Designing a Serious Game of crisis management on top of an Agent-Based Simulation of population evacuation

Mathieu Bourgais[1,*,†], Arnaud Saval[2,†], Pierrick Tranouez[2,*,†], Olivier Gillet[3,†] and Éric Daudé[3,†]

[1]INSA Rouen Normandie, Normandie Université, LITIS UR 4108, F-76000 Rouen, France
[2]University of Rouen Normandy, Normandie Université, LITIS UR 4108, F-76000 Rouen, France
[3]CNRS, Normandie Université, UMR 6266 IDEES

Abstract

Serious games have gained significant popularity, manifesting as board games or role-playing experiences designed to train individuals to think and respond in complex scenarios that are difficult to replicate and control in real life. Concurrently, numerous agent-based simulations are being developed to explore the potential dynamics of intricate systems through the behaviors of simulated actors.

This paper introduces an innovative approach that combines these two methodologies; a serious game built upon an agent-based simulation focused on large-scale population evacuations, immersing users in realistic crisis management scenarios. This integration poses several challenges, as serious games require features that may conflict with existing simulation frameworks. These features include real-time interaction with the simulation, precise replay of scenarios, engaging displays to captivate players, and the capability to manage multiple users playing different roles simultaneously within the same simulation.

To address these challenges, we introduce the ESCAPE-SG serious game. ESCAPE-SG is based on ESCAPE, a well-established agent-based model designed to simulate mass evacuations of populations from urban areas threatened by significant natural or technological hazards. We outline the challenges encountered and the solutions developed, particularly concerning the traffic simulation central to ESCAPE. Additionally, we propose a software architecture that facilitates the connection between the agent-based simulation and a front-end display, serving as an interface for users. This architecture ensures an engaging and interactive experience for players, enhancing the training and understanding of crisis management in transportation and traffic scenarios.

Keywords

Agent-based simulation, Traffic simulation, Serious Games, Evacuation.

1. Introduction

Institutional actors may have to deal with major threats on their territory, whether of natural origin (floods, tsunamis, volcanic eruption) or technological (toxic cloud emitted by industrial plant). Decision making during these events is crucial to protect exposed people inside these areas. One way to get ready for these extreme events is to train in advance, what is currently carried out through life-size training courses for the whole crisis management unit [1]. These training sessions often take the appearance of serious games [2] where a scenario is played steps by steps with a careful analysis of results compared to other solutions at each step.

The primary goal of serious games, contrary to classical games where the objective is to entertain users, is to teach new knowledge to players through their participation and interaction with the game environment [3]. This means that even if a serious game looks like a classical game, for example using pen and paper, a set of card or other kinds of pawns, the way it works answers other standards. Most of the serious games take the form of a social simulation or a role-playing game [4] ; the game is run by a game master who helps the players make decisions in a believable but stylized environment, then the consequences of these decisions, which may put the game in a new state, are presented by the game master. At the end of the game, the game master shows and comments the results so that players

ATT'24: Workshop Agents in Traffic and Transportation, October 19, 2024, Santiago de Compostela, Spain
*Corresponding author.
†These authors contributed equally.
mathieu.bourgais@insa-rouen.fr (M. Bourgais); Pierrick.Tranouez@univ-rouen.fr (P. Tranouez)

understand what happened during the scenario they discovered. To increase the reality of the game and the panel of available scenarios, the simulation of the environment may be done through a computer simulation. The main issue is that none of the decisions made by players should lead to situations not controlled by the game master or designers; each action must make sense in terms of the training and coaching objectives, and the results of these actions, whether good or bad in terms of the game's objectives, must be, if not known, at least within the range of solutions accepted by the game designers. The use of agent-based simulation models should help avoid this pitfall.

Indeed, agent-based simulations are a meaningful framework for the study of a large variety of situations where humans interact in a very detailed environment [5]. Just concerning the use case of massive evacuation of population, there exists simulations of building evacuation [6], of cities evacuation [7] up to evacuations of regions [8], with various level of decision making complexity for the agents [9]. All these simulations, with their many parameters and random processes, produce a vast amount of data that researchers analyze to establish evolutionary scenarios based on probabilities of occurrence.

ESCAPE is such an agent-based evacuation-oriented simulation framework [7]. It is aimed at researchers whose works focus on the analysis of territorial vulnerabilities and mass population evacuation strategies. It provides intuitive tools for building spatial and social data, as well as libraries in the GAMA modeling platform [10] for fine-tuned modeling of mobility behaviors whether on foot, in a private vehicle (car, motorcycle, bicycle) or in a public vehicle (bus). Coupled with the OpenMole software [11], the ESCAPE suite can be used to explore numerous research hypotheses in "what-if" scenario and "how-to" mode, such as a the evacuation from a volcanic eruption with the impact of ashes on the road network [12]. ESCAPE requires the calibration of numerous parameters and exploration of a large number of scenarios to produce territorial diagnosis and implementation of crisis management plan. In this sense, ESCAPE is less a training tool than a support tool to evaluate different strategies and policies, which is a challenge while switching it to a serious-game in respect to the mentioned constraints. ESCAPE-SG [13] is therefore a serious game built on top of ESCAPE, but conceived and designed as a training tool for crisis management over urban areas. To do so, multiple players have to be able to interact with the running simulation of a city and its road network, with various roles and different actions to perform. At the same time, a game manager should be able to run a scenario adding external events dynamically during the simulated evacuation. Finally, the game should be able to display the useful information in real time as well as ending statistics to measure the performance of the decisions made and the actions taken. These conditions are not met with the existing frameworks where detailed agent-based simulation of evacuation are implemented.

Beyond the serious game itself, this article aims to describe the challenges encountered when integrating a real-time interactive display, an agent-based model, and a traffic simulation, as well as the solutions developed to address these challenges.

Section 2 of this article discusses the previous works done with agent based simulations and serious games, in particular in the context of mass evacuation in an urban context. Section 3 presents the challenges which come with building a serious game on top of an existing agent based simulation of an evacuation by describing how ESCAPE-SG has been built upon the ESCAPE and SUMO simulators. Section 4 details the more general discussion about the implementation of such interactive device on top of an existing agent-based traffic simulation. Finally, section 5 concludes the article.

2. Related Works

Agent based simulations have been widely used to study complex systems involving human decisions and behaviors. They enable to recreate complex social and environmental situations and test various conditions of evolution of such complex systems. This section focuses on the reviews of agent based simulations and serious games about disaster risk management and population evacuation.

2.1. Simulating population evacuation

Evacuation of population under hazardous conditions is a case of social simulations [5] with an importance given to the spatial dimension [14]: an evacuation starts with a situation under normal conditions when something happens (it may be an alert sign or an sudden event) and people switch their normal behavior to abnormal conditions. And these changes can be rapid and massive in the case of evacuation in the face of ongoing or imminent danger (tsunami), or slow and gradual in the case of preventive evacuation (slow flooding). Many works simulate the evacuation from the inside of a building with hundreds of agents [9] [6]. Indeed a close environment with a fewer number of agents enables to implement a more complex behavior, with cognitive, emotional and social dimensions as well as a fine description of architecture and geometries.

There are also city-scale simulations of evacuation. Taillandier *et al.* [10] simulate the evacuation of an urban area under flood. Each agent represents a pedestrian trying to find a shelter from a flood, following advises communicated by institutions. The same type of work may be done on a bigger geographical area with a bushfire as the hazard to evacuate from [8]. Finally, Daudé *et al.* [7] propose ESCAPE, a tool to model and simulate massive population evacuations in territories which can be described by very descriptive land-use data, social data and network transportation systems. In ESCAPE, agents which represent individual human may use different travel modes, starting as a pedestrian and then driving a car for example, to achieve their goals. They can also travel by bike, motorcycle or public transport, such as buses or subways. ESCAPE offers a powerful Driving-skill pursuit model [15] and the agents have evolving knowledge of their travel environment. They can thus have knowledge of "experienced" traffic conditions, for example by memorizing the time spent by each person in traffic jams, as well as more macroscopic traffic conditions, which allows some agents to benefit from optimized routes such as can be provided by route optimization applications. This framework has been used to simulate massive evacuation in the case of a volcanic eruption [12] and a flood [16].

2.2. Serious Games and hazardous situations

Serious games are playful activities which have, by learning, a serious goal on top of entertaining [3]. Learning is based on interactions with the model that simulates the crisis domain and on interactions between participants to collaborate and succeed in solving a certain number of tasks. Marne [17] proposes five common denominators for this type of game. The *challenges* are the problems given to the player (for example, to protect the population from imminent danger); *Significant actions* correspond to the steps taken by the player to resolve these challenges (for example, alerting the population, issuing instructions or setting up evacuation routes); the *game engine* is the simulator which reacts to the player's actions; the *graphical interface* links the engine and the player which makes it possible to give a playful aspect to both the problems and the simulator; finally, a *script* that allows the levels of difficulty offered to evolve according to the desired educational progression.

Serious games have been applied in the field of risk management in order to rise awareness about the consequences of catastrophes. In 2018, 45 serious games about disaster risk management were surveyed [4]. This includes a board game aiming at raising awareness about environmental disaster in the multicultural context of the Caribbean [18] or the "Don't Stop !" video game where users play the role of stakeholders who need to prevent damages before critical situation [19]. This latter video game has been expanded lately on the more specific topic of evacuations in front of a flood [20], proving the subject is still active in the community.

Some serious video games (that is to say serious games that uses video games technique as their core mechanics) rely on agent based simulations and multi-agent systems. Adam *et al.* propose a serious game about urban planning in the context of sustainable transport in cities [21]. In this game, users play the role of a group of people responsible for decisions on the urban landscape and the reaction of the population living in the city is generated through a multi-agent system. This technical design may be found in other works but specifically in the context of risk management and emergency evacuation as with the SPRITE game[22] or LitoSim [23]. The goal is to take actions that will have impact on the future catastrophe, the simulation is not in real time as the game simulate multiple months. With the same principle in mind, Moatty *et al.* developed a serious game about the evacuation of a population during a flood using an existing complex model of evacuation created with a multi agent system [10].

2.3. Synthesis

This section reviewed multiple agent-based simulations of large urban areas and evacuation of their population, and then discussed some existing serious games about disaster risk management. However, only few works combine these two field. From the simulation point of view, integrating game mechanisms would help popularize their results to a broader audience. From the serious game point of view, integrating an agent based simulation would make the result closer to a video game which is now a powerful language to communicate complex ideas to people [24].

One of the problem about using an agent based simulation into a serious "video" game in order to study the evacuation of populations on large urban area comes from the technologies used. Even if there exist multiple platforms to perform this type of simulation [25] [26] [27] [28], each with its own strengths and weaknesses, these tools are not suitable to integrate a complex interface which enables users to input discrete event into the simulation which operates in a continuous time. For example, for game purposes, users may need to change the road network of a city during the crisis to deflect the flow of vehicles and open a path for emergency services. But doing so may create other impacts on the overall road network of the city, which users should be able to react to.

3. From Traffic Simulations to Serious Games: Challenges

This section discusses the process used to create a serious game [15] about risk management by taking its foundations in an existing evacuation simulation tool [13]. More specifically, the objective is to create a multiplayer game where each player has a specific role and may take actions while the simulation of a city, and its traffic, is running and reacting to the decision made.

3.1. Simulating evacuation at a city level

The ESCAPE project [7] allows to simulate the evacuation of wide territories exposed to a potentially catastrophic events with citizens and civil servants (policemen, firefighters) represented by agents in the system. It's possible to first simulate the dynamics of the territory, such as the flow of vehicles under normal conditions, before injecting an event (e.g. an evacuation order or a volcanic eruption) that is perceived by agents, who will then react by modifying their behavior [29]. More precisely, the ESCAPE project is composed of the following elements :

- **Environment** : the land-use (buildings, forest, river) and road networks of the studied area are modeled. The pedestrian area are included as well as the type of the buildings (school, hospital, residential, etc.) and the bus stops. A tool using R-Shiny has been developed to directly produce environment data gathered from OpenStreetMap [30]. All these data are pre-processed before being included in the model. Building the environment with open data enables to easily adapt the project to a new use case.
- **People** : each human is represented in the simulation by an agent, and households are represented as a set of agents. In order to create an agent population at a size coherent with demographic statistics, a sample of real people is reproduced from census data. Generative synthetic population libraries [31] are then applied to this sample in order to generate a population of the correct size, with agents having characteristics as close as possible to the real studied population. This include classical demographic information (age, gender, address) but also occupations over a whole day, either for a working or for a non working day. The flow dynamics on ordinary days (i.e. without any crisis situation) is reproduced using both household travel survey and traffic data measured on the network. The agents' occupations are described with a starting time, a mean of transport

and a location to reach at a specific time. For example, someone may take their children to the school at 8 a.m. by car, go to their work place at 9 a.m. by bike, then buy some food at the grocery store at 4 p.m. by bus, get their children from school at 5 p.m. by car and go back to their home as soon as possible.
- **Vehicles** : each simulated person may use vehicles to move in the simulation, i.e. cars, trucks, buses, bicycles and motorcycles. Each vehicle is modeled by a reactive agent whose mechanism is to move from one point to another until it reaches its final destination, subject to compliance with traffic regulations. Personal vehicles (cars, trucks, bikes, motorcycles) are created when a "*people*" agent needs it and is destroyed once the travel has been done. Busses follow a timetable and they go from one bus stop to the next one on their route, looping indefinitely. Each vehicle reacts to its surrounding road conditions which means they stop if there is another stopped vehicle up front for example.
- **Hazard** : hazards and theirs dynamics can be modeled directly or have their geographical footprint uploaded from geographical information system as time-step layers defining their spatial and temporal dynamics. Interactions between agents and hazards dynamics can be modeled to reproduce casualties, or any impact on the environment (e.g. speed reduction) depending on the catastrophe implemented.

With ESCAPE, it is possible to simulate territorial dynamics, both in:

- **Situation of ordinary mobility** : each agent follows its own schedule for the day. This includes going to work, taking care of their children who go to school, going to grocery store or going back to their home among other activities. To perform these activities, agents may choose between different mobility modalities depending on their starting and ending point. Once in a vehicle, the shortest path to the destination is followed. Agents may use different types of vehicles to go to their destination target (starting as pedestrian, taking a car, then a bus and maybe ending with a bicycle). The vehicle is chosen based on the household travel survey as well as depending on the lowest estimated time to move from one point to another. For example, the household travel survey indicates that an agent takes a specific bus to go to a given location. Unfortunately, because there is currently a traffic jam in the city, the bus is very late on its usual timetable. In this case, the agent will evaluate if it is faster to wait for the bus or to choose another vehicle, depending on the vehicles at her disposal.
- **Situation of extraordinary mobility** : when perceiving a hazard (either by seeing it, by hearing an alarm or a message on the radio), agents may change their behavior and give up their normal schedule. Depending on their own characteristics and the received message, they may either choose between evacuating or confining. To do so, each agent may use multiple transportation mode depending on the configuration estimated as the fastest to get to a safe place. Once a decision is made, the agent integrates a vehicle that goes to the selected destination (either out of the city or to the nearest shelter) by following the shortest path toward this target. When arriving at a safe place, the agent is removed from the simulation.

Operationalization of crisis management is modeled through various actions taken by authorities such as the trigger of different types of alarms (global such as cell broadcast and local such as fixed sirens or mobile alarms). These alarms may be heard by people at a given distance who in return will decide to follow or not instructions. This means that not all agents either evacuate or confine shortly after the notification, but may still pursue their activities. This decision making process is fixed by the modeler through the use of different parameters or probabilities functions provided by ESCAPE tools and calibrated upon population surveys [12].

3.2. From ESCAPE to ESCAPE-SG

The ESCAPE project has been expanded into the ESCAPE-SG serious game [13] [15]. The main goal of this operation is to create a training tool which is more understandable than a simulation, but provides realistic dynamics and scenarios. The key features of the ESCAPE-SG serious game are:

- The overall setting is the evacuation of an urban environment, from a part of a town to a few nearby towns. For this, players should be able to interact with the running simulation (e.g. to close a road, to check information such as the number of peoples in a shelter or to select antennas to switch-on the alarm system).
- Several players, each with a dedicated role (i.e. mayor, civil servant, road manager), should be able to play together on the same simulation at the same time. They may act on a medium or large scale: evacuate such or such building or area, block a road, intervene on a fire, etc. but the evacuating individuals are ran by the simulation. Likewise, a game manager should be able to trigger events during the game (adding a car crash, worsening or improving the hazard condition, creating a new hazard).
- The traffic situation inside the simulation should be displayed according to different time frames: one in which simulated time is equivalent to real time to clearly understand the impact of a player's decision, and also discrete (x3, x10) or continuous (from 1 to 10) accelerated modes that allows to stay within game duration consistent with the time allowed by players or game managers.
- Taking the fact that each role has its own actions available, players need to have feedback from the game to understand the consequences of their actions.

Although the primary version of ESCAPE includes its own multimodal traffic and transportation model, ESCAPE-SG utilizes the ESCAPE model by incorporating the SUMO platform [28] for traffic simulation. Additionally, it employs a custom multi-agent system in JAVA to manage detailed individual interactions between mobile agents, extending beyond the capabilities of SUMO. This MAS acts as an intermediate between SUMO and the Unity game engine [32] which is used for the ESCAPE-SG 3D graphics front-end. Indeed, the ESCAPE simulation is running on the GAMA platform which does not enables to easily act in multiplayer upon a simulation running in continuous time. If GAMA graphical interface allows to display simulation information, it is not meant to manage intensive graphical inputs from several players at once. It is also not able of synthesizing inputs from several disjoint graphical interfaces computed on distant networked computers. Finally, GAMA cannot be either interrupted or resumed, nor have its internal values modified by external programs. It was as a consequence not the right software tool for the back-end of ESCAPE-SG.

Figure 1 shows the screen of the game manager watching a particular spot on the road network within ESCAPE-SG. Buildings and vehicles are displayed in three dimensions and the interface shows the various actions which may be taken. Each user watches a similar screen but may focus on an other place of the same simulated area. They may access only their available actions. Figure 2 shows a more macroscopic view of the simulated territory, in which traffic conditions may be distinguished, synthesized here by color gradients on the sections (red indicates traffic jam, yellow means congested traffic, green symbolizes a fluid traffic flow). At this scale, different players can perform actions on different parts of the territory at the same time: closing a motorway, triggering a siren in the city center or ordering a noria of buses to evacuate a retirement house.

4. Generalizing the process

Section 3 presented a use case where a serious game is built upon an existing agent-based simulation of a city evacuation. From this use case, this section discusses in a broader way the challenges related to the development of a serious game starting from an existing agent-based simulation of an evacuation situation of a city. The goal of this section is to present a general software architecture to ease the development of future interactive simulations, such as serious games, especially in the domain of mobility, as well as exposing the underlying challenges that arise.

Figure 1: View of the manager role in the ESCAPE-SG serious game

Figure 2: Partial overview of the simulated area in the city of Rouen

4.1. Simulation versus serious game

Agent-based simulations and serious games focusing on the same phenomenon do not have the same objectives, and so they do not use the same tools. Studying a situation with the use of simulations has the following goals :

- **Realism**: a simulation must recreate as closely as possible the behaviors of the studied situation. Fostering realism may lead to the development of complex decision making models for the simulated agents which may be difficult to explain from the point of view of an observer.
- **Validation by data**: beside demonstrating realistic behavior for the agents simulated, a simulation needs to be validated against data for a specific scenario on a real use case. This implies accessing these data beforehand and sticking as close as possible to this case. This statistical validation, coordinated with the realism of behaviors, implies once again to sacrifice a bit of explainability from the system if necessary.
- **Exploration of non monitored circumstances**: once validated on its behavior and its calibration, a simulation enables to explore unknown situations, either because they are currently not

monitored in real life or because they rarely happen. The goal is then to make predictions on "what if" scenarios which would serve to validate the behavior of the model.

On the other hand, the main objectives of a serious game about a similar situation are :

- **Education**: the primary objective of a serious game is to teach something to its participants. This may be about the consequences of a particular situations and how to handle them or about what to do to face a given situation. To do so, a scripted scenario, which may be an exaggeration of a real case, is at use.
- **Gameplay**: a serious game is still a game. Its gameplay, that is to say the rules enabling players to interact with the state of the game, is crucial to create an enjoyable experience. This may implies to alter the realism of the represented situation for pace consideration and simplicity of actions.
- **Exploring scenarios**: to ensure the game may be played multiple times by the same players, for teaching reasons, multiple variations of the same situation should be scripted. Each game may explore a new variation of the same overall scenario. These scenarios are created for the notions they may teach, not for their validity with existing data.

Because their objectives differ —simulations aim to mirror real data as closely as possible, while serious games seek to create engaging educational experiences— their technical constraints are not the same. Taking the study of urban traffic as an example, an agent-based simulation will attempt to include all human actors as agents and compute a complex decision-making process that closely resembles human decision-making, even if it takes several days to simulate a few hours of traffic. The results are obtained after all computations are completed and are then compared to real-life data.

In contrast, a serious video game addressing the same urban traffic scenario needs to condense several hours of the studied situation into a few minutes to remain enjoyable. To achieve this, the studied area may be smaller, with fewer agents and simpler behaviors, allowing computations to be performed nearly in real time. The game's response to players' actions, which is crucial for the serious game, should also be displayed in real time so players can adapt their decisions accordingly.

Szczepanska *et al.* [33] define six ways to create a collaboration between an agent-based simulation and a serious game; the ESCAPE-SG project and the challenges discussed in the current article fit in the third design type with an existing simulation at the origin of the game.

4.2. Creating a serious game from a simulation

As mentioned previously, one of the main challenges to create a serious game from an existing traffic simulation deals with the fact that players have to perform their actions while the simulation is running; in other words, this means triggering discrete events over a continuous simulation. The problem is then to have a computation time close to real time while keeping as much as possible the complex behavior of the simulation.

Figure 3 shows the software architecture developed to ensure that players could act over a running agent-based simulation from an interactive interface. The system may be decomposed in three parts : the back-end where the simulation runs, the front-end which serves as an interface with players and an API which makes the connection between the back-end and the front-end. This figures displays at the same time the overall general architecture as well as the specific technological choices made for the ESCAPE-SG project which are discussed in section 3.

To ensure at the same time low computation time and a complex decision making process for each agent, the back-end side of the architecture is decomposed in two parts, as seen on Figure 3. The basic traffic simulation is managed by a dedicated traffic simulator. At this level, vehicle agents have a simple reflexive behavior in which they have a target point to drive to using the shortest path computed on advance on the road ordinary network (no road is closed). They follow road traffic regulations with a reactive behavior and do not make complex decisions. Agents not in movement are not included.

The complex decision-making process is managed by a dedicated Multi-Agent System (MAS). This MAS computes the actions of each agent at every time step. Primarily, each agent follows its own

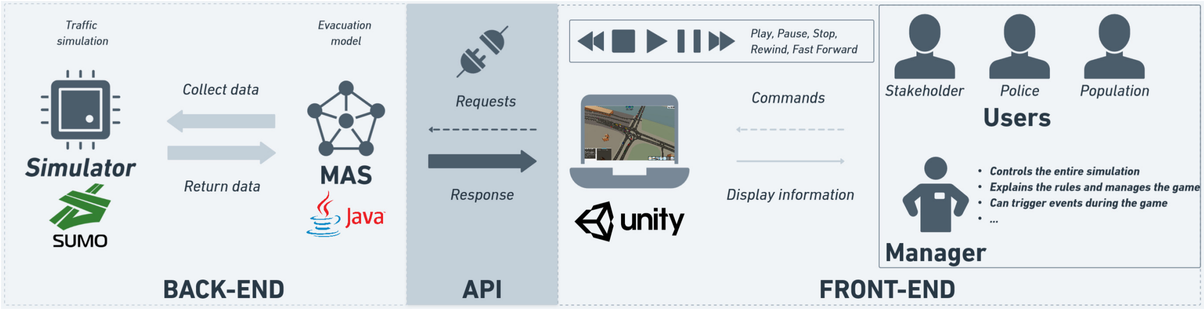

Figure 3: Global software architecture for the creation of a serious game on top of an agent-based simulation

timetable, which is not very demanding in terms of computation time. When an agent starts a new activity, the MAS determines how to carry out this activity, including which mode of transportation to use, taking current conditions into account. Once a decision is made, it is communicated to the simulator, which may then move the agent within a vehicle on the road network. In a crisis situation, the agents will react according to the ESCAPE model [7], based on the various possible alert systems, by communicating with their social network and planning evacuation or confinement.

The MAS then transmits the positions of moved agents to the API, which serves as a connection point with the front-end part of the architecture. This API may also host multiple services accessible to all front-end users, such as a chat system that enables players to communicate with each other if they are in different locations.

On the front-end side, each user has its own client displaying a graphical interface. Depending on the role of each player, the actions available upon the simulation are different. Each action performed at any time by any player is sent to a REST API as a command. This command is then passed on to the MAS which modifies the environment settings. The set of commands includes closing a road, changing the direction of traffic, opening shelters or triggering alarms for example.

Lastly, the game manager has a dedicated interface. With a dedicated set of actions, the game manager may run a given scenario acting on the environment or the hazardous condition by sending commands to the API. This way, the game manager is seen as a player with a specific role.

4.3. General discussion

To ensure an efficient serious game, in regard to the notions learnt by participants, it is important to put the player as close as possible to the real case situation [4] while having a multiplayer experience [34]. These two principles guided the creation of the modular software architecture described by figure 3.

Beside offering a complex agent's behavior with a low computation time, this modular approach enables to make dynamic modifications to the simulated environment in live conditions. A player may want to close a specific road to redirect the traffic flow during the simulation. The action is selected on the graphical interface which sends the command to close a given road to the MAS, through the API. The MAS translates this action by removing the road from the network. However, this action may change the shortest path of some vehicles. The MAS takes in charge to recompute the shortest path for the concerned agents (i.e. agents which actual path include the closed road). On the simulator's side, nothing changes as it receives the new target communicated by the MAS.

To ease the process from the point of view of players, and increase the realism of the game, these actions are differed between their decision and their effective realisation. Let's take the same example of action: closing a road with the mayor's role while the evacuation has started. In real life, closing a road implies giving an order to a worker who needs to travel to the selected point and then places a sign indicating the road is closed, a whole process that could take multiple minutes. To mimic the real world, the player needs to implement this action by selecting the specific road to close which effectively closes a few minutes after the order was given. Multiple actions follow the same principle: either their effect is differed in time or their consequences will start to have an impact multiple minutes after they were decided. This type of differed action offers time for the MAS to compute all the modifications to the agents behavior, and applying it seamlessly once the action is effectively executed.

The same problems arise on the game manager side: the hazard triggered have an effect over the simulation for multiple hours. Hence, the simulation flow of time should be altered in order to have a game covering multiple days around the catastrophe playable in few minutes/hours. With an architecture making a difference between the atomic computation of the next move and the decision making process, it is easier to pause the simulation or fast-forward it; as each part is waiting for the other, a command may be passed to one part which may disconnect until it is executed.

To sum up, connecting a graphical front-end to an agent-based simulation implies the use of a dedicated MAS created depending on the features wanted. Each interaction offered to users would need to redefine a part of this custom MAS. While the MAS is deciding what each agent needs to do, the simulator may continue to make agents do basic movement, and so users are not disturbed by a simulation pausing itself to compute the next step.

Finally, the goal for realism of the game also implies a complex graphical user interface. In real life, stakeholders have a partial knowledge of the events. By implementing one client per role, each player has access to only partial information from the simulation and needs to communicate with the other players before making a decision. With the same principle, each role only has access to a sample set of all the possible actions, reflecting its real life capacities. All these reasons imply the use of a front-end which is not integrated with the back-end, a modularity that eases a personalized display.

5. Conclusion

This article describes the creation of the serious game ESCAPE-SG [15] from the ESCAPE project [7] as an initial use case. This particular case is used to extract a more general discussion on the challenges arising when building a serious game from an existing agent based simulation, especially in the field of traffic simulation. Specifically, the main objectives of a mass evacuation simulation are not the same as a serious game, which implies the proposition of a dedicated software architecture to create a link between these two fields. Dealing with a traffic simulation at the center of the serious game adds problems related to the dynamic of the system that needs to quickly adapt to actions taken by players. The software architecture proposed in this article helps to deal with these issues by implementing a custom Multi-Agent Systems which manages the decision-making process of each agent, creating a link between a dedicated and simple traffic simulation, and the graphical interfaces of players.

A scenario over a technological hazard in the city of Rouen in France is already implemented with 10% of the real population simulated. However, this scenario has not yet been tested with crisis managers to assess its capacity to improve the existing training sessions. In the future, new scenarios on new areas will be implemented in ESCAPE-SG to demonstrate the generic nature of the project. Also, a more precise study will be conducted on the integration of trucks in the simulation and a rewind action, which would enable to come back to a previous state of the simulation, will be tested, creating new features to integrate to the dedicated and custom MAS.

Acknowledgment

This work is supported by the ANR ESCAPE project, grant ANR-16-CE39-0011-01 (French Agence Nationale de la Recherche), by the RIN Tremplin ESCAPE-SG project (Région Normandie) and by Grand Plan d'Investissement TIGA Mobilité Intelligente Pour Tous.

References

[1] V. November, L. Créton-Cazanave, La gestion de crise à l'épreuve de l'exercice EU SEQUANA., La Documentation Française, 2017. URL: https://shs.hal.science/halshs-01484782.

[2] I. Di Loreto, S. Mora, M. Divitini, Collaborative serious games for crisis management: an overview, in: 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2012, pp. 352–357.
[3] C. A. Clark, Serious games, New York: Viking (1970).
[4] A. Solinska-Nowak, P. Magnuszewski, M. Curl, A. French, A. Keating, J. Mochizuki, W. Liu, R. Mechler, M. Kulakowska, L. Jarzabek, An overview of serious games for disaster risk management–prospects and limitations for informing actions to arrest increasing risk, International journal of disaster risk reduction 31 (2018) 1013–1029.
[5] N. Gilbert, K. Troitzsch, Simulation for the social scientist, McGraw-Hill Education (UK), 2005.
[6] M. Valette, B. Gaudou, D. Longin, P. Taillandier, Modeling a real-case situation of egress using bdi agents with emotions and social skills, in: PRIMA 2018: Principles and Practice of Multi-Agent Systems: 21st International Conference, Tokyo, Japan, October 29-November 2, 2018, Proceedings 21, Springer, 2018, pp. 3–18.
[7] E. Daudé, K. Chapuis, P. Taillandier, P. Tranouez, C. Caron, A. Drogoul, B. Gaudou, S. Rey-Coyrehourcq, A. Saval, J.-D. Zucker, Escape: exploring by simulation cities awareness on population evacuation, in: ISCRAM 2019, 2019.
[8] C. Adam, P. Taillandier, J. Dugdale, B. Gaudou, Bdi vs fsm agents in social simulations for raising awareness in disasters: a case study in melbourne bushfires, International Journal of Information Systems for Crisis Response and Management (IJISCRAM) 9 (2017) 27–44.
[9] M. Bourgais, P. Taillandier, L. Vercouter, Ben: An architecture for the behavior of social agents, Journal of Artificial Societies and Social Simulation 23 (2020).
[10] F. Taillandier, P. Di Maiolo, P. Taillandier, C. Jacquenod, L. Rauscher-Lauranceau, R. Mehdizadeh, An agent-based model to simulate inhabitants' behavior during a flood event, International Journal of Disaster Risk Reduction 64 (2021) 102503.
[11] R. Reuillon, M. Leclaire, S. Rey-Coyrehourcq, Openmole, a workflow engine specifically tailored for the distributed exploration of simulation models, Future Generation Computer Systems 29 (2013) 1981–1990.
[12] O. Gillet, É. Daudé, A. Saval, C. Caron, P. Taillandier, P. Tranouez, S. Rey-Coyrehourcq, J.-C. Komorowski, Modeling staged and simultaneous evacuation during a volcanic crisis of la soufrière of guadeloupe (france), SIMULATION (2023) 00375497231209998.
[13] É. Daudé, P. Tranouez, Escape–sg: un simulateur d'évacuation massive de population pour la formation des acteurs à la gestion de crise, Netcom. Réseaux, communication et territoires (2020).
[14] E. Daudé, D. Provitolo, D. Edwige, G. David, E. Emmanuel, L. Patrice, P. Eliane, S.-G. Thierry, Spatial risks and complex systems: methodological perspectives, From System Complexity to Emergent Properties, Springer (2009).
[15] A. Saval, M. Bourgais, É. Daudé, P. Tranouez, Escape-sg-un jeu sérieux pour mieux préparer les évacuations de masse, in: 31èmes Journées Francophones sur les Systèmes Multi-Agents, 2023, pp. 128–131.
[16] C. Kevin, M.-D. Pham, Z. Arthur, Brugièreand Jean-Daniel, D. Alexis, T. Pierrick, D. Éric, T. Patrick, Exploring multi-modal evacuation strategies for a landlocked population using large-scale agent-based simulations, International Journal of Geographical Information Science 36 (2022) 1741–1783. doi:10.1080/13658816.2022.2069774.
[17] B. Marne, J. Wisdom, B. Huynh-Kim-Bang, J.-M. Labat, The six facets of serious game design: a methodology enhanced by our design pattern library, in: 21st Century Learning for 21st Century Skills: 7th European Conference of Technology Enhanced Learning, EC-TEL 2012, Saarbrücken, Germany, September 18-21, 2012. Proceedings 7, Springer, 2012, pp. 208–221.
[18] V. Clerveaux, B. Spence, T. Katada, Using game technique as a strategy in promoting disaster awareness in caribbean multicultural societies: The disaster awareness game, Journal of Disaster Research 3 (2008) 1–13.
[19] A. V. Gampell, J.-C. Gaillard, Stop disasters 2.0: Video games as tools for disaster risk reduction, International Journal of Mass Emergencies & Disasters 34 (2016) 283–316.
[20] I. A. W. Hutama, H. Nakamura, Expanding the conceptual application of "stop disasters!" game

for flood disaster risk reduction in urban informal settlements, in: International Conference on Indonesian Architecture and Planning, Springer, 2022, pp. 581–599.
[21] C. Adam, F. Taillandier, Un jeu sérieux pour sensibiliser aux enjeux d'une mobilité urbaine durable, Academic Journal of Civil Engineering 40 (2022) 29–32.
[22] F. Taillandier, C. Adam, Games ready to use: A serious game for teaching natural risk management, Simulation & Gaming 49 (2018) 441–470.
[23] E. Beck, A. Monfort, M. Amalric, B. Anselme, N. Becu, A. Laatabi, C. Pignon-Mussaud, Chapter 9 - land use management for coastal flooding prevention: a participatory simulation platform applied to camargue (france), in: P. Pereira, E. Gomes, J. Rocha (Eds.), Mapping and Forecasting Land Use, Elsevier, 2022, pp. 193–221. URL: https://www.sciencedirect.com/science/article/pii/B9780323909471000028. doi:https://doi.org/10.1016/B978-0-323-90947-1.00002-8.
[24] I. Bogost, The rhetoric of video games, MacArthur Foundation Digital Media and Learning Initiative, 2008.
[25] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, A. Drogoul, Building, composing and experimenting complex spatial models with the gama platform, GeoInformatica 23 (2019) 299–322.
[26] U. Wilensky, I. Evanston, Netlogo: Center for connected learning and computer-based modeling, Northwestern Univ., Evanston, IL (1999).
[27] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, K. Nagel, K. Axhausen, Matsim-t: Architecture and simulation times, Multi-agent systems for traffic and transportation engineering (2009).
[28] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of SUMO - Simulation of Urban MObility, International Journal On Advances in Systems and Measurements 5 (2012) 128–138.
[29] K. Chapuis, P. Taillandier, B. Gaudou, A. Drogoul, E. Daudé, A multi-modal urban traffic agent-based framework to study individual response to catastrophic events, in: PRIMA 2018, Springer, 2018, pp. 440–448.
[30] OpenStreetMap contributors, Planet dump retrieved from https://planet.osm.org , https://www.openstreetmap.org, 2023.
[31] K. Chapuis, P. Taillandier, A. Drogoul, Generation of synthetic populations in social simulations: a review of methods and practices, Journal of Artificial Societies and Social Simulation 25 (2022).
[32] Unity, Game Engine, http://www.unity3d.com, 2024.
[33] T. Szczepanska, P. Antosz, J. O. Berndt, M. Borit, E. Chattoe-Brown, S. Mehryar, R. Meyer, S. Onggo, H. Verhagen, Gam on! six ways to explore social complexity by combining games and agent-based models, International Journal of Social Research Methodology 25 (2022) 541–555.
[34] C. Roncoli, Ethnographic and participatory approaches to research on farmers' responses to climate predictions, Climate Research 33 (2006) 81–99.