1 Introduction

Physics engine is computer software that provides an approximate simulation of physical system. E.g. ODE - Smith (2001-2007), Bullet Physics- Coumans (2003-2016), Box2D - Catto (2007-2015) are open source physics engines that are widely used in areas where fast solutions are required. Typical industries are games and film productions.

Computational methods used in physics engines are divided to modules that handle collision detection and modules that handle simulation. Simulation can further be subdivided into time control, motion solver, constraint solver and collision solver modules. Velocity-based formulation is used in constraint based rigid body simulation frameworks as collisions cannot be handled easily in acceleration based formulations. Friction is typically taken into account. Joints are handled by constraint equations. Detailed description of various components can be found in e.g. Erleben (2005). In this work focus is on solving new types of joints between rigid objects.

Plasticity is not typically taken into account in gaming solutions. Breaking of various objects typically takes place based on collision or impulse and breaking of steel or reinforced concrete structures does not look realistic. Theory for handling of plasticity has been presented already in Terzopoulos and Fleischer (1998). Müller et al. (2004) presents a method for modeling and animating of elastic and plastic objects in realtime using point based animation but it is not widely used.

Goal of this work is to find method that can be used in widely used physics engines to make simulation more realistic by connecting rigid objects by spring based constraints which have additional properties for handling plasticity. Main target is that plastic deformation takes place if force or moment exceeds given limit, deformation absorbs energy and joint breaks if plastic capacity is exeeced.

Methods presented in this work can be used in gaming industry to provide more realistic simulations without significant extra work. For gaming purposes presented method works best in scenarios where connected parts are relatively heavy. This allows normal integration timestep to be used without stability issues. This kind of metodology also opens quite large area of combining old structural analysis methods to modern simulation frameworks.

References

Bathe, K.J. (1982). *Finite Element Procedures in Engineering Analysis*. Prentice-Hall Inc. ISBN 0-13-317305-4.

Camp, C. (2015). *CIVL 1101 - Introduction to Strength of Materials*. url: http://www.ce.memphis.edu/1101/notes/concrete/section_1_strength_of_mat

Catto, E. (2007-2015). *Box2D A 2D Physics Engine for Games*. url: //box2d.org.

Cook, R.D. (1981). *Concepts and Application of Finite Element Analysis*. John Wiley & Sons. ISBN 0-471-03050-3.

Coumans, E. (2003-2016). *Bullet Physics Library*. url: //bulletphysics.org/.

Erleben, K. (2005). *Stable, Robust, and Versatile Multibody Dynamics Animation*. Ph.D. thesis. University of Copenhagen.

Gibson, S.F. and Mirtich, B. (1997). *A survey of deformable modeling in computer graphics*. Technical report. Citeseer.

Halme, T. (1995). *AGIFAP (Advanced Graphical Interactive Fram Analysis Program).*

Irving, G., Teran, J., and Fedkiw, R. (2004). Invertible finite elements for robust simulation of large deformation. In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 131–140.

Mills, N. and Mills, E. (2015). Taming the energy use of gaming computers. *Energy Efficiency*, pp. 1–18. ISSN 1570-646X, doi:10.1007/s12053-015-9371-1, url: http://dx.doi.org/10.1007/s12053-015-9371-1.

Moisio, S. (2013). *A Soft Contact Collision Method For Real-Time Simulation of Triangularized Geometries in Multibody Dynamics*. Ph.D. thesis. Lappeenranta University of Technology.

Müller, M., et al. (2004). Point based animation of elastic, plastic and melting objects. In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 141–151.

Nikula, S. and Halme, T. (1996a). *FINSTRIP, User's Manual.*

Nikula, S. and Halme, T. (1996b). *VTB-Program, User's Manual.*

O'brien, J.F., Bargteil, A.W., and Hodgins, J.K. (2002). Graphical modeling and animation of ductile fracture. In: *ACM transactions on graphics (TOG)*, vol. 21, 3, pp. 291–294.

O'Brien, J.F. and Hodgins, J.K. (1999). Graphical Modeling and Animation of Brittle Fracture. In: *Proceedings of ACM SIGGRAPH 1999*, pp. 137–146. ACM Press/Addison-Wesley Publishing Co. url: `http://graphics.cs.berkeley.edu/papers/Obrien-GMA-1999-08/`.

Oscar (2015). *TECHNICAL ACHIEVEMENT AWARDS (ACADEMY CERTIFICATES)*. url: `//www.oscars.org/news/21-scientific-and-technical-achievements-be-h`

Ricordel, P.M. (2005-2015). *Rigs of Rods Soft Body Physics Simulator*. url: `//rigsofrods.github.io`.

Smith, R. (2001-2007). *Open Dynamics Engine*. url: `//ode.org`.

Smith, R. (2002). *How to make new joints in ODE*. url: `//ode.org/joints.pdf`.

Terzopoulos, D. and Fleischer, K. (1998). Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *Computer Graphics*, pp. 269–278. doi:10.1145/54852.378522, url: `http://web.cs.ucla.edu/d̃t/papers/siggraph88/siggraph88.pdf`.

Timoshenko, S. (1940). *Strength of Materials*. D. Van Nostrand Company, inc.

University, C. (2015a). *Introduction to Computer Game Development*. url: `//www.cs.cornell.edu/courses/cs3152/`.

University, C. (2015b). *Physically Based Animation for Computer Graphics*. url: `//www.cs.cornell.edu/courses/cs5643`.