Timothy Rolich

Lattice-Based object presentation script

Slide 1: Title

Hello there, my name is Timothy Rolich. I am here to present my idea for Lattice-based 3D objects and object interaction.

Slide 2: Mission Statement

The goal of this project, my passion, is to create tools for future generations of developers, designers, and creators.

Slide 3: Road Map

This presentation will be broken into six sections.

What it is. In this section I will discuss the problem I want to solve and my solution to this problem.

How it works. Here I will discuss how the tool will work, the different features that will be integral to the tool, and the specifics of what I plan to build.

Where it will be used. Here I will develop out some use cases and showcase the possibilities that I see for the finished tool.

How it will be done. Here I will explain how I plan on building the tool, the time I believe it will take for the major milestones, and the amount of money that I am requesting to build it.

Next, why it matters, where I will explain why I am passionate about this project, how it will support the community, and how it makes a good use of Unreal Engine.

Finally, I will answer some frequently asked questions and talk a bit about myself, specifically my qualifications and goals.

Let's start with the problem.

Slide 4: The Problem

Object interactions in games, simulations, and other 3D animations lack realism and take too long to create. Things like objects crashing into each other and breaking either have no sense of realism or take way too many resources to create in a realistic fashion.

One example of this could be a hammer smashing a rock. In most cases the rock will just disappear and in others it will break predictably and uniformly in a semi-realistic fashion. This won't be good enough for many games moving forward. This brings me to the solution.

Slide 5: The solution

My solution to these problems is to create objects that not only have form but mass as well. In this I mean lattice-based objects which contain numerous points and bonds within the object. By having these additional structures and altering the physics engine to handle them, lattice-based objects will fix the problems presented previously. Now we can see a little more about how this will work.

Slide 6: How it Works

There are 3 main parts to the tool. The first is the lattice. It is the overarching structure that would be comparable to 3-Dimensional shapes. There are a few key differences: lattices are made up not only of the points that make up the outer shell but molecules and bonds within. Lattices will have density, malleability, rigidity, mass, and volume. With these properties, lattices will be able to change shape, grow and shrink, and interact with other lattices.

The next are the molecules. Think of these as the points that make up triangles and quads in OpenGL but with a few additions. Firstly, molecules will make up tetrahedrons, pyramids, and cubes instead of the 2-Dimensional equivalent. Secondly each molecule will have its own mass and be able to interact with other molecules.

Finally, we have the bonds. The bonds are the connecting pieces between the molecules. Like real bonds, they will have a certain elasticity and strength. They will also have different values based on the molecules they connect. These properties allow the bonds to not only break and reform but to also change in length, allowing the lattice to change shape. In future iterations bonds may hold energy allowing for more reactions when they break i.e. light or sound generation.

Next, we will examine some of the core features required of these objects.

Slide 7: Features

The three main features of lattice objects are: being made up of different molecular structure, breaking due to collisions, and detailed detection for collisions. We'll start with the different molecular structures.

Slide 8: Different Molecular Structures

As I alluded to before, the lattices are made up of molecules and bonds. In nature, these typically take the form of different crystalline structures which connect in numerous different ways depending on the molecules, the surrounding temperature, and the pressure on the object. This will be mimicked in the lattice-based objects. These different structures will allow for more unique reactions and essentially help to make collisions and breaks more realistic based on the type of materials interacting. Next we'll visit breaks.

Slide 9: Collision Breaks

The most important feature, the driving idea behind this tool, is breaking objects based on collisions. With the use of the molecules and bonds, objects are able to take on a whole new form. When objects collide in reality, they affect each other, not just in the force they put on one another but in an intricate way, they break, they crack, they scratch, and they dent. To get a better idea of how this will work, imagine small spheres that are connected to each other with lines. If all of the lines that connect one half of an object to another are broken, then the object becomes two. The spheres on either side of the lines that broke become the new points for the faces of these two objects. The same can be done with smaller things such as scratches and dents. Next, I'll speak about detailed detections.

Slide 10: Collision Detection

As any gameplay programmer will tell you, collision detection is difficult and frustrating. With these objects, many things will be made much easier and more robust. By having many different points within an object, detailed detection is possible. This detailed detection will allow for uniquely programmed behaviors for different sections of an object. Imagine a trigger within an object, where the player must hit a very specific spot on an object, possibly inside the object. These objects will let the programmer decide that functionality and the location that must be hit. This is an open-ended feature intended to make the life of the gameplay programmer easier. Next, I'll talk through a couple different use cases.

Slide 11: Use Cases

I can imagine three main use cases for this tool: video games, simulation, and 3D animation. I will walk through instances of video game and simulation use cases next.

Slide 12: Video Game Scenario

In this scenario I ask that you imagine an RPG. In this game, the player has the ability to cut down trees. There are many instances of this gameplay across dozens of games, but not many of them take into account the effects on the axe used by the player. Think about those that do. What happens to this axe? If you are in some games your axe will lose some arbitrary amount of hit points for each use. You may even see some wear and tear on the axe, but it is scripted and happens the exact same way for every axe and for every use and misuse of the axe. If you are in most games, nothing happens to the axe, it never wears down, it never breaks.

By using lattice-based objects for the axe and the objects which the axe can hit, such as trees, rocks, houses, etc., we can show specific effects on the axe. With constant use, the blade may dull, rendering it useless. By misusing the axe, it may dull quickly, crack, or even break. This extra layer of realism helps players to really immerse themselves in the game. Now we will explore a simulation scenario.

Slide 13: Simulation Scenario

For the simulation scenario, imagine testing a new engine for an airplane. The main purpose of the simulation would to be ensure all things function as expected for different adverse circumstances. For this scenario, imagine testing the plane in flight and something gets sucked into the engine. Current technology would not be sufficient for three reasons. Currently all possible adverse scenarios would need to be coded, at least in part, by hand, which takes too much time and leaves room for more mistakes. Even after all of the time spent coding these scenarios, it isn't possible to think of all the possible problems. These problems stem from the fact that object interactions currently cannot mimic reality in a substantial way. Lattice-based objects will help to mitigate these concerns allowing for a much more realistic interaction between objects. Things like chain reactions, high energy transfers, wear over time, and more are impossible with current technology, but simple with lattice-based objects.

Now that I've spoken about how it will be built and some possible uses for it, I will explain the basic timeline for the major milestones of production.

Slide 14: Milestones

The first milestone is to begin working. While I have experience with different game engines, I haven't had enough time with Unreal, especially concerning tool building. The first milestone will consist of learning everything I can about Unreal Engine: the inner workings, the available tools, and the overall structure. It will also consist of planning the specific class hierarchies I will use to create the tool. My best approximation for this is about 100 hours but I imagine it may take anywhere from 50 to 200 hours depending on the complexity required.

The second milestone is the construction of the main classes. The bulk of this milestone will be actually working with Unreal Engine and building the infrastructure of the main classes. Once this milestone is complete, the user will be able to create objects with the new structure and transform other objects into lattice objects. I am hoping to be able to read in objects from as many sources as Unreal Engine will allow and transform them into lattice objects. My approximation for this milestone is about 300 hours.

The third milestone is lattice interaction. This is the logic that allows lattice objects to collide, break, and generally affect each other. This milestone will also include changes in the physics engine that need to take place for the objects to behave appropriately. Once this milestone is completed the user will be able to create objects and let them interact with themselves and their surroundings. My approximation for this section as about 400 hours.

The fourth milestone is one that I believe too many people don't plan for in their projects, efficiency updates. The main purpose of this milestone is to refactor everything done previously into a more efficient and effective codebase. When this milestone is complete the user should be able to create more objects, create them faster, and use less resources. For this milestone I will be limiting myself to 100 hours because there can always be improvements, but the project needs to get done.

The fifth milestone, and the one that I believe will be the most difficult, will be shaders and textures for the objects. The main purpose of this is to find the most efficient way to reorganize the faces of an object when it gets broken down. Once this milestone is complete the project will be near completion. Objects will be creatable, they will be able to interact with each other and their surroundings, and they will be visually accurate to reflect the different changes they undergo. This will be the culmination of the main part of the project and I approximate around 500 hours to get everything working correctly.

The final milestone is that of user testing. In this section, I will allow outside users to test the functionality and usefulness of the tool. I will be taking criticisms and attempt to flush out as many problems as I can. While this is the main testing for the project, each milestone will also

have its own testing section in which I, and anyone else I have helping with the project, will run the code through as many scenarios as possible to squish bugs. I approximate this final section at 100 hours, though I'm sure this is an underestimate. Overall, I approximate my time at about 1500 hours of work. With a simple break down of 10 hours a week, I believe it will take just under 3 years for me to finish this project.

So far we've talked about what the tool is, how it will work, use cases for it, and how long it will take to create, but now I will talk about why it matters to me, and to the Unreal and game development community.

Slide 15: Why Does it Matter?

In this section I will discuss 4 topics concerning the tool: why it makes a great use of and asset to Unreal Engine, how it will support the community, how it will leave room for lots of expansion, and finally why I am personally invested in this idea. Let's start with Unreal Engine.

Slide 16: Use of Unreal Engine

As a game engine, Unreal is known for its super high-quality graphics and realism. With this tool, Unreal will be getting something that pushes it further in this direction, allowing for even more realism within the engine.

It is also known for its incredible efficiency. In this regard, I believe my tool will work solely on Unreal Engine, truly pushing the engine to its potential.

All in all, this tool will help to make Unreal Engine even more powerful, robust, and useful. It will also provide a great amount of support for the designing community.

Slide 17: Supporting the Community

With the help of an Epic Megagrant, I will be able to provide a powerful and robust tool for the community that is both free and open-sourced. This tool will aid in the furthering of graphics and physics engine capabilities, paving the way for even more innovation. It will also leave room for tons of expansion and extensions.

Slide 18: Expandable and Extensible

A couple of ways that I see this tool being able to expand are with different states of matter (liquid and gas), and new side effects, such as different materials interacting and forming new materials or even exploding. There are also many ways that I see this tool being extended, especially with respect to more materials and more crystalline structures. There is also room for more definition for specific use cases beyond what I will create. This is a tool I love thinking about and something that I would be proud to work on. That is not to say I won't have other benefits. Next, I will describe what I plan on getting out of this tool, beyond building something great.

Slide 19: What's in it for Me?

I have broken this down into personal and professional goals.

My personal goals. Something I've wanted for a long time is to learn about all of the intricacies of Unreal Engine. The inner workings and code structure intrigue me and this gives me a good excuse to sit down and actually put in the time to learn it. Another personal goal is to get name recognition for the work that I've done. I don't want to be a household name, but it would be nice to be appreciated for creating something useful. Finally, the experience in and of itself is a major goal of mine. I am passionate about video games from all angles and being able to work in an area that I hold dear is an amazing advantage.

My professional goals. Once this project is complete I hope that I will be able to continue work on more free and open-source tools and software. I believe that most software can only be truly appreciated if it is used by as many people as want to use it. Another professional goal I have is to gain work experience in my preferred field, game engine and tool design. I hope that my experience in building this tool helps to qualify me for potential jobs working with game engines. My final professional goal is something I hinted at before, paving the way for others to continue to build new and exciting tools. If my tool helps to advance even a small area of game engine design, graphics engines, or physics engines, I would be ecstatic.

Next I will lay out some other important information.

Slide 20: Other Important Information

There are 3 main points that I haven't yet spoken about: time and money requirements, my personal qualifications, and other questions you may have. Let's start with the most awkward part first, time and money.

Slide 21: Time and Money

As I previously discussed, I believe at 10 hours a week it will take around 3 years to complete this project. My approximations are sure to be off as I haven't done a project of this size yet, but I believe they are fairly accurate.

In my calculations, I believe that $50,000 over three years is a reasonable request. I have broken my request into three categories. First, a new computer system on which I can run Unreal at a decent rate and be able to create the tools would be around $3,000. Next, office space to both work at and store the new computer. This section has the most variability because it depends solely on how long it takes to create the tool, or how long it takes me to move into a larger house where I can have a room dedicated as an office. For 3 years, this comes to about $9,000. Finally, I need to be able to sustain myself and take care of my family. For every hour I spend on this project I am not spending it making money at my job or taking care of things at home. Due to this, I am requesting the largest sum of money for my time spent on the project. At 10 hours a week, at about $25 an hour, it comes to approximately $38,000. These figures sum up to $50,000. I think this is a reasonable request for the potential prospects of this tool, but I understand if it is asking too much and would appreciate any amount of money to work on this tool. However, the less funding I receive, the more likely I will need to do something more to monetize the tool and make it less available to the community. Next, I will speak more about myself and why I would be a great person to be taking on this project.

Slide 22: My Qualifications

My first and foremost qualification is my Bachelor's of Science in Computer Science from the University of Wyoming. I spent years of my life focused on learning everything I could about computer science and programming. In my degree, I have taken classes that advanced my knowledge in many areas, especially those pertaining to game design. Some of these classes were Computer Graphics, Scientific Computing, and Software Design. In my classes I have worked with many languages including Python, Java, and C#, but none more than C++, the main language used with Unreal.

This brings me to my next point. I have been using C++ for over 5 years in numerous different scenarios, from building a game engine with the SFML (Simple and Fast Multimedia Library), to creating basic security tools, and much more.

Finally, while this last point isn't exactly work or school experience, I believe it is some of the most significant experience I have. I have been playing video games for as long as I can remember, and I always critique the gameplay. I like to look beyond what is built and ask how things could be better, more fun, or more efficient. With a lifetime of experience doing this, I

have come up with many ideas for how to make things better and I will use these to help develop my tool. The last section I have to speak about is a few questions that didn't fit well within the rest of the presentation.

Slide 23: FAQs

The first question is: "Can't this behavior be created at a higher level when desired?". The answer is yes. But there are many issues with trying to build complex behavior on top of an already complex engine. Some of these issues include poor efficiency, overuse of resources, and complexity for the user. By building it as the base layer, these issues can be circumvented.

The second question is: "Why is now the time to do this?". There hasn't been a better time to create this tool. The technology available, specifically graphics cards and processors, is finally able to handle the intense calculations required by the tool.

Slide 24: Usability

The third question is: "How exact or realistic will the interactions be?". There are tons of different contexts that these objects will be in. Considering the size of the objects, the resources available, efficiency requirement and more, there must be different levels of precision. For this, I plan to implement a significant amount of dynamic editability. This will allow for the user to edit both the number of molecules in an object and the distance between the molecules. I may also include an option to allow less dense molecules towards the center of an object and more dense molecules at the edges. Finally, I have a few questions for the audience and a few closing remarks.

Slide 25: Appendix

Once this tool is complete, where else would you like to see it implemented, and what should be the next expansion to the tool?

Lastly, if there are any other questions or recommendations, please contact me by email or phone. Thank you so much for your consideration. I am truly honored to apply for a Megagrant and I hope to hear from you soon.