Physically-Based Modelling and Simulation of Track-based Main Battle Tank System for a realistic 3D Game

Yodthong Rodkaew
Computer Animation
School of Science and Technology
Bangkok, Thailand
yodthong_rod@utcc.ac.th

***Abstract*— This research aims for 3D modelling and simulation of the track-based in Main Battle Tank System (MBT). The MBT is based on the M60 Patton model. We interested in exploring the challenges in simulating the tank motions. We use Open Dynamics Engine (ODE) for the physics SDK. The results are the tank motions of the weight transfer acceleration and deceleration and can be used for a simulation and a realistic 3D game.**

Keywords*— *Physically-based vehicle, Track-based vehicle, Tank Simulation, Vehicle Simulation, M60 Patton, 3D Tank, Open Dynamics Engine, OpenGL

I. INTRODUCTION

Over the past few years, the growing of the realistic game industry requirement has focused on the visual simulations, cool graphics shaders, details on trees and shrubs, grass physics, fluids and smoke. The simulations extend to physics of breaking walls, structures, and other things that can be broken by the players. The vehicle simulations are still the most important things to bring players move into the far distance in game or control them for surreal combats. There are many car simulations inside the racing games and tank simulations in the war games. However, there is few technically derives especially in the field of 3D games. In this research, we are aim for 3D modeling and simulation to motion a tank movement base on physics simulations.

In this paper we chosen M60A3 due to the availability of data specification and very well-known main battle tank (MBT). The M60 Patton is an American second generations MBT. It was introduced in December 1960 [1]. The development of the M60 Patton began in 1957 [2]. The first prototypes were built in 1958-1959. The M60, was the first Main Battle Tank (MBT), built in the United States. It was designed around a new concept of general-purpose medium tank, that had the firepower of a heavy tank and a weight and mobility of a medium tank. M60 was designed to counter the threat posed by the Soviet T-54 and T-55 medium tanks. More than 15,000 of M60 series tanks and variants were built by Chrysler for the US Army and export customers. In 1978 work began on the M60A3 variant which improve the armor protection and the fire control systems. The M60A3 has mass 49,583 Kg (49.5 tons); Length 6.946 meters; Length with gun forward 9.309 meters; Width 3.631 meters; Height 3.27 meters; The engine AVDS-1790-2 V12 air-cooled twin-turbo diesel engine with 560 kW (750 bhp); Suspension Torsion bar suspension; Speed on road 48 km/h. The tank requires four crew members for drive and operate.

This paper is organized as follows: Section II about the related works on physics vehicle and physics tanks simulator; Section III about the tools for physics simulation; Section IV about the setting up the modeling and simulation; Section V is the results; the conclusions are in the Section VI.

II. RELATED WORKS

A. Physics Vehicle

Bourge and Bywalec [3] provide the physics equations and technics on the car simulation for game developers. The detail equations focus on rigids body simulations. Palmer [4] shows the physical effects on gears and wheel torque, aerodynamic drag, rolling friction, how to computing acceleration and velocity, breaking, wheel traction, crash model.

Srisuchat [5] works on "Development of a car physics engine for games" for develop a car physics system engine for game. The simulation cars are based on aerodynamic drag, rolling resistance, gravity, traction. The frameworks are also considering in car cornering when the car is turning the wheels at high speed, this cause car drifting. The weight transfer which is an important movement effect when car change its acceleration. The car will nosedive when breaking and leans back when accelerating. The car body will roll when cornering. This improves the visual effects and realism to car simulator.

B. Physics Tank Maker

CHOBI [6] creates a physically-based tank maker. He provides a game templates on the Unity asset store. The package can create the main functions of a battle tank such as: Driving, Track and Wheel Manipulation, Suspensions, Turret, Aiming, Reticle, Firing, Muzzle Fire, Recoil, Recoil Brake, Bullet, Explosion, Damage to the Model and AI. The user reviews given 5/5 stars. The tanks are nice moving, shooting and damaging reactions, and have Physics response on the tank chassis and ground. However, the video of tank maker of tank's movement seem not implements the weight transfer effects.

C. Physically-based model on the vehicle simulation

The acceleration of the vehicle is determined by the net force from the Newton's second law:

$$a = \frac{F}{M} \tag{1}$$

a denoted the total acceleration ($m \cdot s^{-2}$), F denoted the total force (N or $kg \cdot m \cdot s^{-2}$), M denoted the total mass (kg), of the vehicle.

The vehicle velocity is determined by integrating the acceleration overtime:

$$v_{t+1} = v_t + a \cdot \Delta t \tag{2}$$

Hence Δt denoted the time difference (s), v_{t+1} denoted velocity change from v_t by change of acceleration in the fraction of time $a \cdot \Delta t$

978-1-7281-0719-6/19/$31.00 ©2019 IEEE

The vehicle's position is determined by integrating the velocity overtime:

$$p_{t+1} = p_t + v_t \cdot \Delta t \quad (3)$$

the Δt denoted the time difference, p_{t+1} denoted a new position at time *t+1* from p_t by $v_t \cdot \Delta t$

The engine torque T_{engine} (in unit $N \cdot m$ or $kg \cdot m^2 \cdot s^{-2}$) can be calculate by:

$$T_{engine} = 9.5488 \cdot \frac{P_{engine}}{S_{engine}} \quad (4)$$

The P_{engine} is the power of engine in kW (M60 Patton use AVDS-1790-2 V12 $\approx$ 560kW) hence, unit W or $kg \cdot m^2 \cdot s^{-3}$, S_{engine} is the engine speed in RPM.

The wheel torque (T_{wheel}) can be calculate by:

$$T_{wheel} = T_{engine} \cdot G \quad (5)$$

The T_{engine} denoted torque of engine, G denoted the gear and final drive ratio.

The force applied to the right angle by the distance from the center fulcrum is defined using the cross product of equation:

$$T_{wheel} = r \times F \quad (6\text{-}1)$$

Hence F denoted force at r distance from the fulcrum (wheel radius). Which can rewrite to:

$$T_{wheel} = r \cdot F \cdot sin\theta \quad (6\text{-}2)$$

Since the wheel force $F_\perp$ always perpendicularly to the position (i.e. $sin(90°) = 1$). We get:

$$T_{wheel} = r \cdot F_\perp \quad (6\text{-}3)$$

Then

$$F_\perp = \frac{T_{wheel}}{r} \quad (6\text{-}4)$$

For a stationary vehicle, the total weight can compute as:

$$W = mg \quad (7)$$

The W denoted total weight of the vehicle (N or $kg \cdot m \cdot s^{-2}$), the m denoted the mass of the vehicle (kg), and g denote the gravitational acceleration of the Earth ($m \cdot s^{-2}$).

Seperation of weight on the front (W_{front}) and rear wheel (W_{rear}) can calculate with:

$$W_{front} = \left(\frac{df}{df + dr}\right) \cdot W \quad (8\text{-}1)$$

$$W_{rear} = \left(\frac{dr}{dr + df}\right) \cdot W \quad (8\text{-}2)$$

Where df is the distance from the center of gravity (CG) to the front wheel, and dr is the distance from CG to the rear wheel (Fig.1 upper left).

The vehicle wheels connect with the suspension act like a spring model in physics (Fig.1 lower left).

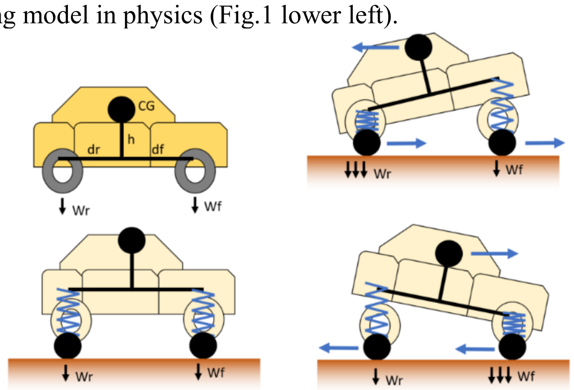

Figure 1. Diagram of weight transfer when vehicle change its acceleration.

When the vehicle acceleration, the wheels are moving forward this will take the CG going backward, due to the Newton's first law, results the front wheels have less weight and the rear wheels gain more weight (Fig.1 upper right).

When the vehicle deceleration, the wheels are moving slowly than vehicle this will take the CG going forward, due to the Newton's first law again, results the front wheels have more weight and the rear wheels have less weight (Fig.1 lower right).

III. Tools for Simulations

The tank model has many wheels to take the weight transfer formulations which need to resolve the rigid body and mass spring model. The physics model needs to take care of numerical error at the higher degree. After some preliminary researches and test the equations, we decide to use the physics engine that's capable to do rigid body constraints simulation and make a tank's movement with the real weight transfer effects.

A. Physics Engine

The physics engine is a computer software that do an approximate simulation of physical systems. It can calculate rigid and soft body dynamics, collision detection, and some physics engine can resolve fluid dynamics. The physics engine is widely use in computer graphics, simulations, video games and animations. In the work of Boeing and Braunl [7] evaluate the real-time physics simulation systems though Physics Abstraction Layer (PAL) and Open Physics Abstraction Layer (OPAL). There are many physics engines to compare: AGEIA PhysX/Novodex, Bullet, JigLib, Newton, Open Dynamics Engine, Tokamak and TrueAxis.

The constraint stability in the physics engine is the one of importance topics for vehicle simulation. If constraints are unstable by numerical errors, the constrained bodies can drift part apart. Based on the research, the ODE, Tokamak and Bullet are best at constraints solver. Noted that the ODE solver is the most accurate but slower. On the Material collision test shows two best engines that are ODE and Bullet. The test is the most importance in simulation tasks to response the collisions and results in accurate simulations. There are two candidates in our works ODE and Bullet.

Russ Smith [8] has developed the physics engine named "Open Dynamics Engine" or "ODE" in short. The ODE is an open source based on C/C++ API. ODE is useful for simulating vehicles, solid and soft objects. It supports the rigid body simulation and physical constraints. It is widely used in many computer games and simulations tools. The current version 0.5 is based on GNU and BSD license.

Erwin Coumans [9] the main author of "Bullet Physics", should be the largest group community on the physics engine, provide SDKs with well documents and examples. It is an open source based on C/C++ and a Python version "pybullet". Some of the implementations on the robot with deep reinforcement learning applications can be done in the simulated environments and can compared to the real world. It supports the rigid body simulation and physical constraints. The Bullet Physics is capable for Unity Game Engine components and has many of plugins for others software. It is optimizations [10] for the new GPU hardware: PlayStation 3

Cell SPU, CUDA and OpenCL. The current version 2.88 is subject to the terms of the ZLIB license.

The ODE physics engine is chosen. This is meet our requirements. First, the API function sets are small and easy to migrated on the project. Second, The light weight of the engine. Thus, its accurate, stable and mature for not frequent update.

B. *Graphics Engine*

The graphics engine is a computer software that do a graphics to display on the systems. It can calculate 2D and 3D mathematics in matrix and vectors form, import 2D and 3D files, and some graphics-related such as a camera system, an input from mouse and keyboard, collision detection, lighting and real time rendering.

The graphics engine is based on the graphics API, there are two major graphics API, Microsoft DirectX/Direct3D and OpenGL. We are choosing the OpenGL [11] for manage a real time graphics by its portability and stable for working on the Cross-Platform. OpenGL was introduction in 1992. It has become the industry's standard and most widely used for 2D and 3D graphics and can continue use the same source code without change, the API has supported all legacy functions hence no function deprecated.

In this paper we choose the graphics/game engine "ARKOMES" [12]. It is a light weight engine and open source, based on C/C++ and run on the OpenGL API. The engine was mainly developed and used for teaching the undergraduate and graduate students in computer graphics and related courses in four universities for six years.

IV. THE MODELLING AND SIMULATION

This section will discuss the tank model on setting up and its simulations.

A. *M60A3 structure*

The M60A3 consists of main parts which called tank chassis, tank track and a rotatable turret with main gun. The tank is designed to rotate main gun backward and forward by -180 to +180 degrees. The fig. 2 shows tank parts in details. We use a standard tank model from the internet. Since we aim for tank motions, the tank track is our most interested part.

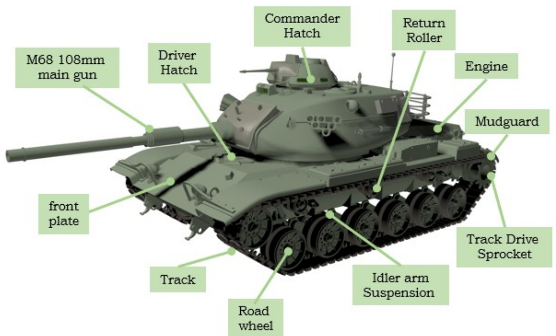

Figure 2. M60A3 Patton 3D Model.

The main rigid body of the tank is created by the ODE functions and then binding with the tank wheels. The binding use ODE's "hinge2" constraints. The "hinge2" is suitable for the wheel simulations as a build-in spring model and can simulate damping-like results. In the small diagram in the fig. 3 shows "hinge2" constraints which create for two axis joints perpendicular and binding the "body1" and "body2" together. The "body1" is the tank chassis, and the "body2" is the wheel. The track side will contain 11 wheels and "hinge2" 11 joints. The tank has total 22 wheels and 22 joints for track setup.

The ODE physics engine is calculating frame by frame for each wheel and resolving for rigid body of the tank. The active position and rotation of the tank movement is used for setting the three local vectors: Front, Side, Up vectors as in the fig. 3.

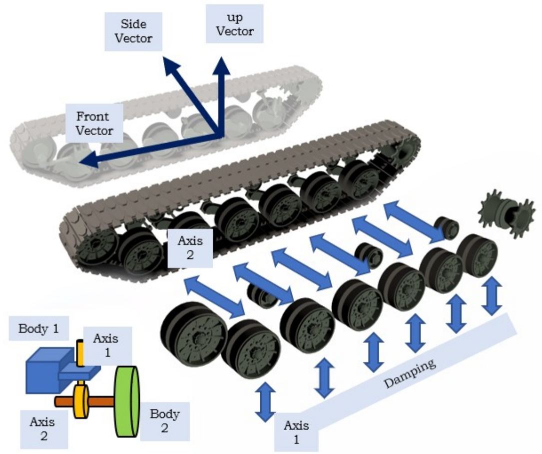

Figure 3. M60A3 Patton tracks compare to the physics model.

B. *Tank track*

Tank track or caterpillar track is a continuous track that treads of track are driven by two or more wheels. The track extends for large surface of the ground and distributes the vehicle weight better than the wheel. The tank has two tracks connected by two tank drive sprockets driven by the tank engine. The direction of the tank can drive both tracks forward, backward and separated direction.

On the physics simulator, we can't set sprockets motion and drag all their tank tracks along the wheel because they will do a very expensive calculation. By contrast, in all vehicle simulators, we usually do apply torque forces to each road wheel. The torque of road wheel and the road's friction will make the vehicle move forward.

The spline curve's control points are along with the wheel positions. The points are interpolated to the positions by same ratio distance. Each 3D parts of the track are place with the direction guiding by the vectors that calculate by the first derivative of curves (Fig.4). The control points attached to the wheel have a benefit. When the physics engine calculates the position of a wheel, the control points are follows in the wheel. This make a tank track look more realistic (Fig.5). After tested, we add the tank body to the main rigid body ODE.

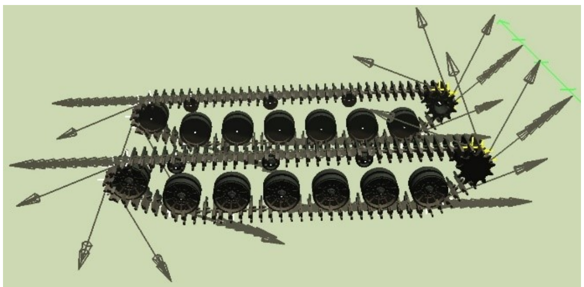

Figure 4. Tank track with the vectors.

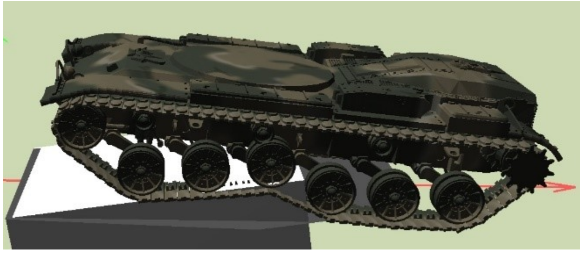

Figure 5. Test of the tank chassis in the physics simulator.

C. *Tank turret*

The tank turret is added on top of the tank chassis. The ball joint constraint connected two parts together. User can control turret to rotate around the axis of tank chassis (Fig.6).

D. *Terrain*

The ODE physics has support on the terrain data. The *Heightfield Class* can be used for height map terrains. There is a callback function from the physics engine to retrieving user's data. The AKOMES engine already has *Terrain Class* which can manipulate terrain data. The *Terrain Class* load data from two textures file: the height field data, and the texture terrain. After function modified, the data can pass through the ODE's callback function. In Fig.7 shows the terrain data and the results.

Figure 6. A complete physically-based tank model.

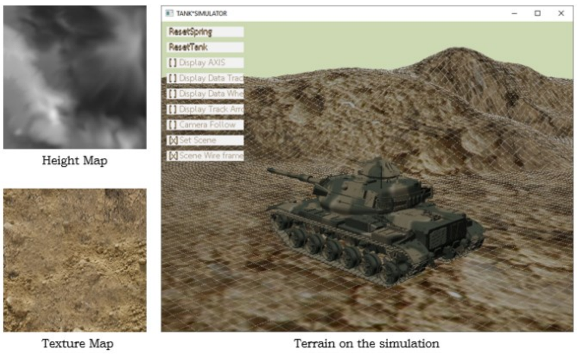

Figure 7. Terrain data and the simulation.

V. RESULTS AND DISCUSSIONS

The movement of the tank based on the physics model results works as we expected. The tank can accelerate and decelerate in which the way we want to. It can extend for a tank simulation in the real world. Although the physically-based modelling has some considering things:

A. *Physics stability*

When the tank run faster and faster, the tank's behavior may somehow unpredictably, may spin up and the front wheel lifted due to the Newton's first law.

B. *Wheel rotation madness*

This is the well-known problem in the physics simulation. The wheels appear to rotate off their axis constraints and had become visual bent. It is the numerical errors caused by the high rotation speed. We fixed it by omitting the rotation data from the physics engine. We display the wheel's rotation angle based on our calculation vector from the front vector.

C. *Euler Axis confusion*

OpenGL use axis by the right-hand based on Y-axis is up, and the plane lay on the X-Z. However, The ODE physics use axis by Z-axis is up, and the plane lay on X-Y. This made confusion when putting objects across the OpenGL's coordinate and ODE's coordinate. We solved this problem by a translate function between in and out OpenGL and ODE system.

D. *Tank track collision*

The tank tracks in this model do not collide to any rigid body. In the future, we wish there will be physics engine that have a specific method to handle it. We leave this case to the open problem.

The showcases are in the fig.8 and fig.9. The tank run over the obstacle shows the wheel suspension moving up and down. The terrain simulation shows the tank orientation and rotation lay by the slope of terrain. When the tank moving, the wheel suspensions do the great visual results.

VI. CONCLUSIONS

In this work, we created the physically-based modelling and simulations of the tank based on the M60A3 Patton MBT. The tank movement based on physics simulation. The acceleration and deceleration motion work as we expected. We uncovered problems as follows:

a) The real physics formulations are hard to implement which need to resolve the rigid body and mass spring model. The physics model needs to take care of numerical error at the higher degree.

b) Every physics engine has pros and cons, choosing the good physics engine may not the best case. Trying to choose the right one that suitable for the job constraint.

c) Physics stability and accurate simulations are the keys for success create the simulator.

d) The wheels can rotate off their axis constraints when the simulation runs with fast rotation. The physics engine is calculating by iterating physics formulations by the time frame. This may cause the cumulative numerical errors.

e) There is a very expensive calculation when calculates solution for the tank track simulator. This is the one challenge for whom that design the physics engine.

Although, the realistic level in 3D game can't compare to the real one. The contribution of this work is the method to construct the motion movement of the weight transfer. We create the new simulation framework for track-based vehicle. We hope in the future a new framework can be implemented and do physics fitting on another tank model and other track-based vehicles.

Acknowledgment

We have many thanks to the people who have many tanks at the Defence Technology Institute (public organization) (DTI), the 17th tank battalion king's guard and the ICT Silpakorn university for a site visit and on the real driving M60A3.

References

[1] ____, "M60_patton1", Wikipedia, https://en.wikipedia .org/ wiki/ M60_Patton

[2] ____, "M60_patton2", military-today, http://www.military-today .com/ tanks/m60_patton.htm

[3] D. M Bourge and B. Bywalec, "Physics for Game Developers : Science, Math, and Code for Realistic Effects", 2nd Ed. O'Reilly Media Inc, 2013.

[4] G. Palmer. "Physics for Game Programmers" New york: Springer, 2005.

[5] P. Srisuchat, "Development of a car physics engine for games", Master Thesis, faculty media communication, bournemouth, 2012, access online March 2019.

[6] CHOBI, "Physics Tank Maker", Unity Asset Store, https://assetstore.unity.com/packages/templates/systems/physics-tank-maker-50485, access onlin e March 2019.

[7] A. Boeing and T. Braunl, "Evaluation of real-time physics simulation systems", Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia 2007, Perth, Western Australia, 1-4 December 2007, 2007.

[8] R. Smith, "Open Dynamics Engine", https://www.ode.org/, access online March 2019.

[9] E. Coumans, "Bullet Physics", http://bulletphysics.org, https://pybullet.org/wordpress/, access online March 2019.

[10] ____, "Bullet_(software)", Wikipedia, https://en.wikipedia .org/ wiki/ Bullet_(software), access online March 2019.

[11] ____, "OpenGL", https://www.opengl.org/about/, access online March 2019.

[12] Y. Rodkaew, "The last eternity a 3D role-playing game with a cross-platform development", Computer Science and Engineering Conference (ICSEC), 2013 International, 2013, pp. 313-318.

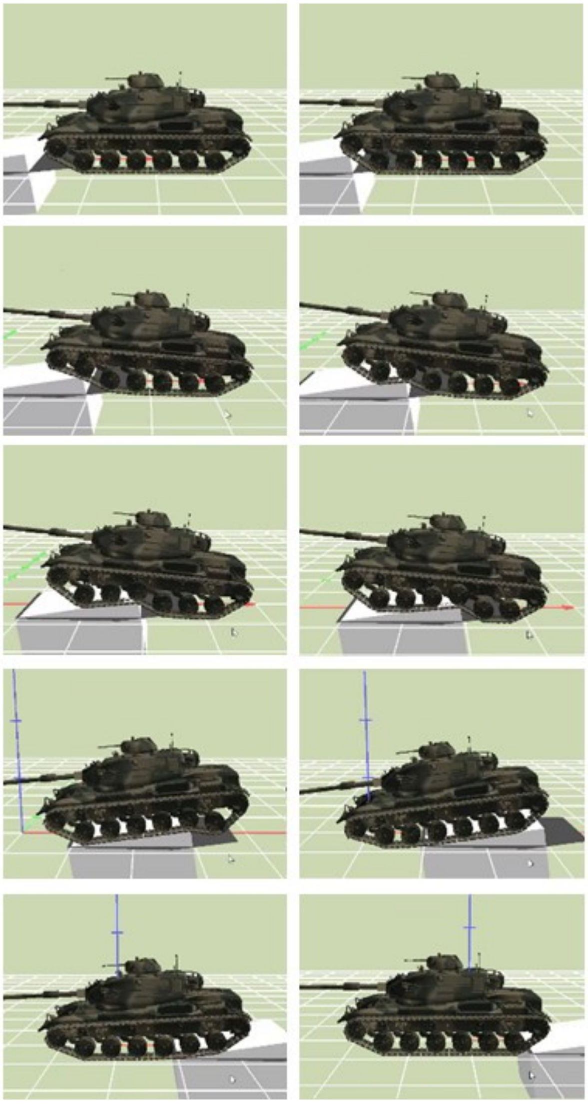

Figure 8. The tank runs over the obstacle.

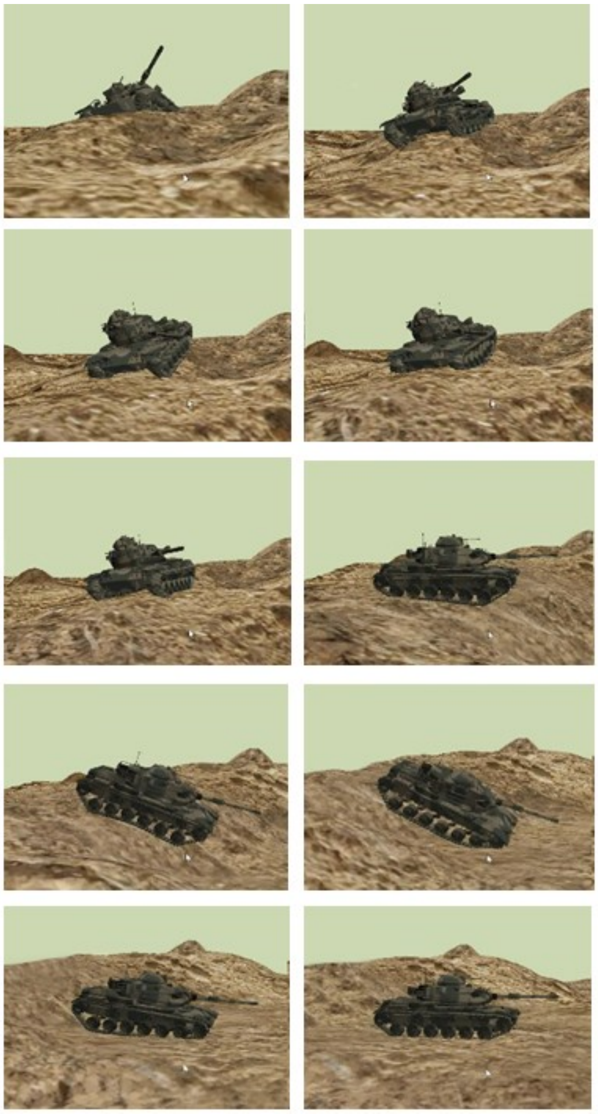

Figure 9. The tank runs over the terrain.