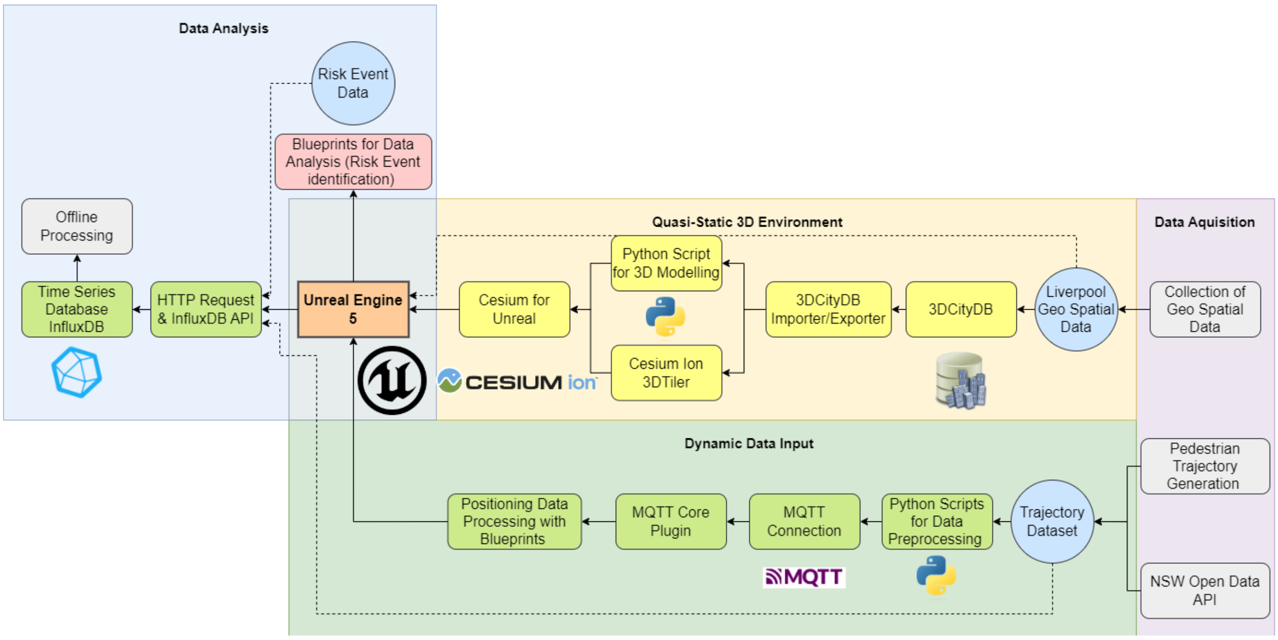


Figure 1 Schematic Overview of the different Components used in the development of UDT.

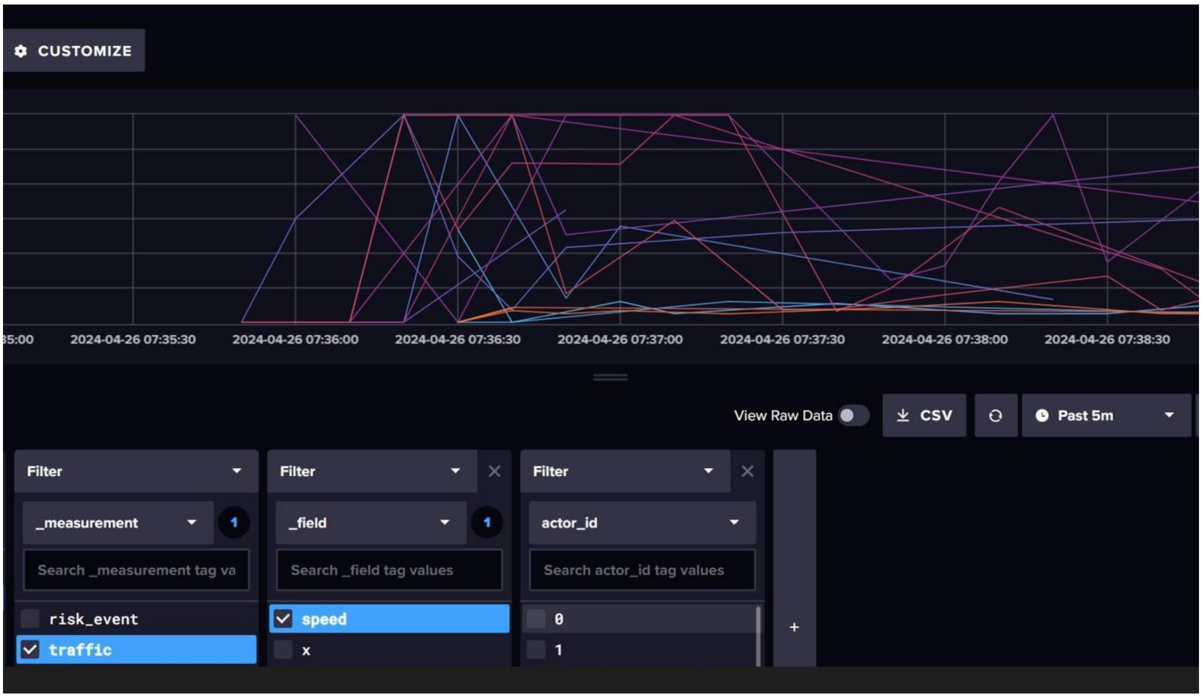


Figure 3. InfluxDB User Interface with data streaming from Unreal Engine.

collection. For a slowly changing large scale environment, it is just not feasible to conduct surveys within short cycles, while sensors attached to a specific object usually remain in position and keep sending data, as soon as they are installed and maintained relatively cheaply. The workflow for this research work is depicted in Figure 1.

3.1 Quasi-Static 3D Environment

One major aim of our case study was the evaluation of the potential of data input streaming from a geospatial database into Unreal Engine to provide an efficient static environment for the chosen use case. One very efficient way of handling large environments as in our case, is tiling the environment to only load and visualize the area visible by the user of an application. Within Unreal Engine the Cesium Ion Plugin already supports tiling using the 3DTiles OGC Standard (OGC 3DTILES Standard). Additionally, Cesium Ion can digest CityGML models in a limited manner, converting CityGML buildings into 3DTiles data, that can be directly integrated into an Unreal Engine Scene using the Cesium for Unreal Plugin.

However, using this pipeline, largely limits the contents of the CityGML dataset, not allowing import of roads and intersections. We therefore created a 3D model from the road data which was composed of polyline geometry representing the centrelines using a simple python script, that automatically generates a road mesh and sidewalk meshes on both sides of the road, loosing semantic information on the one hand, but maintaining the 3DTiles streaming capabilities by delivering it via Cesium for Unreal on the other hand. This approach was chosen, since the main interest of the road data, was to allow for additional constraints for identifying risk events, by monitoring, whether a pedestrian crosses a road or not. For the intersections, a different way of data integration was chosen by storing the intersection data in a csv file, which can be integrated into Unreal, while still being editable, where data regarding the specific intersection shall be viewable as set of properties.

3.2 Dynamic Data Input

As monitoring subjects, pedestrians and vehicles or busses were chosen, to represent both the potentially weakest and the strongest users in an urban traffic environment. The sensor data in the form of positioning information and timestamp for each of these users was either simulated in the case of pedestrian movement or obtained via the NSW open data API for public transport actors. Busses were chosen in this case, due to their availability. Figure 2 shows all relevant dynamic objects in the simulation.

The API response is in General Transit Feed Specification (GTFS) format, which is parsed for relevant data using Python. After processing the data into homogenous messages composed of location, actor ID and timestamp, they are streamed into Unreal Engine 5 using an MQTT connection, which is an open-source data transmission protocol for IoT sensors. The MQTT protocol was chosen over other data transmission methods such as REST APIs or HTTP requests, due to its lightweight structure, leading to better scalability, coupled with simple usability for IoT applications (Rantanen et al., 2023).

For highly dynamic use cases, live data analysis has to be light weight, due to time criticality. However, for future prediction of more complex scenarios and error adaption, it can be sensible to perform offline processing as well.

Using the gathered data and output of the decision-support system at specific points in time to optimize decision-making iteratively enables the step-by-step introduction of deep learning into the process.

For this reason, we keep our online data analysis component for real-world actuation light weight, by only implementing a scenario identification regiment for actors in our environment to generate feedback to a human monitor and encode the information for storage in a time series database for offline processing.

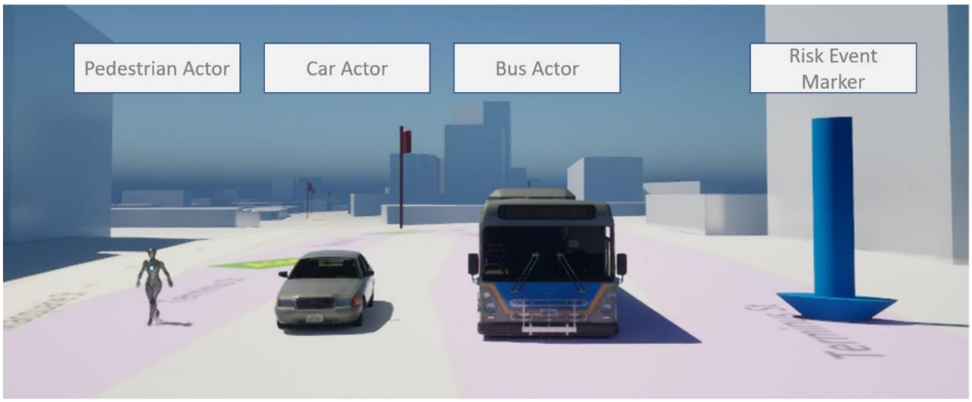
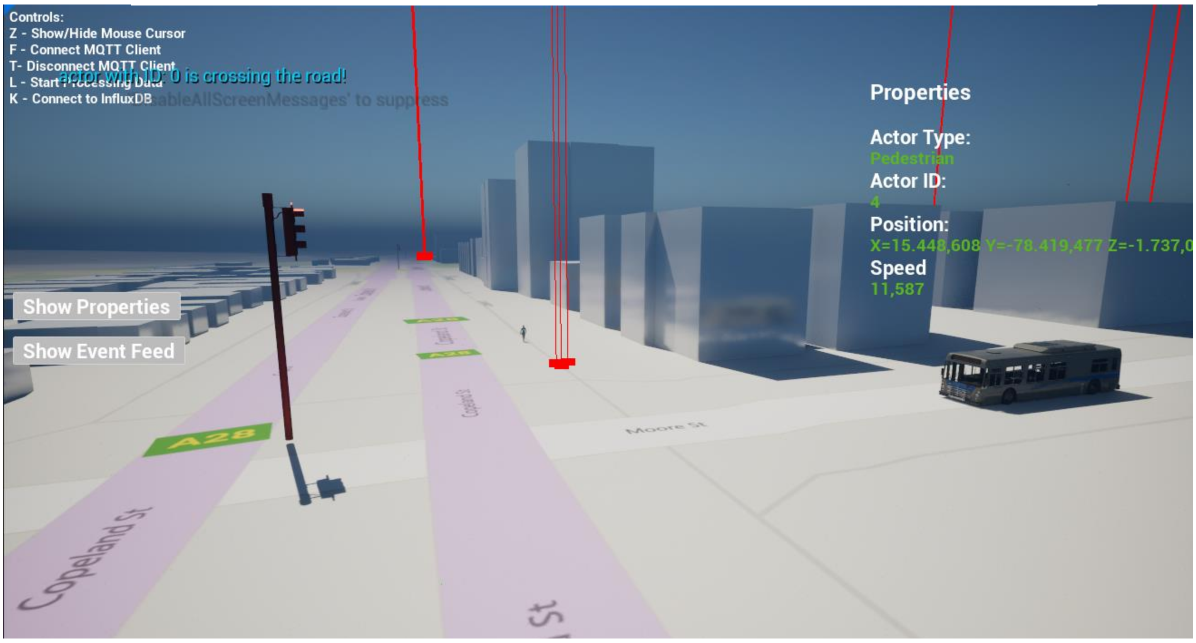


Figure 2. Simulated Actors & Objects in Case Study


Controls:
Z - Show/Hide Mouse Cursor
F - Connect MQTT Client
T- Disconnect MQTT Client
L - Start Processing Data
K - Connect to InfluxDB

actor with ID: 0 is crossing the road!

Properties

Actor Type:
Pedestrian
Actor ID:
4
Position:
X=15.448,608 Y=-78.419,477 Z=-1.737,0
Speed
11,587

Figure 4. Snapshot of Properties visualization of Pedestrian Actor in Unreal Engine 5.

For pedestrian-vehicle interactions within an adjustable proximity, a risk event marker is spawned at the location of the incident. Both the sensor data from 10 simultaneously simulated pedestrians and 10 busses, as well as the resulting risk event positions with their time of occurrence and involved actors are stored in a local InfluxDB time series database. The connection between Unreal and InfluxDB was established using HTTP requests. Two different message types were created, the risk event is composed of the XYZ-coordinates of the incident, the angle of the current direction of both actors, the timestamp as well as actor IDs and speed, when the incident occurred, the traffic message is synchronized with the data received by the individual actors and adds the current speed to the position and timestamp coming from the MQTT message. Figure 2 shows the InfluxDB user interface, while the Unreal use case is running and posting new actor status messages.

3.3 Simulation in Unreal Engine 5

For proper simulation, a main requirement is the spatio-temporal synchronization of the actors, to being able to analyse interactions between them. While our fabricated pedestrian trajectories could be adjusted to match any cycle, the update cycles provided for the open data for busses is limited to a resolution of 10 seconds.

In a realistic scenario, update cycles might not be adjustable, which is why, there are two options in general:

1. Interpolation of time and location, which might distort the outcome of the data analysis, if an actor is not moving on a straight line from location 1 to location 2 or
2. Incorporate more information about the environment, by using the road data to employ a shortest path algorithm for the most likely route taken between messages or the implementation of Kalman- or Particle-filters for position estimation.

While option 1 is simple, it might be less accurate than option 2, which in turn adds significant overhead to the live simulation. For this contribution, we used simple linear interpolation to minimize the overhead of the simulation. Another trade-off exists between simulation appearance and actuality. Since locations might only be updated every 10 seconds, in order to minimize temporal delay between digital and physical environment, the actors should be teleported to the latest location, which however makes the simulation less interpretable for a human monitor.

Therefore, we make this trade-off optional by implementing both pathways, adding a delay of 10 seconds for the option of smoothly moving actors compared to teleporting actors. As stated in section 2.2 the simulation involves several measures of the actors’ positions, which are derived by their relative positioning to the static environment. In that way, we use the intersection data to inform the monitor on which actor is crossing a specific intersection and use the road dataset for position validation, informing the monitor, if an actor leaves his typical domain. If for example a pedestrian enters the road space, the simulation informs the monitor with the actor ID and position.

By involving the static environment into the simulation, risk events can be better identified, for example, when a pedestrian is crossing the road and a vehicle with high speed is nearby.

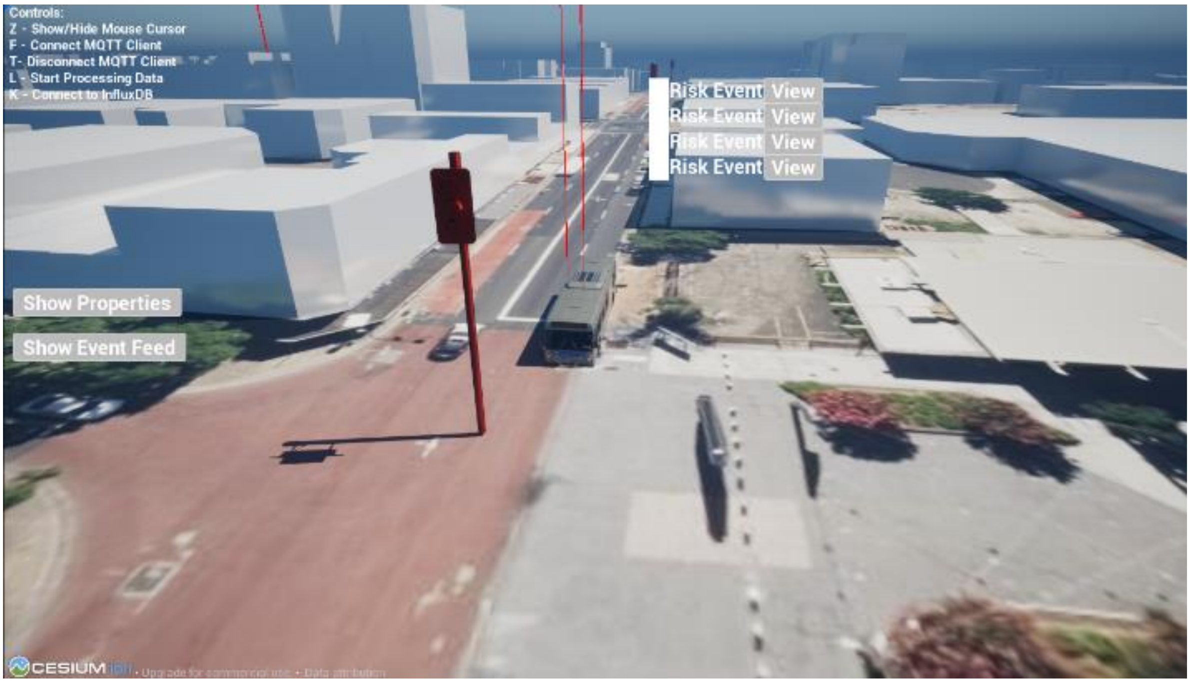


Figure 5. Example of a Bus Actor at an Intersection with past Risk Events List

3.4 Interactivity

As stated in the beginning of section 2, we chose to implement our use case within Unreal Engine, which is sensible, if a human monitor shall interact with the digital representation in some way. If the use case would run fully automatically, visualization of a digital representation would be pointless, if not for mere supervision of the actions taken automatically. Hence, we complement our use case with interactivity in form of the possibility to view the properties of all actors, as well as the properties for the functional static environment, specifically intersections. The monitor can freely navigate through the world and interact with objects around Liverpool. The captured events are listed within an event-HUD with a classification of the specific event type. Here we implemented the tracking of different events such as pedestrians crossing roads or an actor exceeding the speed limit at a position, each of the listed events can be automatically navigated by clicking. Additionally, we implemented functionality for user input bound to specific location and timestamp, that can be stored into the time series database, to allow for human feedback. Figure 4 shows an example of the visualization of properties of a pedestrian walking towards an intersection with a bus waiting for green traffic lights

4. Results and Analysis

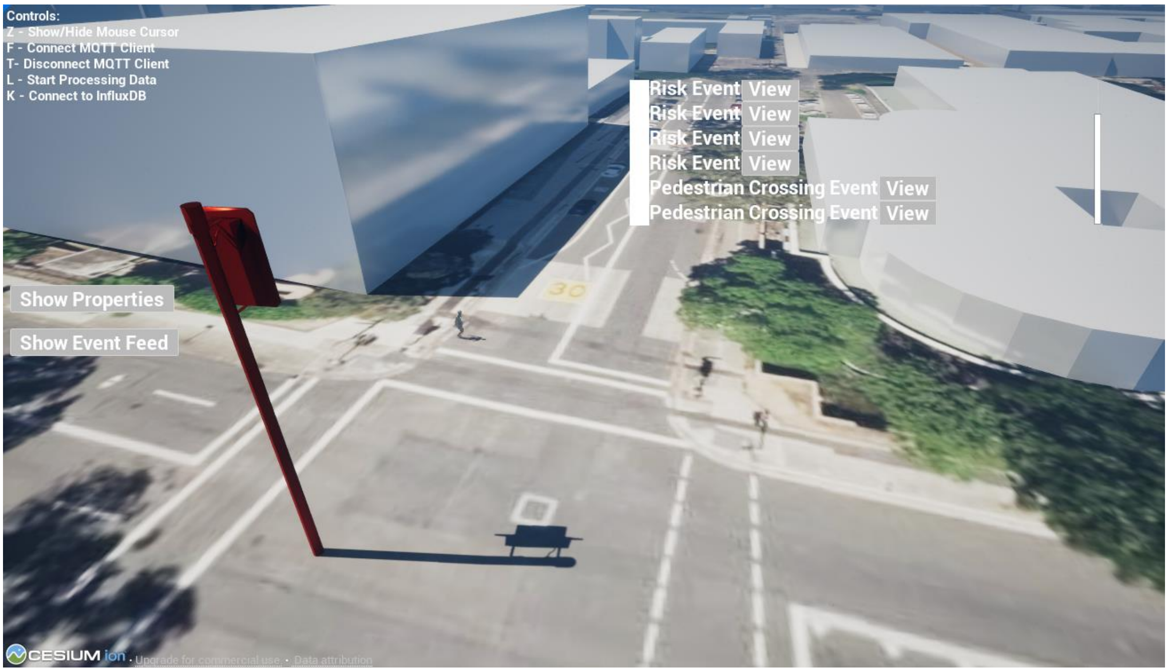
We combined several different 3D geospatial features, traffic provided by the open data portal of NSW and fabricated pedestrian movement trajectories as static 3D data. We used MQTT for the dynamic data. This integration allowed us to achieve the dynamic flow of traffic users in the streets of Liverpool. An illustration of the intermediate Digital Twin Project is represented in Figure 4, which shows the movement of the vehicles, pedestrians, and their possible points of intervention on the street.

4.1 Use Case Performance (UE5)

To test the performance of our implementation, we conducted several stress tests targeting different components. For testing performance of the MQTT connection, we varied the frequency of message publishing and the minimal time between messages. It turned out that the MQTT connection can easily handle 1000 messages per second with 10 busses. However, increasing the number of busses reduces the possible frequency of messages, since more area has to be rendered. Visualizing 80 busses simultaneously and updating all locations once per second (80 messages per second), the FPS in game fall to a mean of 25 FPS compared to 38 FPS with only 10 busses and a high frequency.

The performance decline is based on hardware limitations, since the busses outside Liverpool were used additionally leading to more area of the static environment being streamed as 3DTiles and rendered. Therefore, it can be stated that the MQTT connection is capable of a high frequency of messages, while the streaming input of our static environment is sensible for our limited hardware capabilities, originally being a Laptop with a Nvidia RTX2070, 32GB of RAM and an AMD Ryzen 7 6800H processor.

In our case study, we hosted all additional components locally, further affecting the performance of our Unreal use case, with multiple python scripts for message pre-processing running in parallel and the InfluxDB being hosted locally. Still, the performance of our use case is sufficient for actual monitoring purposes. Figure 5 shows an example of the list of risk events that occurred during simulation, while figure 6 shows a pedestrian crossing the street and spawning a road crossing event in the event feed.



Controls:
Z - Show/Hide Mouse Cursor
F - Connect MQTT Client
T- Disconnect MQTT Client
L - Start Processing Data
K - Connect to InfluxDB

Figure 6. Example of a Pedestrian Actor crossing the Road

The future development of this research study can be extended to evaluate online capabilities of Unreal Engine based UDT use cases and by incorporation of further sensor data to extend the Digital Twin system with additional use cases. Also partially updating the digital representation of Liverpool itself is a large challenge that is to be researched in future work.

4.2 Data Integration

During our case study several challenges concerning data integration had to be overcome. Since we combined different data formats, not primarily intended to be used within Unreal Engine, a direct integration, was not possible without the use of several interfaces. Especially the integration of georeferenced geospatial data turned out to be nontrivial. In the native Unreal Engine level design, a cartesian coordinate system is used, which limits Digital Twin use cases in standard Unreal Engine levels to a relatively small scale, due to earth curvature, which led to substantial differences in length depending on the relative position of an asset to the local cartesian origin.

For this reason, the use of a Plugin such as Cesium for Unreal, was necessary, since it integrates a foundation for georeferenced object placement on the Cesium earth ellipsoid within Unreal. However, Cesium for Unreal and the ArcGIS Maps SDK plugin have their respective 3D representations and are not compatible with each other within Unreal Engine. While Cesium provides an ellipsoid, ArcGIS limits the “playing field” to a local projection from Web Mercator separately. Therefore, we had to implement several python scripts to process the initial input data in Esri Shapefile format to match a local Transverse Mercator projection with respect to our local Unreal Engine coordinate system placed on the Cesium earth ellipsoid with a georeferenced anchor point. This process was necessary, since assets are interacted with in Unreal coordinates, while it is only possible to extract the georeferenced position from objects, not to reposition or even animate them via georeferenced coordinates directly.

This issue is in a way comparable with issues in BIM and GIS integration, which currently is another major topic in construction informatics (Zhu & Wu, 2022, Diakite and Zlatanova 2020). Especially the aim for direct connection to 3DCityDB on a PostgreSQL service and Unreal Engine turned out to be problematic utilizing free and open-source tools. Though there are Plugins for querying PostgreSQL databases within Unreal another substantial missing part for CityGML integration was the actual mesh construction and property management of the CityGML assets, which`s implementation would be linked to a significant programming effort. However, creating those Plugins is out of scope for this work and might be developed to supplement our use case in future research works.

4.3 Data Consistency

Regarding data quality, issues arose from non-specified accuracies for the live data and incomplete data for the quasi-static data. Knowledge of the accuracy of input data is very important, since it is a precursor for proper data analysis.

In our case the accuracy of live positions of the bus locations, was prone to relatively large variance in correspondence to the static environment usually positioning busses on the correct side of the road, but also positioning the actors on the wrong side of the road or even on the sidewalk, several times in a span of several minutes in simulation. Inaccuracies in the positioning of actors can be further increased by the reprojection into our local system, by standard floating-point inaccuracies.

4.4 Dynamic Data Analysis

Our data analysis relied on the data available or derivable by this data, so we implemented an adaptive constraint-based risk evaluation scheme, based on four adjustable constraints. The distance between actors, their difference in speed, the angle between both heading directions and an indicator for pedestrians travelling over the road were implemented utilizing the derived road dataset as bounding volume of road and sidewalks.

Supplementary data, which would have been needed to further strengthen the risk estimation would be live data of traffic light phases to determine, whether the pedestrian is rightfully crossing the road at a specific position or not, which unfortunately was not available.

Still, the IoT data collected is stored within the InfluxDB database and could be used as input for a machine learning application for risk event forecasting, given that occurring risk events can be collected as ground truth data in parallel. A potential way of collecting ground truth, would be the use of surveillance camera data, which is available for the area of Liverpool, however the automatic detection of those events would require an image classification approach, which could be subject to future work as well.

4.5 Feedback for practical implementation of the research

The component separating a Digital Twin use case from a Digital Shadow use case is the feedback into the physical world. Though for ideal implementations an automatic actuation would be desirable, since this contribution is a case study, with limited integration into administrative decision-making, our feedback is provided to a human monitor in form of the actual position and involved traffic users of a potential risk event at a certain time. As stated in section 4.3 with additional ground truth data and a risk event forecasting model, the actuation in the real world could be automized eventually.

4.6 Challenges and Opportunities of Unreal Engine as SSOT

Leveraging the gaming technologies such as Unreal Engine 5 and geospatial data in conjunction, UDTs offer immersive, real-time traffic representations of urban environments, allowing for more accurate analysis and decision-making processes. However, amidst this potential of using game engines lie several challenges, concerning the interfaces between various software platforms, interoperability between different data formats and the latency issues inherent in real-time data processing. While the Unreal Engine community is constantly developing plugins to interface between a wide range of data formats, depending on the employed supplementary tools like Cesium for Unreal in our case, data must be adapted to the specific environment, necessitating customized workflow development in many cases (Rantanen et al., 2023). The existing plugins and APIs streamline the integration process and enhance interoperability between different software ecosystems, however, still existing limitations in particular are:

- Direct data streaming from PostgreSQL
- CityGML interpretation for static environment and ingestion into Cesium for Unreal without previous conversion to 3DTiles to maintain instance properties, shapefile integration,
- ArcGIS communication with Cesium Ion for enabling ArcGIS tools for the geospatial data and proper topological path management

Another significant challenge for traffic monitoring is managing latencies and delays in real-time use-cases. In Digital Twin use cases like traffic monitoring, where the application relies on live sensor data, such as vehicle tracking with GPS systems, minimizing the temporal delay in real time data is critical to ensuring the accuracy and responsiveness to any emergency situation. However, there are several factors contributing to latency, including the processing time required to harmonize heterogeneous data sources, network delays in data transmission, and computational overhead in rendering complex 3D scenes in real-time (Lei et al., 2023; Mihai et al., 2022). To address these challenges, many solutions have been proposed to optimize latencies, such as data caching, parallel processing, and predictive modeling, to improve the overall performance of 3D UDTs for traffic monitoring (Gao et al., 2024; Ramu et al., 2022). These functionalities could be implemented into Unreal Engine to support conducting real-time simulations and managing data synchronisation consistently.

5. Conclusions

The development of 3D UDT systems has great potential to optimize various urban subdomains, such as traffic monitoring, transport management and urban planning applications. Utilizing game engines to provide high resolution visualisations in combination with large-scale multi-purpose data provided by GIS, will lead to a growing set of Digital Twin use cases, that are running in parallel to establish a more comprehensive UDT system, by facilitating synergies between the different use cases focussing on specific aspects of urban management.

In this case study, we achieved integration of static environment and dynamic traffic actors in the urban environment, which allows a monitor to analyse occurring risk events quickly. The use of Unreal Engine led to a visually appealing visualization, which improves the understanding of the context in which risk events occur. We were able to integrate actual live data from a public open data portal and store the results in a time series database connected to Unreal Engine, though the geospatial data integration was limited to 3DTiles format, due to the decision to use Cesium for Unreal, the required data was integrated using Python workarounds anyway. We could show that it is possible to implement a lightweight traffic monitoring use case within Unreal Engine using only open-source components.

However, challenges related to software interfaces and latency in real-time use cases must be addressed to fully realize this potential. By leveraging innovative solutions and interdisciplinary collaborations, researchers can overcome these challenges and pave the way for the widespread adoption of 3D UDTs as powerful tools for enhancing urban sustainability and resilience.

The identified importance of seamless data exchange between different software platforms and databases, emphasized the need for standardized formats and interoperable tools to facilitate easier collaboration and data sharing.

The development of interactive Digital Twin use cases for urban environments, will support identification of critical factors and relevant aspects of the highly complex environment, which in turn will enable the development of more sophisticated use cases relying on a high degree of automation.

However, considering certain data sources to be relevant for UDT development, to this point remains a task performed by a human actor. Providing a Digital Twin use case with a visualization enables a greater extent of user engagement and promotes collaborative problem-solving and active participation in the urban scenario exploration and optimization. Additionally, with the UDTs game engine simulations and virtual experiments, urban planners and decision-makers could test different interventions and policies in a risk-free environment, leading to more informed and effective urban management strategies. The optimization of urban settlements and built environments is a central goal of 3D UDTs in general. By simulating and analysing various urban traffic scenarios, from traffic signal optimization to pedestrian-friendly infrastructure, these digital twins enable the identification and practical implementation of strategies for improving mobility and enhancing overall efficiency of urban subdomains, ultimately leading to the smarter, more sustainable, and more resilient cities.

Acknowledgement

This research was funded by the German Research Foundation (DFG), as part of the Collaborative Research Center 339 (SFB/TRR 339) (project ID: 453596084). The financial support from the DFG is gratefully acknowledged.

References

Aleksandrov, M., Diakité, A., Yan, J., Li, W., and Zlatanova, S. 2019, System architecture for management of BIM, 3D GIS and sensor data, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sc*i., IV-4/W9, 3–10, 2019, https://doi.org/10.5194/isprs-annals-IV-4-W9-3-2019

Aleksandrov, M., Heslop, D. J. and Zlatanova, S., 2021, 3D Indoor Environment Abstraction for Crowd Simulations in Complex Buildings, *Buildings*, 2021, 11(10), 445, https://doi.org/10.3390/buildings11100445

Aleksandrov, M., Zlatanova, S., Kimmel, L., Barton, J. and Gorte, B. 2019, Voxel-based visibility analysis for safety assessment of urban environments, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*., IV-4/W8, 11–17, 2019, https://doi.org/10.5194/isprs-annals-IV-4-W8-11-2019

Chen, L., Xie, X., Lu, Q., Parlikad, A. K., Pitt, M., & Yang, J. (2021). Gemini Principles-Based Digital Twin Maturity Model for Asset Management. *Sustainability*, 13(15), 8224. https://doi.org/10.3390/su13158224

Cureton, P., & Dunn, N. (2020). Digital twins of cities and evasive futures. In *Shaping Smart for Better Cities: Rethinking and Shaping Relationships between Urban Space and Digital Technologies* (pp. 267–282). Elsevier. https://doi.org/10.1016/B978-0-12-818636-7.00017-2

Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., & Yamu, C. (2020). UDTs for smart cities and citizens: The case study of Herrenberg, Germany. *Sustainability* (Switzerland), 12(6). https://doi.org/10.3390/su12062307

Diakite, A.A, L. Ng, J. Barton, M. Rigby, K. Williams, S. Barr, and S. Zlatanova, 2022, Liveable City Digital Twin: a pilot project for the city of Liverpool (NSW, Australia*), ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., X-4/W2-2022, 45–*52, https://doi.org/10.5194/isprs-annals-X-4-W2-2022-45-2022

Diakité, A. and S. Zlatanova, 2020, Automatic geo-referencing of BIM in GIS environments using building footprints, *Computers, Environment and Urban Systems*, Vol. 80, March 2020, https://doi.org/10.1016/j.compenvurbsys.2019.101453

Gao, J., Peng, C., Yoshinaga, T., Han, G., Guleng, S., & Wu, C. (2024). Digital Twin-Enabled Internet of Vehicles Applications. *Electronics*, 13(7). https://doi.org/10.3390/electronics13071263

Li, W., S. Zlatanova, A. A. Diakite, M. Aleksandrov and J. Yan, 2020, Towards Integrating Heterogeneous Data: A Spatial DBMS Solution from a CRC-LCL Project in Australia, *ISPRS Int. J. Geo-Inf.* 2020, 9(2), 63, https://doi.org/10.3390/ijgi9020063

Lei, B., Janssen, P., Stoter, J., & Biljecki, F. (2023). Challenges of UDTs: A systematic review and a Delphi expert survey. *Automation in Construction* (Vol. 147). Elsevier B.V. https://doi.org/10.1016/j.autcon.2022.104716

Lee, A., Lee, K.-W., Kim, K.-H., & Shin, S.-W. (2022). A Geospatial Platform to Manage Large-Scale Individual Mobility for an UDT Platform. *Remote Sensing*, 14(3), 723. https://doi.org/10.3390/rs14030723

Mihai, S., Yaqoob, M., Hung, D. V., Davis, W., Towakel, P., Raza, M., Karamanoglu, M., Barn, B., Shetve, D., Prasad, R. V., Venkataraman, H., Trestian, R., & Nguyen, H. X. (2022). Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Communications Surveys and Tutorials*, 24(4), 2255–2291. https://doi.org/10.1109/COMST.2022.3208773

Ramu, S. P., Boopalan, P., Pham, Q.-V., Maddikunta, P. K. R., Huynh-The, T., Alazab, M., Nguyen, T. T., & Gadekallu, T. R. (2022). Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions. *Sustainable Cities and Society*, 79, 103663. https://doi.org/10.1016/j.scs.2021.103663

Rantanen, T., Julin, A., Virtanen, J.-P., Hyyppä, H., & Vaaja, M. T. (2023). Open Geospatial Data Integration in Game Engine for UDT Applications. *ISPRS International Journal of Geo-Information*, 12(8). https://doi.org/10.3390/ijgi12080310

Rezaei, Z., Vahidnia, M. H., Aghamohammadi, H., Azizi, Z., & Behzadi, S. (2023). Digital twins and 3D information modeling in a smart city for traffic controlling: A review. Journal of Geography and Cartography, 6(1), 1865. https://doi.org/10.24294/jgc.v6i1.1865

Rundel, S., & de Amicis, R. (2023). Leveraging digital twin and game-engine for traffic simulations and visualizations. *Frontiers in Virtual Reality*, 4, Article 1048753. https://doi.org/10.3389/frvir.2023.1048753

Zhu, J., & Wu, P. (2022). BIM/GIS data integration from the perspective of information flow. *Automation in Construction*, 136, 104166. https://doi.org/10.1016/j.autcon.2022.104166

Epic Games. (2024). Unreal Engine. https://www.unrealengine.com/en-US.

Unity Technologies. (2024). Unity. https://unity.com/ .

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T. H., 2018. 3DCityDBa 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, 3(1), 1–26.

Zlatanova, S., Yan, J., Wang, Y., Diakité, A., Isikdag, U., Sithole, G., & Barton, J. (2020). Spaces in Spatial Science and Urban Applications—State of the Art Review. *ISPRS International Journal of Geo-Information*, 9(1). https://doi.org/10.3390/ijgi9010058.