3D Blender Physics Simulation Documentation

Michael Cai[1] Nandana Weleweriya[2] Shameer Abdeen[3]

Columbia University[1] University of Georgia[2] Georgia State University[3]

October 2023

Table of Contents

Introduction

Welcome to a consolidated resource for Blender simulations used in the physics pre-lectures for UGA. This project was conducted with the help of Dr. Nandana Weliweriya, professor at University of Georgia, Shameer Abdeen, Professor at Georgia State University, and Michael Cai, undergraduate student at Columbia University. If there are any questions or concerns about the blender files or information presented, please contact him. If there are any issues or questions please contact the following:

Contact: Michael Cai

Email: mhc2167@columbia.edu.

Organization: The information presented will give a brief introduction to what blender is (most information can be found online) and the different simulations and how they were created for each one.

Blender is an open-source tool we've used to visualize complex physics concepts for undergraduate students at UGA. We use the physics modeling that Blender provides alongside other resources to create videos that simulate concepts.

Purpose: The goal of this project is to see how students learn with simulations given as 3D simulations and how they affect learning. This document provides direct links to the blender videos provided to undergraduate students that are presented by Professor Weliweriya shown here. As you navigate through, we hope you gain insightful perspectives on the synergy between technology and pedagogy, and the transformative potential it holds for modern education.

File Names and Topics

The files are ordered in such a way where the name of the file and the topic covered are ordered as such: [name of file]/[**topic covered**]

Name of File	Topic Covered
Velocity_car.blend	2D Velocity Vectors
Unifished1.blend	Block on Frictionless Inclined Plane
TireVectorAxis.blend	2D Displacement Vectors
Stickman.blend	1D Kinematics-Motion
Stickman_Displacement.blend	1D Displacement
Slide30.blend	Applied Force Perpendicular to Plane Inclined Plane with Friction
Slide25.blend	Applied Force Parallel to Plane Inclined Plane with Friction
Slide17.blend	Applied Force at Angle on Plane with No Friction
Slide13.blend	Applied Force Perpendicular to Plane Decreasing Normal Force
Slide9.blend	Applied Force Perpendicular to Plane Increasing Normal Force
Ramp_NoForce_NoFriction.blend	Block on Ramp Sliding
Ramp_NoForce_Friction.blend	Block on Ramp Sliding with Friction
No Friction Object.blend	Stationary Block Increasing in Mass
mgsin(theta)anglefinder1.blend	Derivation of Angle of Fg Perpendicular
Inelastic_Collision_.blend	Inelastic Collisions

Flat_Force_NoFriction.blend	F=ma on Block with Force Applied
Displacement_car.blend	2D Displacement Vectors
4_21_Eclipse.blend	Solar Eclipse
3_10_thirdVideo.blend	Two Blocks Connected by String Pulled Horizontally
3_10_fourthVideo.blendSimulation	Half Atwood Machine
3_9_templateFile.blend	Template for Human Body
3_9_secondVideo.blend	Contact Forces from Force Applied
3_9_firstVideo.blend	Newton's 3rd Law (Block and Human)
2D Components.blend	2D Velocity Components
2D Components_CarPOV.blend	Car POV with 2D Velocity Components
2_11_uniformCircular.blend	Uniform Circular Motion with Car
2_7_relativeVelocity.blend	Relative Velocities with Boat and Water
1_30_maxheight.blend	Max Height of Football and 2D Projectile Motion
1_30_Football.blend	Velocity Components as a Function of Time of Football
1_26_Animation3.blend	Velocity of Ball Thrown Vertically Upwards
1_21_Animation_2.blend	Velocity of Ball Dropped Vertically Downwards
1_21_Animation_1.blend	Velocity of Ball Dropped Vertically Downwards V2
1_6_1D_Components.blend	Velocity of Car in 1D Motion

Starting Off

System Requirements

Resource Cited: CGCookie

Operating Systems: Updated Windows, macOS, and Linux machines

Minimum Requirements:

1. 64-bit quad-core CPU with SSE2 Support
2. 8 GB RAM
3. Full HD Display
4. Mouse, TrackPad, or Pen & Tablet,
5. Graphics Card with 2GB RAM, OpenGL 4.3

Note: I used a laptop 16 GB RAM, i7 intel CORE, intel iRISx Graphics Card and was able to run small simulations. However, I do recommend using a device that can handle much more than what my HP Pavilion laptop could.

Optimal Hardware Recommendation:

1. 64-bit eight-core CPU
2. 32 GB RAM
3. 2560×1440 display as a separate monitor
4. Three-button mouse or pen+tablet
5. Graphics card with 8 GB RAM

This will be more than enough to run most if not all physics simulations. This additionally has added bonuses such as using different types of lighting, shading, reflections, and dealing with larger physics simulations.

Rendering Time:

In Blender, simulations must render and this takes time. For small simulations such as Slide9.blend could take around 30 minutes. However, larger simulations such as 2D Components_CarPOV.blend could take upwards of 3 hours. You can use cloud based rendering power:

1. Rentaflop
2. Blender Grid
3. Rebusfarm
4. Render Street

Using cloud based online rendering services is much faster but costs money.

Prior Knowledge

Prior Knowledge of Physics Concepts: Understanding basic physics concepts can enhance your grasp of the simulations. For instance, having a foundational understanding of kinematics, forces, or Newtonian principles can provide context to the simulations.

1. Linear Algebra
2. Kinematics
3. Forces
4. Momentum
5. Energy
6. Rotational Mechanics
7. Simple Harmonic Motion
8. Gravitation

Prior Knowledge of 3D Modeling: Familiarity with 3D modeling or animation software, even if not Blender specifically, can be advantageous. You'll likely recognize standard navigation controls, basic tools, and the process of manipulating objects in a 3D space.

1. Autodesk Maya
2. Autodesk 3ds Max
3. Cinema 4D
4. Fusion 360

5. TinkerCAD
6. SolidWorks
7. Revit

Prior Knowledge of Coding and Scripting: If the simulations employ any coding or scripting (for instance, using Python in Blender), having some foundational coding knowledge can be beneficial. It allows you to understand and possibly modify the scripts for custom simulations.

Installing blender

Mac: How To Download Blender For Windows 10 & Mac | Install Blender

Windows: How To Download Blender For Windows 10 & Mac | Install Blender

Linux: How to install Blender in Linux | 2023

Understanding Blender Basics:

Please watch the videos in the following order to understand a basic conceptual idea of what blender is and moving on into more useful and applicable concepts. **The video from 1 to 3 might be old and some concepts are different but it's important to just pay attention to the key movement and techniques presented.**

1. Navigation: Viewport Navigation - Blender 2.80 Fundamentals
2. Understanding the Workspace: Workspaces - Blender 2.80 Fundamentals
3. Understand the Interface: Interface Overview - Blender 2.80 Fundamentals
4. Donut Tutorial: Blender Beginner Tutorial - Part 1 **(Watch all Parts)**

After finishing the following videos. You should be more than ready to start making physics simulations in Blender

Important Techniques

1. Low Poly

Low poly is a polygon mesh in 3D computer graphics that has a relatively small number of polygons. I believe that using low poly helps in reducing the amount of time needed to create simulations but adds a sort of "gamish" feeling to the overall.

(Low Poly Setting)

In physics simulations, the benefits are:

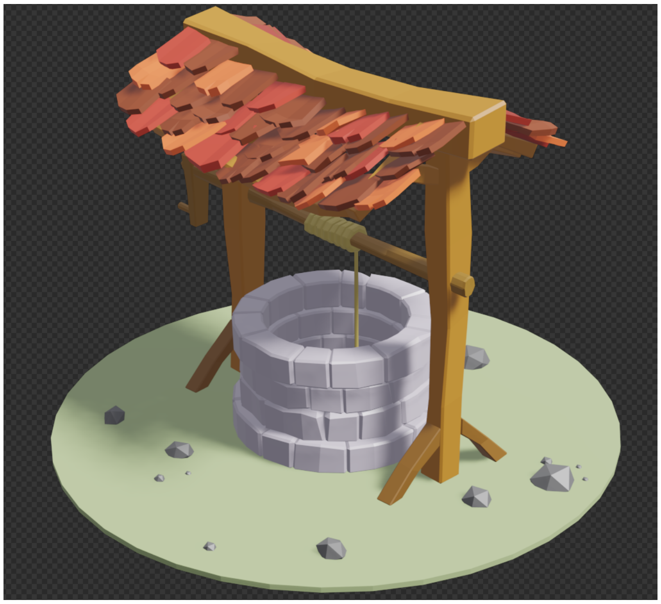
1. Reduction of Rendering Time
2. Reduction of Modeling Time

While there is a reduction of modeling and rendering time, the quality of the simulation is still maintained and the conceptual ideas are still maintained.

How to get started with Low Poly:

1. Low Poly Animals | Quick and Easy | Blender 2.8 | Basic Tutorial
2. Create A Low Poly Well | Beginners Tutorial | Blender 2.8 | Easy

These two tutorials can get you started on Low Poly and most of the concepts involved. They will help you make many of the future simulations while maintaining a good style.



(Low Poly Well I created)

2. Background

For Physics simulations, it's important to have a background that is bright and contrasts with the overall color of the other objects of interest in your simulation

For your background, choosing natural sunlight or complete darkness could be a good choice based on what you do.

There are free online resources for backgrounds
Free Background

You can also use other websites for your background or the simple lighting provided by Blender themselves.

3. Minimization and Optimization

If your computer is not as strong, there are ways to minimize the resources used to ensure that Blender simulations and renderings can be done.

Viewport Display: 10 ways to improve viewport performance and FPS in blender

- Simplify: Under the Render Properties tab, there's an option called Simplify. This allows you to reduce the subdivision levels, shadow samples, and other settings for faster viewport and render times.
- Limit Texture Size: In the Viewport Shading dropdown, limit the texture size to reduce the quality temporarily for better performance.
- Use Bounding Box or Wireframe Mode: Switching to these modes will reduce the load on your GPU.

Reduce Geometry: How To Reduce Polygon Count In Blender

- Decimate Modifier: Apply the Decimate modifier to reduce the number of vertices and polygons in your model without drastically altering its appearance.
- Avoid Subdivision: Minimize the use of subdivision surfaces or keep the subdivision levels as low as possible during modeling.

Optimize Simulations: How to Optimize a Model [Blender | Tutorials]

- Use Low-Res for Testing: When working with particle systems, fluid, or smoke simulations, use lower resolutions for test simulations. Increase the resolution only for the final render.
- Bake Simulations: This will pre-calculate the simulation, so you're not recalculating every time you play or render.

Rendering: EVERY way to SPEED up Cycles! Up to 1000% - Blender 3D

- Use Render Regions: If you want to check how a specific part of your render looks, use the render region option to render only that portion.
- Reduce Sample Rate: Lowering the number of samples can speed up render times but might reduce the image quality. Find a balance.
- Optimize Tile Size: Depending on whether you're using a CPU or GPU for rendering, adjusting the tile size can improve rendering speeds.

4. Physics Simulation

A great guide to physics simulations and different kinds are shown below:

Manual For Physics Simulation

You can look through the different techniques necessary and that you deem important to the physics simulation that you are doing.

Examples

1D Kinematics

2D Kinematics

Projectile Motion

Forces

Rotational Motion

Conclusion

In physics education, visual aids can make abstract ideas tangible and easier to engage students. This resource consolidates an array of Blender simulations tailored to undergraduate physics students. By creating complex animations in Blender, it helps benefit students and educators alike.

While Blender can be relatively challenging to work through, it has an uncanny ability to model physics concepts and is able to transform the pedagogy of future academic institutions.

I encourage future educators and others to use their own simulations or the simulations presented here in their own curriculum. This will inevitably help students succeed in the physics curriculum and understand concepts in a more thorough manner.

Feedback

Your feedback is invaluable to us. We are continuously striving to improve and update our resources to better serve the academic community. Whether you are a student, educator, or just an enthusiastic learner, we would love to hear about your experiences with our Blender simulations, suggestions for improvements, or any other insights you might have.

Please reach out to Michael Cai at mhc2167@columbia.edu with your feedback, questions, or comments.