Qualitative Physics for Movable Objects in MOUT

Suiping Zhou
Nanyang Technological University
Singapore 639798
asspzhou@ntu.edu.sg

Shang-Ping Ting
Defence Science & Technology Agency
Singapore 109679
tshangpi@dsta.gov.sg

Abstract

We describe our work on qualitative physics for modelling movable objects in Twilight City, a virtual environment for Military Operations on Urbanized Terrain (MOUT). Movable objects such as chairs, boxes, bottles, etc. are quite common in real life. However, the computational cost will be high if the exact physics laws are used to model the behaviour of these objects. To reduce the computational cost of simulating moveable objects in Twilight City, a framework for qualitative behaviour simulation has been proposed and integrated into our simulation engine. Example results show that the proposed framework is successful in modelling the dynamic behaviour of various movable objects in Twilight City in terms of human players' perceptions and responsiveness of the system.

Keywords: qualitative physics, virtual environments, games

1. Introduction

According to a report released by the United Nation in 1999, it is expected that 60 percent of the world population will be urban by 2030 [1]. In view of the rapid urbanization, disasters in a city may have great impact on people's lives. Nowadays, Military Operations on Urbanized Terrain (MOUT) are getting more and more important due to the increasing threat of terrorist attacks in urban areas. However, MOUT training in real urban environments is expensive and often restricted to the physical mock-ups of buildings and other urbanized terrain features. Thus, building virtual training environments for MOUT seems to be a logical alternative.

One of the major considerations in building a virtual MOUT training environment is the fidelity of the virtual environment. As movable objects such as bottles, boxes and chairs, etc. are quite common in real-life, it is important to incorporate these objects (see Fig. 1) in the virtual environments. However, in the existing MOUT simulations, besides the avatars of the soldiers and some vehicles, most objects are static. The reason for this is that modeling and implementing movable objects is a challenging task in terms of computational costs and mathematical complexities. Traditional methods to implement movable objects are based on exact physics models and numeric analysis, thus are very time-consuming [2]. The responsiveness of the system may be greatly affected by the behavior of the movable objects.

In our work, we adopt an approach of using qualitative physics to model the behaviours of movable objects in MOUT simulations. This approach will help to reduce the computational cost of simulating the behaviour of movable objects and increase the fidelity of the system.

Qualitative physics is a research area that aims to provide a new repertoire of reasoning techniques for computers. Qualitative physics attempts to model the real world using some qualitative rules rather than resort to the exact physics [3]-[5]. New modeling languages have been developed with the progress of qualitative physics. These languages describe entities and processes in conceptual terms and embody natural notions of causality [6]. Qualitative physics has been used in various areas. For instance, Marc Cavazza and his team used qualitative physics in digital arts to simulate various fancy behaviors of objects [11], [12]. Syed, Pang, and Sharifuddin used qualitative process theory to model chemistry processes [13].

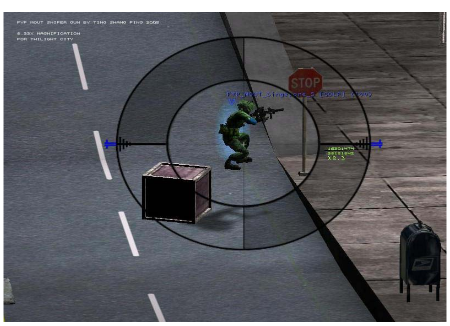
Figure 1. Movable objects in MOUT

In this paper, we describe our work on qualitative physics for modelling the movable objects in *Twilight City*, a virtual environment for MOUT. The remainder of this paper is organized as follows. In Section 2, an overview of the *Twilight City* is given. The qualitative behavior simulation framework is discussed in Section 3. Example results are discussed in Section 4. Section 5 concludes the paper.

2. Overview of Twilight City

The objective of building the *Twilight City* [15] is to create a high fidelity training and simulation system for MOUT. A typical scenario in *Twilight City* is the special squad operation to save hostages held in a building by some terrorists. The squad may consist of several human-controlled characters and some AI (artificial intelligence)-driven Non-player characters (also known as *bots*). The terrorists may also be some human-controlled characters and bots. Various tactics can be implemented in *Twilight City*, and the users (trainees) can get familiar with the real operation environments, access the effect of different tactics, and choose the best tactics to be used in the real operation.

(a) Squad operation



(b) Sniper's view

Figure 2. Twilight City

Twilight City is built using a commercial game engine, the *Unreal Tournament* (UT) engine. The motivation of using a game engine lies in the rapid growth of commercial game engines that are affordable and also provide excellent support for high fidelity 3D modeling. With the support of game engines, developers may focus on human behavior models and various tactics rather than on the 3D graphics.

Twilight City is designed to work on top of the UT engine with various modifications. These modifications adapt various UT game features to cater for MOUT simulations. With these modifications, *Twilight City* can be used as a MOUT simulation platform for various scenarios. *Twilight City* uses client-server architecture and allows for multi-player setup. In particular, an AI framework is implemented to generate realistic tactical behaviors for AI bots, a speech recognition module is introduced for human voice recognition, customized animations and special effects are used to enhance the fidelity of environment and to enrich a human player's experiences in *Twilight City*.

Various tactical scenarios are used to evaluate the performance of *Twilight City*. Case studies show that *Twilight City* has high fidelity on visual effects and the AI bots also demonstrate some human-like tactical behaviors [15]. Benchmarking tests have also been conducted to compare *Twilight City* with some commercial UT game levels in terms of 3D graphics processing, rendering speed, and responsiveness. Fig. 2 shows some screen shots of *Twilight City*.

3. Qualitative Behavior Simulation

In this section, we first describe the software architecture for qualitative behavior simulation, then explain how to model the behavior of a movable object using qualitative physics.

3.1 Software Architecture

To make the currently static objects movable, new object classes are extended from the UT engine's base classes. These new classes are added as Qualitative Physics (QP) Actors, and necessary models are attached to these Actors. A framework for implementing these QP Actors is proposed so that different movable objects could be simulated using a standard template. Fig. 3 shows the software architecture of the framework, which consists of QP Actors and a QP Engine on top of the UT engine.

The UT engine relies on event management system to support most of interactions among the objects in the game. To implement QP Actors, the event management system of the UT engine thus needs to be investigated and extended.

The UT engine implements two types of events: *primitive events* built within the game engine and *programmed events* that can be scripted by developers for customization of the object interactions. To implement QP Actors, we need to override the native event management mechanism of the UT engine. The event handlers of the QP Actors are used to detect various actions such as users' actions and collisions, and to generate various events. These events are sent to the QP engine which will determine various effects on the QP Actor based on the logic rules. The logic rules may refer to various threshold values and kinematics libraries. Based on the information returned by the QP Engine, the QP actors will then exhibit suitable behaviors in the virtual environment.

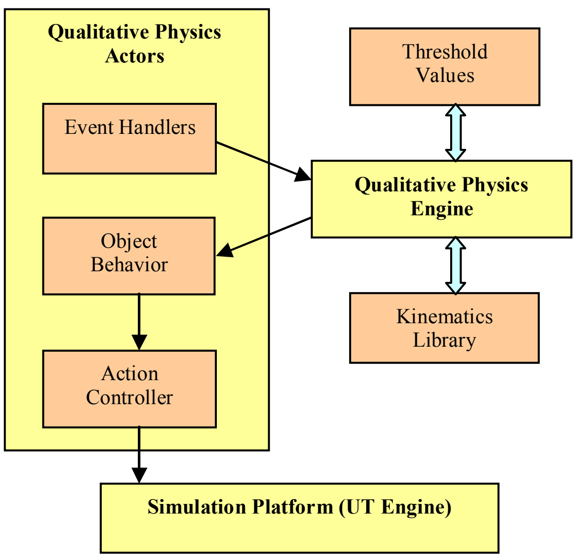


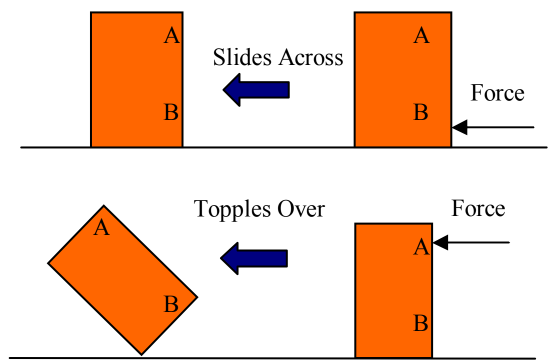
Figure 3. Qualitative physics framework

3.2 Object Behavior

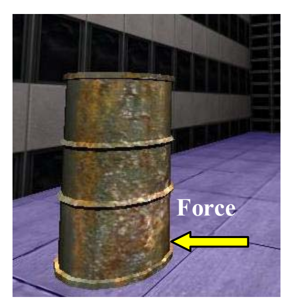
With qualitative physics, a continuous attribute (e.g., location) is quantized into a set of discrete values, and a set of desired behaviors are attached to these discrete values. The main idea of using qualitative physics is to reduce the quantitative precision of the behavioural descriptions but retain the major distinctions, thus the computational cost of simulating movable objects could be greatly reduced. The rationale of using qualitative physics to model movable objects in virtual environments is that humans in their daily lives are more sensitive to the qualitative (or causal) relations among various factors rather than the exact physics laws among these factors.

Each QP Actor will continuously update its attribute values. Once some of its attribute values reach a certain threshold, the QP Actor will send events to the QP engine. In turn, the QP engine identifies the situation and status of the object. Then, an analysis of the necessary conditions and relationships affecting the situation is conducted with qualitative process methods [5] and appropriate results will be determined. The QP engine will then signal the relevant QP Actors with appropriate behaviors.

Fig. 4 illustrates some qualitative behaviors of a barrel. Collision events are generated when the barrel (a QP object) come into contact with other objects. It is common knowledge that objects may react differently when different points of contacts are involved. For example, when a barrel is hit at a low point on its body, it will slide across the floor. However, when the same barrel is hit at a high point on its body, it will tend to fall over. In Fig. 4, contact regions A and B are marked out. When region B is hit by a force, an event (containing B) is sent to the QP engine. Assuming no other forces present, the QP engine will decide the barrel will slide across the floor. Similarly, if a force hits region A, the barrel will fall over. In this situation, the force must be large enough to overcome the inertia of the barrel for the barrel to move.



(a) Effects of applying forces at different contact points



(b) Topple over (c) Slide arross

Figure 4. Different behaviors of a barrel

Some preconditions are used to identify the current situation of a QP Actor and to select the type of qualitative physics process to invoke. Each QP process has its own set of logics and rules. In general, these logics and rules involve the relations between the states of the QP Actor and some threshold values. The QP engine will send the appropriate influences to the QP Actor which in turn uses the influences to adjust its movement parameters so that it can exhibit the appropriate behavior in the virtual environment. With this framework, an object's behavior can be easily described, and an object may exhibit different behaviors in different scenarios. For example, a heavy barrel will sink in water while a crate will float on water.

At the current stage of our project, the QP behavior of a number of common objects (e.g., crates, bottles, barrels, chairs, tables and etc.) in MOUT simulation are implemented using the framework.

4. Example Results

To enhance a human player's experiences in *Twilight City*, we have implemented various moveable objects using qualitative physics (see Fig. 5), e.g., crates, bottles, barrels, chairs, tables, etc. In this section, we demonstrate some example results of the qualitative simulation framework.

(a) Falling barrels

(b) Falling boxes

Figure 5. Moveable objects in Twilight City

The first example is on how object will behave after collision. A collision event will be generated when an object come into contact with other objects. Fig. 6(a) shows the logical formulas for the qualitative process of collision. Some preconditions are used to determine the current situations. The preconditions for collision are that there must be contact between the objects involved and at least one of the objects must be exerting a force. In our framework, when the preconditions are met, the qualitative physics engine will start the computation as required by the QP logic of the QP actors involved. For object collisions, the contact points and the forces exerted on the objects are used by the QP engine to determine the resulting behavior of the objects. After its computation, the QP engine sends the appropriate positional and rotational parameters (i.e., influences) to the objects.

Situation: Object Collision

Entities Involved: Object A & Object B

Preconditions: Contact(Object A, Object B),
(?Force(Object A)||? Force(Object B))

QP Logic: ?(Force(Object A)> Force(Object B))
?Contact_Regions(Object A)
?Contact_Regions(Object B)

Influences: Object Position Parameters,
Object Rotation Parameters

(a) Logical formula

(b) Before collision (c) After collision

Figure 6. Object collision

Another example of using qualitative physics in *Twilight City* is to simulate object behavior in water. In *Twilight City*, we have modeled the sewage system that a bot may swim in or walk through. Fig. 7(a) shows the logical formulas of the qualitative process for objects behaviors in water. When an object is within water, the event handler will detect this scenario and generate an event indicating this scenario. The event is then sent from the object to the QP engine. The QP engine will start the qualitative process and determine the influences on the object using some logical rules. Here, the precondition for the qualitative process is simply that there must be contact between the objects and water. The logical rules involve the density of the object, the density of water, the motion of the object and water, etc. In this case, the influences refer to the positional and rotational parameters of the object. Upon receiving the influences, the QP object will exhibit the corresponding behaviors.

Different objects will exhibit different behaviors in water. For example, heavy barrels will sink while light boxes will rise as demonstrated Fig. 7(b).

By combining the influences of object collision and water effect, we may easily model object collision in water as shown in Fig. 8. In *Twilight City*, we can clear see the damping effect of water on object's movements and collisions.

Situation: Objects In Water

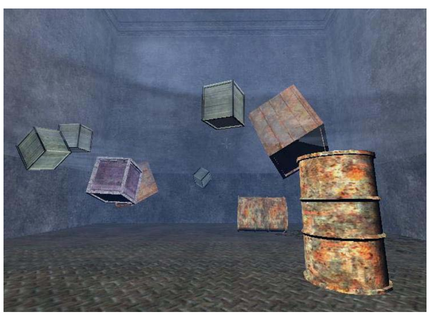
Entities Involved: Objects & Water

Preconditions: ?Contact(Water, Object B)

QP Logic: ?Density(Object A, Water),
?Water_Flow(Water)

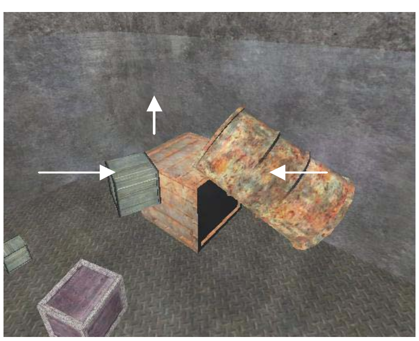
Influences: Object Position Parameters,
Object Rotation Parameters

(a) Logic formula

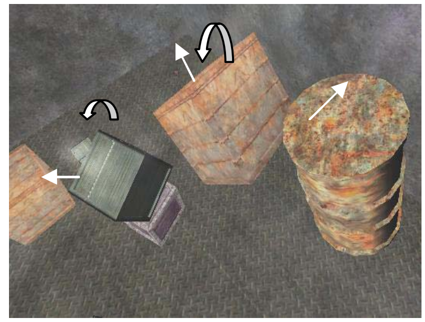


(b) Objects in water

Figure 7. Object behavior in water



(a) Before collision



(b) After collision

Figure 8. Object collision in water

5. Conclusions and Future Work

In this paper, we propose to adopt qualitative physics in virtual training environments. Qualitative physics has been applied in *Twilight City*, a virtual training environment for military operations on urbanized terrain, to model the behavior of various movable objects in the environment. With qualitative physics, the computational costs to represent the behavior of movable objects will be greatly reduced, thus more moveable objects could be introduced into the virtual environment to enhance the fidelity of the virtual environment and also to enrich a trainee's experience in the environment. As human beings are more sensitive to the qualitative and causal relations among various objects and factors, and rely on these relations in their reasoning about the environment, we believe that qualitative physics could also be very helpful in enhancing the non-player characters' spatial reasoning abilities so that they could have some *common sense* about the environment and the relationships among various objects and factors.

We will continue to work on the proposed qualitative simulation framework, and introduce more moveable objects into *Twilight City*. In addition, qualitative reasoning techniques will be developed to induce common sense and situation awareness to the non-play characters.

6. References

[1] Source: United Nations, *World Urbanization Prospects*, the 1999 Revision

[2] Bessa Machado, V., and Bredeweg, B., "Towards Interactive Tools for Constructing Qualitative Simulations", *the Fifteenth International Workshop on Qualitative Reasoning*, 17-19 May, 2001, San Antonio, Texas

[3] B. Bredeweg and K. Forbus, "Qualitative Modeling in Education", *AI Magazine*, Volume 24, Number 4, 2003, pages 35-46

[4] P. Salles, B. Bredeweg, and R. Winkels, "Deriving Explanations from Qualitative Models", *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, B. du Bouley and R. Mizoguchi (eds), 1997, pages 474-481, IOS-Press/Ohmsha, Japan, Osaka

[5] Forbus, K.D., "Qualitative process theory", *Artificial Intelligence*, volume 24, number 1-3, 1984, pages 85-168.

[6] Novak, J. D., and Gowin, D. B., *Learning How to Learn*, New York: Cambridge University Press, 1984

[7] Rickel, J., and Porter, B., "Automated Modeling of Complex Systems to Answer Prediction Questions", *Artificial Intelligence*, 93(1-2), 1997: pages 201-260.

[8] Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S., Kaminka, G.A., Schaffer, S., Sollitto, C., "GameBots: A 3D virtual world test bed for multiagent research", *in proceedings for the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS,* 2001, Montreal, Canada.

[9] Lewis M. & Jacobson J., "Game Engines in Scientific Research", *Communications of the ACM*, 45(1):27–31, January 2002.

[10] David Arendash, "The Unreal Editor as a Web 3D Authoring Environment", Proceedings of the ninth international conference on 3D Web technology. Pages 119-126. 2004

[11] Cavazza, M., Hartley, S., Lugrin, J.-L. and Le Bras, M., "Alternative Reality: Qualitative Physics for Digital Arts", *Proceedings of the 17th international workshop on Qualitative Reasoning*, 2003, Brasilia, Brasil

[12] Cavazza, M., Hartley, S., Lugrin, J.-L. and Le Bras, M., "Alternative Reality: A New Platform for Digital Arts", *ACM Symposium on Virtual Reality Software and Technology(VRST2003)*, pp. 100-108, Osaka, Japan, October 2003

[13] Syed, S. M., Pang, J. S., and Sharifuddin, Z., "Application of Qualitative Process Theory to Qualitative Simulation and Analysis of Inorganic Chemical Reaction", *the Sixteenth International Workshop on Qualitative Reasoning*, June 10-12, 2002, Barcelona, Spain

[14] Michael van Lent, McAlinden R., Brobst P., Barry G., Silverman, O'Brien K., and Jason C., "Enhancing the behaviorial fidelity of synthetic entities with human behavior models", *the 13th Conference on Behavior Representation in Modeling and Simulation (BRIMS 2004)*, May 17-20, 2004, Arlington, Virginia

[15] Z. Shen., S. Zhou, C. Y. Chin, and L. Luo, "Design of an AI Framework for MOUTbots", the *14th Conference on Behavior Representation In Modeling and Simulation (BRIMS 2005)*, May, 2005, Universal City, CA

Author Biographies

SUIPING ZHOU is currently an Assistant Professor in the School of Computer Engineering at Nanyang Technological University (Singapore). Previously, he was a Post-doctoral fellow at Weizmann Institute of Science (Israel). He received his B.Eng., M.Eng. and Ph.D in Electrical Engineering from Beijing University of Aeronautics and Astronautics (P.R. China). His current research interests include: distributed interactive applications, parallel/distributed systems, and human behaviour representation for virtual training systems.

SHANG-PING TING is currently an Engineer in the Simulation and Wargaming Solutions Centre at the Defence Science and Technology Agency (Singapore). He received his B.Eng. in Computer Engineering from Nanyang Technological University (Singapore). He is currently reading his M.Eng in Nanyang Technological University (Singapore). His current research interests include: human behavior representation for virtual training systems, terrain reasoning and military operations on urban terrain.