PROFESSOR: Hey, everybody. Welcome to 6.849, Geometric Folding Algorithms. I am Erik Demaine. You can call me Erik. And we have as TA Jayson Lynch, who's right there. And this class is a bit unusual, at least for me, because I'm trying for the first time a new experiment, which is inverted lecturing. And I wrote this on the poster for the class, and everyone started asking me, what's inverted lectures?

Well, it's not a new idea, but I've never tried it before. The concept is to make these in-class times, where we're all here together, more interactive by taking the lecture component of the class, which is covering all the material into videos that you watch online. So this class is basically going to alternate between real in-person things, as you are here-- of course, we are also being video recorded, so slightly a contradiction in terms-- but that's for the people in the interwebs to be able to watch this. So we'll alternate between the real in-class part and video lectures from the last time I taught this class, which was fall 2010.

So this is actually what the course web page looks like as of an hour ago. So we've got these-- L01, L02, L03-- those are lectures. That's content, lots of material packed in there, which is how I'm used to teaching a class-- put as much material as I can for a lecture. Those are already available. You can start watching those videos.

Lecture one is optional. I'm going to basically cover a shorter version of that today. It says right there, optional. Then the Cs, which don't exist yet, because we're doing it right now. The Cs are the in-class sessions, the contact hours. Some of those may be recorded, like this one, we're going to record. But some will not be, and you need to attend both. So you've got to watch the videos, and you've got attend all the classes. That's the set up.

So this is a bit unusual, so I'll spend a few more minutes on how this class is formatted. So-- requirements. We've got-- watch video lectures. I'll talk a little bit about timing of that. And then we've got attend classes. And then to the actual material, a regular part of the class that you'll be graded on consists of two parts: problem sets and a project. The project also has a presentation. This is a project focus class, so the bulk of your grade is based on the project and presentation.

Project can take on many different forms. This is a theoretical computer science class, so there's the typical kinds of projects, like, to read a bunch of papers and summarize them, survey them. Though we ask that you keep them disjoint, more or less-- avoid stuff that is well covered in this book, *Geometric Folding Algorithms,* or in the class itself. But otherwise you can survey material.

In addition to the standard survey paper, or instead of a standard survey paper, you can write a bunch of Wikipedia articles about folding stuff. So where survey stuff should be new material that you haven't seen, Wikipedia stuff could be material that you've seen in this class but is not well covered in Wikipedia. And that will help take over the world, our usual goal.

If you see a cool algorithm, in this class you could implement it if you're a coder and see how it actually works, make it demoable, which will be fun. Or you can work on an open problem. So if there's interest, we'll run an optional open problem-solving session, where we'll all get together and try to solve open problems. I might also do them in these class times, because now we have a lot of freedom to be more attractive, to actually fold paper or to have fun in class.

And instead of me trying to pack as much material as I can-- because that's already been done in the videos-- you could pose an open problem. Especially if you come from another field, you may have some interesting folding-related problems. Those could be really cool. Or you could build something, a physical sculpture, physical structure of some kind, furniture, architecture, whatever. Or you could do it in a virtual world and just design something interesting. Though there you might want to do more than one, make it harder for yourself.

All of these are possible projects. There's a lot of different options for different types of people, different backgrounds, whatever. Then we'll also have problem sets, which are weekly, roughly, and they shouldn't be too long. And they will also have an option of not doing all the problems. So every problem set will probably have the rule that you can drop, we will drop the lowest grade on one of the problems that's on the problem set.

So that means if you just don't want to do a problem, you can skip it and your grade will be determined by the others. If you want you could try to do all of them, and we'll just drop the lowest grade. So if there's some problems are easier for you, harder for others, you can mix and match. Cool.

There's a lot more details about this stuff on the website. If you click on Project, you'll get lots of details about these different styles of project implementation-- survey, open problems, whatever. Because what I said was pretty brief. So look at that for more details. Project, you don't have to worry about right away, but one of the luxuries of having video lectures is you could skip ahead a little bit to see what looks interesting to you and work on a project related to that.

I wanted to briefly show you a little bit about this website and how it works, so you get a flavor for it. So let's say the very next thing you should do, right after you leave here, sometime between now and noon on Monday, you should watch lecture two. So you can just click on lecture two here on the left. And this is the way it's set up. You could zoom to fit everything on your screen. On the upper left you've, got your video. And this should start playing.

[VIDEO PLAYBACK]

-All right, welcome back to 6.849.

[END VIDEO PLAYBACK]

PROFESSOR: Hey it's me. I'm even wearing the right t-shirt. I had a coincidence. The fun thing about these videos, as a few nice features. One is as you jump around in the video, you see the slide down here in the lower left updates to whatever I'm covering. And also if I jump farther, the page handwritten notes that I'm covering on the right changes. So as I scrabble around, I think is the term-- too many buttons. Who designed this website? All right, I forgot this. Reload. So to jump around. I'll turn down the volume.

You can also speed up the video. This can be pretty entertaining if you go really fast. But you can go 1.1, 1.2, that's pretty comfortable. 1.5. Yesterday I listened to a video to at 2x. it's interesting.

[VIDEO PLAYBACK]

-In general, when I make that fold, it might come out of here. I'd have to wrap around that [? cone ?], sometimes.

[END VIDEO PLAYBACK]

PROFESSOR: Thankfully, I do not sound like a Chipmunk. But if you haven't seen lecture before, it's a little harder. But I'll be watching the lectures with you, experiencing the same total number of hours you have to experience, So we're in this together. Let me know if you have any comments on the website. It's all been done by hand, so if there's anything you want changed, it's easy to change.

You can also do fun things. Like when it's paused, you can jump around in pages on the right and say, oh, yeah, I didn't really understand that page. And you click on this time, and it will start playing from that time. At full speed. We could slow it down, too, which can be pretty entertaining. Anyway, you get the idea.

Let me know how it goes. I want to make it as easy as possible. In fact, if you don't like my software, you can just skip it all, download this video, like 720p version or the 360p version. Play it in VLC with whatever fancy speed up, speed downs, you want to use, or put on your iPad or other tablet, whatever. This page should work on an iPad, but I haven't tested it lately. If it doesn't, let me know. Cool. So that's what it looks like.

And then at the top of this page, you see here a link to completion form. When you finish watching the video, you click on this form. It's pretty minimal. You enter basically your name, your username or your email address, you say yes, I watched the video, and then that's all that's required, you click Submit. But we highly encourage you to say what you think at this point.

If you have any questions about the lecture, that wasn't clear, or anything you didn't understand. Or I a briefly mention something that sounded cool to you, and you want to know more about it. This is your chance to influence what I cover in class, and this is why you have to submit this form by noon the previous day.

So watching video lectures and filling out that form is due by noon on Mondays and Wednesdays, the day before class. That will give me 22 hours to prepare class, which should be enough, we'll find out. And so I can adapt the in-class time to be whatever people care about most. So it's kind of like a poll. If lots of people say I didn't understand x, I will cover x. If lots of people are curious about y, I will cover y.

So whatever you want to know about, typically related things, but if you have unrelated things you want to bring up as well, feel free to put it in there. And we'll try to schedule it in when it's time or when it fits. Yeah, that's the plan. I've never done this before, so if you don't like the form, tell me. Anything you don't like or think should change, just send email. Or put it in the form, but email's good, too. Question?

AUDIENCE: I noticed you have a "no" option. Is that for, like, oh, shoot, it's 11:45 on Monday, and I haven't watched this. Should I still submit the form and say, "no?"

PROFESSOR: Interesting. You're so honest. Why would you fill out "no" on this form? I guess so. You can fill out the form multiple times. So you could you could say no and give your excuse here, if you want it. I'll watch it tomorrow, maybe, and then later on fill it in with yes. So in particular, if you also have questions, like you watched the video early on and then later on you have a question about it, if it's still before class time, feel free to fill this out again. It just makes another entry in our table. This is a Google form.

Another question? I guess that's the reason for no. Maybe also we couldn't put a single option. No, it's to force you to claim, honestly. This all honor system, right, so this is like forcing you to make a decision to be dishonest, I guess. Or to be honest, rather. Cool.

To make this work, you've got to ask questions, you've got a request topics. Otherwise I'll just fill your class times with more material. That would be my natural temptation, so you've got to hold me back and ask lots of questions. One particular request I wanted to highlight is to send me cool things to cover. In particular, you all surf the web. And if you see cool folding things, or any kind of folding-related thing, on the web, send it to me. By email. And the plan is to do one a day, or maybe two a day, for in class time.

There's lots of fun things out there, and it's hard to see it all. So when you discover things, tell me and I will schedule it in throughout the semester. You can think of this as a standing homework problem. You should do it sometime before the end of semester. Another experiment. I think to be fun. That is the end of the class organizationally. Unless there are any questions about requirements or style or format.

Then in the rest of this class, I'm going to cover, as I said, a short version of lecture one, which is just an overview of the whole class. So I want to give you a flavor of what the course is about, what kind of theorems we prove, what kind of algorithms we develop, so you can decide whether you are interested in this class, whether to take it. There is also a survey that, where you put your name, email address, will add you to the mailing list.

And I announce every lecture, so if you're listening to the class, you just want to listen to things you're interested in, come to class for interesting things, just listen to that email and see when cool things come up. And also there's a little survey on that piece of paper. Does anyone not have the piece of paper? Couple people, so we will get them to you. A little survey just to get your background.

There's no required background for this class, because lots of people come from different areas. Hopefully, you know at least one of the background areas, but anything can kind of be filled in. And one of the points of this questionnaire is if there's something you don't understand, like, oh, gosh I should know XYZ algorithm, and you assumed it in lecture, but I don't know it, could you cover it? And if enough people request that, I will. So that's the format.

Out of curiosity, how many people here are a course six? And how many people are course 18? And how many people are other? OK. So reasonably balanced. A lot of 6, though.

So we are talking about geometric folding algorithms. So there's geometry, there's folding, and there's algorithms. Algorithms are how we compute things at least theoretically, and in general what this field is about is both mathematics and computer science. Say, the mathematics and algorithms that underlie how things fold. And the technical term for things is geometric objects. And we're interested not only in how they fold, but also how they unfold.

In general, you have some geometric structure they can reconfigure, like, my arm here can reconfigure in interesting ways. We want to know-- what are all the ways they can reconfigure, what all the way it could fold? This piece of paper is another geometric object. We want to know what all the ways that it can fold.

Before I talk about more formally what this means, I want to show you that folding comes up everywhere, pretty much. Almost any discipline you can name, there's some, if it involves physical things at least, has geometric objects. Geometric objects tend to comply. Even if you're working with completely rigid objects, you don't want them to fold. That's a folding problem.

So I have here a list of a bunch of different application areas, and I'm going to show you some pictures and videos related to them. So the first one is robotics. So if you have things like a robotic arm, it starts in one configuration. You want to continuously move it to reach some other configuration, so it can pick up somebody, drop it off over here. I want to know, how should I plan the motion of my robotic arm?

A different kind of robotic application is to make sheets of material that are themselves robots that told themselves into origami. This is a self folding sheet developed by collaboration with MIT and Harvard. And you just send a little bit of electrical current. These little metal pieces heat up, which causes them to pull the creases shut, and, boom, you get your origami boat-- with no origamist required.

The same sheet can fold into many different shapes, and the underlying algorithm here, or the underlying mathematics, is that this pattern of creases, square grid with altering diagonals, can fold into essentially any shape that is made out of cubes. Here we're making a paper plane out of the same sheet, sending it different signals. It does not fly but probably floats. So that's the idea.

And you can imagine a sheet like this could just be arbitrarily reprogrammed to fold into this gadget, so I don't have to bring around so many gadgets. Maybe my laptop could fold smaller, and later could unfold in something to my desktop. And I don't know, my phone could reconfigure into something else. That's the vision. This is what we call programmable matter, where you can change shape. In the same way that we program software, we want to be able to program shape. Obviously, we're just getting there now. So that's robotics.

Next up is graphics. One example is you want to animate a character from one position to another. That's a key frame animation that's a folding problem. I have here an example of the two dimensional analog, where you have one polygon and another polygon, and you want to morph continuously from one to the other.

And these animations are all found by algorithms that are motivated by folding stuff that we will cover in this class. And that's not so easy, but it's lots of fun things. All of these motions avoid collision, and they approximately preserve the edge lengths, also, if it's possible. If they match on the left and the right, they will preserve those edge links. So that's graphics, morphing.

Mechanics motivated a lot of mechanical linkage problems. In particular is this great book from 1877 called *How to Draw a Straight Line.* You may think you know how to draw a straight line, but the point of this book is to figure out-- by turning a circular crank, can I draw a straight line? And this is motivated by steam engines, where you have a steam piston is moving up and down in a straight line, and you want to turn a wheel on your train.

So how do you do that? Well, these are two old ways to do it, and here's what they look like this in computational land. These are the original drawings.

So we have on the top a design by Watt, the unit, in 1784, and it's approximately straight. It's not perfect, but this highlighted point moves along a circle, and the green point moves along the figure eight. And the figure eight is fairly straight here in the middle. So if you just use that part, it works pretty well. You lose a little bit from the wiggle, but it was used in a lot of locomotive engines.

Down here, you have the first correct solution by [? Percy ?] in 1864, almost 100 years later, and if you move the highlighted a point along the circle, this point moves along the red line. And there's a whole bunch of mathematics related to this, generalizing this result, and we will cover it in this class.

Next up, we have manufacturing. This is a fun example of bending a piece of wire. This is a machine called DIWire, do-it-yourself wire bending. So you can build one of these machines. Here, it's making a planar shape. You can also make three-dimensional shapes. Lots of things are possible. This is a recent machine.

There's actually a lot of wire bending machines that are out there. This is a nice and simple one. But can you make everything? No, because this piece of wire can't collide with itself and can't collide with the machine. That's a constraint. And you want to figure out, what shapes can you make? Can I make my pair of glasses, or am I limited somehow? That's a folding problem, which we will talk about.

Next up, medical. So one example of a medical application is to build a stent. This is called an origami stent, developed at Oxford about two years ago. And the idea is you want to do non-intrusive heart surgery, so you want to take this big thing, fold it down really small, like this, and stick it through some small blood vessels in your body until you get up near your heart where you've got nice big vessels you want to de-clog. And so you want to expand it back out to its larger size.

So it's essentially a transportation problem. You want to make something small, until it gets where you need it, and then you make it big. That's one example of a medical application.

Related to that kind of transportation problem, you also have motivation of aerospace. So you want to send a large object into space, and your space shuttle isn't big enough to do it. You'd like to fold it down to become smaller. This is an example by Robert Lang, who is pictured here, leading origami designer. And this is a prototype. It's only five meters in size.

The goal is to make 100-meter telescope lens, which definitely does not fit in your space station. So you do lots of foldings like this, until it fits in your space shuttle. Send it out. When you're in space, you've got lots of room, and you can unfold out. And you get your giant lens, much larger than Hubble's lens, for example. In general, this area is called deployable structures, when you want to make something small for transportation.

Then we have biology. Protein folding is a big problem. A lot of people work on it. And a protein folds kind of like a robotic arm, geometrically. It's a little different , and it's not fully understood how it works. But there's a lot of cool folding problems, geometry problems, that we've studied, motivated by how proteins fold and figuring that out.

Applications, things like drug design, you want to kill a virus without killing the host. Design a proteins that folds into the right shape, so it kills one thing and not the other. That's the sort of general goal, and have some cool things which we will cover in this class related to that.

Next is sculpture, and origami design is an obvious motivation for why you might be here. This is why we initially got interested. Origami has reached incredible heights. Both of these are folded from one square, no cuts. So just one square paper, you can make a three headed dog. You can make, I guess I shouldn't call him a man, but you can make a Nazgul on his horse. Or is it one entity? I don't know. But it's one square paper, that much I know.

This is designed by Jason Ku who's the President of the origami club at MIT, OrigaMIT, which you should all check out. They meet on Sundays at 3:00. I know there's a bunch of origami people here.

So how are these possible? These are possible through mathematics and kind of algorithms that we will cover in this class. In fact there, is a guest lecture by Jason Ku, which you'll be watching, I think, in lecture six. And he's not actually in town, but still he can give the video lecture, because he already gave it. It's like time travel. I have just a few examples of some of our sculpture. This is with Martin Demaine, who's your cameraman here.

And these are based on curve creases, which we may talk about to some extent in this class. And we didn't talk about it too much two years ago, but a little bit. Curve creases are not very well understood mathematically. Almost everything we will cover is based on straight crease design, like most origami. Curve creases are pretty amazing, though. They do some really cool shapes, and we're still trying to figure them out-- how to design them algorithmically.

These pieces are on display in DC right now if you want to check them out, for the next several months. These pieces of paper are circles with a circular hole, and there's more than one. This has two pieces. This has three. This has, I think, five.

OK. So that was a brief sculpture origami design. Lots of different other possible sculptures, kinetic sculpture, you might want to try building. Also architecture, reconfigurable architecture. I think this is an underexplored area. How many people here are from architecture? A few. You might want to make some reconfigurable buildings.

Hoberman is one example of someone embracing this a lot. He started in folding toy design, but now he does mostly folding architecture. This is one example, from his company Hoberman Associates, from the winter Olympics in Salt Lake City 2002. It's very dramatic. You have this folding structure they can almost completely close up, except for a little circle here, and when it opens up, you've got this huge stage. You can see the scale, of people.

He co-designed this giant U2 stage. If you've seen U2 any time in the last few years, like I did. This incredible folding structure is based on similar principles. And Chuck Hoberman is actually teaching a class at the Graduate School of Design at Harvard on Monday afternoon, and I think anyone's welcome if you want to check it out. You should go see it. Talk about his techniques. Maybe we'll get them to do a guest lecture. We'll see.

So that was a brief overview of a bunch of different applications. There's tons more. In fact, if you have more, please send them to me. I know a bunch more, and we will get to them, but this is a little survey of different fields that folding touches on. Now I'm gonna switch over to more mathematical stuff, which is the bulk of the class. Unless there are questions? Cool.

So we move to what kind of things we're interested in, in folding. There are essentially three types. Linkages, which we usually think of as one dimensional structures. So we have one dimensional, usually straight segments, connected together at hinges, and here I've drawn it in two dimensions. It could also live in three dimensions.

Like-- my arm is made up of one dimensional bones, and then there's sockets for them to join, essentially universal joints. How can those things fold? What shapes can they fold into? General constraints here are that the edges should stay the same length-- I can't stretch my arm longer-- and they have to stay connected at the joints. I can't detach my elbow and later reattach it, at least not ideally. The mathematical version is you're not allowed to.

One dimension up, we have sheet folding, which we usually refer to as paper folding, but it could be sheet metal, anything. You have your sheet of material, and you want to fold into things like origami, like space stations, like telescope lenses, whatever. General rules here. Like this piece of paper, you cannot stretch the paper. It can't get any longer. It can't collide with itself, and you can't tear the paper. You're not allowed to cut it, because that tends to make things too easy.

So the sort of mathematically pure version, and also the modern origami pure version, is that you're not allowed to cut your piece of paper. So just folding, no stretching no crossing.

If you want to go another dimension up, we typically call this polyhedron folding. So a polyhedron is made up of a bunch of polygons, like a cube. And typically we're interested in unfolding a polyhedron. So think of this as just a surface that's hollow inside. If you wanted to build that surface, you'd like to find a shape, like this cross, that folds into that cube. Because if you have sheet material, you want to start from some original shape that's flat that can fold into your shape.

You can also consider the reverse problem, which is folding. Suppose I give you this cross, what shapes can it fold into? And we've looked at both, and I'll talk about them in a second.

So in general, for each of these categories, and you'll see these categories over and over because they're on the book. Here it says linkages, origami, and polyhedra. And there's three parts to the book. There's linkages, origami, and polyhedra. Part one, part two, part three. It's just like this.

Our class will not follow this order, though. After today, I think we'll talk about paper for a while, then go to linkages for a while, then go to polyhedra for a while, and then we'll repeat, go to paper for a while, and so on. Different parts will be of interest to different people, so we'll do lots of different coverage. But I think paper's the most exciting, so we'll start there.

In general, for each of these types of things you want to fold, there are two kinds of problems we're generally interested in. And based around the idea that, well, there's some folding structure that you're interested in, and then there's the way it can fold. And you can start from either one.

If you start from a folding structure, and you want to know what it can fold into, this we typically call a foldability problem, or you could call it an analysis problem. I have something like this, and I want to know what it can fold into. That's an example.

What can the cross fold into? Or I have some linkage, and I have a robotic arm. It's already fixed. I've already designed it, built it. What shapes can it fold into?

The reverse problem is a design problem or synthesis problem. So there I start with what foldings I would like to have. So for example, I want to fold the butterfly. So I design a butterfly, and then I'd say, well, I have a rectangle paper, can I fold it into a butterfly? So that's a design problem. Here you want to design a folding structure that achieves your goals, like shape. So that's generically what we like to do.

And then there's three types of results we typically get. I'm giving a super high level first, and then we'll get to some actual examples. So these are typical results. So three types are universality, decision, and hardness, and these relate to what is possible. In particular, is everything possible? And this makes sense both from a foldability standpoint and from a design standpoint.

If I give you a robotic arm, I'd like to know, can it folding to all the possible configurations, or are there some that it can't reach? That, if they can reach all of them, we have a universality result. Typically when you can prove that you can make anything, you actually get an algorithm to do it. So you could say, oh, well, how do I fold into x? Well, here's how you do it. There's an algorithm to compute that for you.

Also, in design, I'd like to know, OK, I can make a butterfly, and I can make a Nazgul. Can I make everything? If yes, you get a universality result. That's a design universality. If universality is not true, if you can't make everything, the next best thing you could hope for is a decision algorithm, an algorithm that tells you quickly, is this possible, is this impossible.

So this is like a characterization of what's possible, and what's impossible using algorithms. Sometimes, though, that's not possible. There is no good algorithm to tell whether something's foldable, and then we aim for hardness result to prove that there is no good algorithm to solve your problem.

So that gets in complexity theory. Not everyone's going to know that. We will be reviewing it in the video lectures, and in here as needed. So those are the sort of typical outcomes for any of these kinds of problems. Let's see some examples, unless there are generic questions at this point, only generic questions about generic material. Next we'll get to specific material, and you can ask specific questions.

So first up is linkages. And, of course, there's lots of results in each of these fields. I'm just going to show you a couple in each. And the first question you are typically interested in about a linkage is, is it rigid? Does it move it all? Let me give you some examples of this. Do a little pop quiz. It's OK to get it wrong. Just makes for some fun interaction. So, yes or no, is this rigid? Or, rigid or flexible, I should say.

AUDIENCE: Rigid.

PROFESSOR: Rigid. Everyone agrees. Good. Correct. Rigid or flexible?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Flexible?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Depends on the dimension. Very good. It's rigid or flexible. It depends. It's rigid in 2D, because these are triangles. But it's flexible in 3D, because you can rotate one triangle around this hinge. And this is rigid or flexible?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Flexible. Everyone agrees and is correct. In general, given a structure like this linkage, you want to know, is it rigid or is it flexible? This has lots of applications. Like, you want to build a bridge, you don't want it to move. You're building a building, you don't want it to move.

So here's the mathematical state of affairs. Distinguishing this boundary, which is a rigid in 2D versus flexible in 2D, we understand super well as great algorithms to do it. We will cover them. This boundary between rigid in 3D and flexible in 3D, we don't really understand. And there aren't great algorithms to solve it. It's a little disconcerting given the number buildings and bridges we live in, but that's the state of the world. And we will talk about all that. So that is rigidity.

Next up we have universality. So this is the robotic arm question I mentioned. I have a robotic arm. Does it fold into everything, into every possible configuration? And the answer, again, depends what dimension you live in. In two dimensions, there are cool ways to reconfigure your robotic arm. So this is actually a polygon, a closed arm, if you will, a closed chain, and this is one way to unfold it. And then, in principle, you could refold that into any other shape you want.

And we will cover-- this is one algorithm to do it. There's actually three algorithms to do this. This is the most recent one. This was originally my Ph.D. thesis, to figure out whether this was possible. It's called the carpenter's rule problem. This algorithm preserves the fivefold rotational symmetry of the polygon, which is pretty cool. And in general, you take any 2D polygon, you can unfold it while preserving all the edge lengths, keeping all the connections, and avoiding collision.

I'll show you some other fun examples. This is a kind of spider, 500 vertices. We're kind of zooming out. As we expand here, you can see the lengths get really tiny, but, in reality, all the lengths are staying the same throughout time. And we fold. So this is the state of affairs in 2D. We have good algorithms. It's always possible. Universality.

So what about 3D? 3D is harder, in particular because of examples like this. We call this knitting needles. You've got two blue segments, which you can think of as long needles, and then a short, purple thread connecting them. And there's no way to undo this. In particular, if you started with a straight, robotic arm, you could not hold it in the shape, or vice versa.

We call this a locked configuration, and in general lock configurations exist, obviously. And we don't know how to distinguish good or bad. So here, we actually don't know whether there's a good decision algorithm or hardness, but it's definitely not universal.

And 3D's very interesting, because it relates to robotic arms in real life and protein folding and all these things. I'll be talking about that. In 4D. If you're lucky enough to live in four dimensions, you get universality, everything's great. And we'll talk about that too.

So next up, we have paper folding. That was our brief overview of linkages. So paper folding. I'm gonna mention one problem in the foldability world and some problems in the design world. This is probably my favorite are of paper, origami design. But foldability's interesting, too. The obvious and the first question you might wonder about in origami foldability is, if what crease patterns fold flat? So if you take a classic flapping bird, you unfold it, these are the creases you get.

But in general, if I gave you some pattern of creases drawn on your piece of paper, does it fold into anything? And here, to make it interesting, we say, does it fold into anything flat? Flat origami. You have to fold along all the creases. Sadly, deciding this is a problem that we call NP-hard. So there's basically no good algorithm to solve this, and we will prove that in a few classes, a few lectures. So that was the foldability problem.

Now onto design. If I give you a square paper, what shapes can I make? That's the obvious origami design problem. And it turns out you can make any polygon, so you can make any flat shape you want. If you have a piece of paper that's white on one side, and patterned or colored black on the other, you can make any two-color pattern you want. If it's polygonal, it's made of straight sides. If you want to make a 3D thing, you could make any 3D polyhedron, also with two color patterns, whatever you want.

We proved this way back in 1999. I think it was actually the first paper to use the term "computational origami." And it's not hard to prove. We'll prove it in a couple of lectures, I think. We don't have a practical way to do this. There is an algorithm here, but it doesn't give you a good way to fold anything. And so the quest continues for a good way to fold everything.

And one such approach is called Origamizer. You give it an arbitrary 3D surface, it designs a crease pattern. This, you fold from a square piece of paper-- solids are mountains, dashed are valleys-- and you get this bunny. This is a classic model in computer graphics called the Stanford bunny, and this takes about 10 hours to fold. If you're Tomohiro Tachi, who designed it. And there's free software available called Origamizer by Tomohiro.

And he was just here a couple weeks ago, and we've been proving that this algorithm always works. And it also seems very practical, most of the material here gets used in the surface. You don't need a very large square of paper to make your bunny.

I have one little example here, sort of classic in the origami world, which is to make a checkerboard. And this is a four-by-four checkerboard, obviously not the full eight-by-eight. But you can fold an eight-by-eight. We'll talk about different ways to do it. This one is folded from one square paper, white on one side, green on the other. And, actually, it has a pretty good scale factor, just collapses by factor of two here. See if I can get it back into correct state. So there you go. You can make your own origami checkerboards if you ran out of your regular wooden ones.

Next example is what we call maze folding. This is the poster. If you saw it, this is a folding of this crease pattern, and the design was-- In general, you take any orthogonal graph, meaning horizontal and vertical lines on a grid, and you want to extrude that graph out of the plane, then there's an algorithm. It's online. You can play with it. I think erikdemaine.org/maze will get you there. You just draw that pattern. It will give you this crease pattern. You print it out, and-- eight hours, maybe-- you can fold it into realities.

So this is by [? Jenny ?] and [? Eli, ?] who I think are students in this class. They got started a little early. And I didn't bring it here, but at some point I will bring to the real 3D one. So that's maze folding.

And the next one I have is called folding cut. This is actually the first problem we worked on, so kind of a personal favorite. You take a rectangle of paper, and then you fold it flat, and you take your scissors and make one complete straight cut.

This is a magic trick performed by Harry Houdini and other magicians, and you get, in this case, two pieces. And the question is, what shapes can you make by folding and one straight cut? Here I made a little one. OK, you're not impressed, so I'll do a harder one. This one is actually quite challenging to cut. I was folding it as you were arriving, as you probably saw. That's about right. It's a little hard to open up, but this should be the MIT logo.

AUDIENCE: [INAUDIBLE]

PROFESSOR: All right. Good.

AUDIENCE: [APPLAUSE]

PROFESSOR: So that's the idea. You can impress all your friends. Print these out, and fold them. It's a good challenge to fold along the crease pattern, and as you might guess there's an algorithm here. Given any polygon or collection of polygons, you can fold a piece of paper to line up all the edges in that polygon, cut along it, you're done. OK, that is paper

Next I want to briefly mention polyhedra. Actually, I have a polyhedra right over here. Let's just go there. So, again, there's an unfolding problem, and a folding problem. And I'll start by showing you a little bit about the folding problem. Oh, this shows a little bit of both. This a video we made back in '99. I was a grad student. I had lots of hours to make videos, animations. So this is one example of an unfolding of a cube. But the unfolding we want to consider here is this cross unfolding that I keep talking about. It's kind of a classic so fun to analyze.

So if you take that cross shape, you want to know what other shapes, what other convex polyhedra, can it fold into? And here the rules are little different from origami. You have to get exactly the surface you want no, overlap. Here we happen to get a doubly covered quadrilateral, which is kind of a convex surface. A little bit cheating.

But if we change the creases in this way, we get a five-sided polyhedron with a plane of symmetry down this axis. Or we can change the creases in this way and get a tetrahedron. And it's exact coverage, so this little tab fits into exactly that pocket. Exactly what you want.

As you might guess, there's an algorithm that will enumerate all the different convex polyhedra you can make from a given polygon of paper. And it's based on a theorem in Russian by Alexandrov, which is in the background here. If you're curious what the Cyrillic background is, this theorem is the key to the algorithm. So that's what's possible by folding a given polygon, and then you can ask about unfolding.

Let me give you a brief overview. Unfolding. So when you're unfolding, you might be interested in only cutting along the edges of the surface, which is what happened in this cross unfolding. Cut along these edges. Or you might allow cutting anywhere on the surface. And the answers depend on those two versions. You could try to convex polyhedra, or you could try to do arbitrary polyhedra.

So here's the story. This one, we don't know. This one, we don't know. This one, the answer is yes. We'll prove it. This one, the answer is no. We'll prove it. In fact, I can show you some examples.

This is an edge unfolding of a convex polyhedron. The oldest set that we know this from this 1525 book by Albrecht Duerer. And this is a snub cube. He didn't have Wikipedia at the time. You couldn't just look at the image and draw it. He had to build a physical model. And so he cut out pieces of paper to build physical models, so he could paint it. And various of his prints have convex polyhedra in them. And this is an example of this open problem, which we don't know the answer to. Can you edge unfold convex polyhedra?

If you want to take non-convex polyhedra you cannot always do it. This is an example of a spiky tetrahedron that cannot be edge unfolded. We'll prove that. But it can be generally unfolded. If you can cut anywhere, you can find a one-piece unfolding. And so big open problem is, can you do any polyhedron with general cuts? I think the answer is yes. Very cool problem. And that's polyhedra.

And the last thing I wanted to show you is hinged dissection. This is one more thing. After one, two, and three, we have four, hinged dissection. This is a chain of blocks, which are kind of two dimensional, but it's not really paper folding. So it's like a thick linkage between one and two, if you will. And this chain folds into an equilateral triangle or a square, and it's over 100 years old. Can you do anything? Turns out there's a universality result.

This is just a picture of part of the proof, which we will cover in a later lecture. But you can fold, you can take any set of polygons of the same area, there's one chain of blocks which will fold into each of those polygons. So you can get a universality result. You can do an equilateral triangle to a regular hexagon to a swan to whatever. As long as they're all the same area, you can do it.

And this also generalizes to 3D. We use that to make a little sculpture here, little sculpture. About 1,000 blocks, you pick up with your gloves, these blocks, rearrange them. And the theorem is, you can make any shape you want out of the blocks by this kind of intersection.

And that was a brief overview of a few of the results that we'll be talking about in this class. There's a lot more. And you can check out the 2010 web page if you really want to know everything that's covered, but we'll be posting lectures up there. And let me know if you have any questions.