PROFESSOR: All right, so before we go into the stuff that I'm going to cover for the readings today, I want to get a sense of who's already found a team. How many of you are already in a team? Half the class, OK. And the rest of you, what we're going to do is try to solve that problem right now and get you in a team. For the people who are in teams, can you say what mechanic they're working with, and how many people you have in your team?

AUDIENCE: What if we have two mechanics, so we're not sure of that yet?

PROFESSOR: That's fine. Just talk about both of them, and then you'll figure it out, probably today.

AUDIENCE: OK.

PROFESSOR: So what are they?

AUDIENCE: We were thinking between either like pathfinding or some kind of resource management.

PROFESSOR: OK, pathfinding--

AUDIENCE: I'm sorry, path-building.

PROFESSOR: Path-building and resource management.

AUDIENCE: And we have three people. OK, so three right now,

PROFESSOR: OK. What else?

AUDIENCE: Hidden information, and we're four people.

PROFESSOR: You have four. Hidden information, so that's four.

AUDIENCE: Stealing or like team sharing.

PROFESSOR: I'm sorry, stealing, or?

AUDIENCE: Or like teammate sharing.

PROFESSOR: Team mate--?

AUDIENCE: Sharing. Sharing your teammate.

PROFESSOR: Oh, OK. So stealing or team sharing. I think they say team sharing. How many people? Four. Is that it? OK, and how many people are looking for teams? OK, one, two, three-- is your hand up?

AUDIENCE: No.

PROFESSOR: Ok, one,two, three, four, five, six people. All right. So we could make two complete teams out of the remaining six. I think that's right, did I count that right? So that's what I would suggest is that all the people who don't have teams-- we try to make two teams out of that, rather than try to join this team. Because otherwise, we end up with a two-person team somewhere, and that's not good.

All right, so of the people who aren't on a team, we went through brainstorming on Monday as a whole class-- I can bring up the list again, but was there something that you remember from Monday that you felt like you would like to work on?

AUDIENCE: Either voting or bankruptcy. Or basically trying to bankrupt people.

PROFESSOR: Bankruptcy.

AUDIENCE: Your numbers-- there's 16 people. You went ahead and said there's four 4-teams and six unassigned.

PROFESSOR: There are six unassigned right?

AUDIENCE: But there's-- you counted a total of four, four, three. Someone's being counted on two teams. [INAUDIBLE] When they come. They're not here right now.

PROFESSOR: The remaining people who are not here, if we have a bunch of three-person teams, it's a lot easier for extra people to jump one. And three-person teams are pretty easy to work with. Four-person teams are also. We'll be working, but it'll be a lot easier to schedule a meeting with three of you. So three-person teams are good. Two is pushing it because if someone gets sick, you're in trouble. Yeah?

AUDIENCE: Back to mechanics, I was interested in trading, or maybe building, or expansion.

PROFESSOR: Trading, building... Did you say expansion?

AUDIENCE: Expansion. Territorial expansion.

PROFESSOR: OK, so by expansion, we mean territorial expansion. OK. Yeah, we see these three things in one game a lot, right? But for this assignment, let's see where we can deal with one.

AUDIENCE: I'm interested in building or area control. I would be interested in deception or what the unfairness means.

PROFESSOR: What?

AUDIENCE: Unfairness.

PROFESSOR: OK. So All right, so here are the different kinds of game mechanics that the people who aren't assigned are currently-- at least one person is interested in them. So what I'm going to do is I'm going to go through each one of them, and for the people who aren't assigned, put up your hand if you think that you might be willing to work on a team on that concept. OK? So, starting with voting, we have-- OK so three, bankruptcy-- one, trading-- three, building-- four, expansion-- it's territorial expansion again-- four, back building-- one, area control-- four, area control and expansion might actually end up beating each other then. Deception-- one, two, three, four, and unfairness-- one, OK. So we've got a bunch of things-- I thought there was a lot of interest in area control. I think I'm going to leave that off because we will be revisiting the topic later in the semester. So you'll get a chance to look at these. So I'm going to take these two out for now. I'm gonna take out all the one word ones that will make it tricky. That leaves us with voting, trading, building, and deception. Of the people who aren't assigned, was there one that you're not interested in? Any one that you're not interested in working with any of these.

AUDIENCE: I don't particularly like the [INAUDIBLE].

AUDIENCE: No, he's saying are you not interested in all four.

PROFESSOR: In one, two, three, or four. If you're interested in one of these, I think we can make teams out of this. All right, so later on in the class, when we start the prototyping, there's going to the time to actually talk with each other. What I'm going to actually suggest is all the people who are not on teams switch with the front row, and all people who have teams, switch with back row. Probably, one corner will also have to be a team that already exists. So yeah, actually, let's do that now. So if you're not on a team, switch with front row so we can all have discussion.

[SIDE CONVERSATION]

PROFESSOR: You might as well sit with your teams if you already have one. Because we're going to talk about something new together today.

[SIDE CONVERSATION]

PROFESSOR: All right, for the people who aren't in teams, remember the goal is trying to make two teams out of this, of any combination. If you end up changing the game mechanic, that's fine. All I'm looking for is two teams, out of the six people who aren't assigned any.

[SIDE CONVERSATION]

PROFESSOR: Who hasn't signed in yet? Who hasn't signed in yet?

AUDIENCE: I haven't. My name isn't on it.

PROFESSOR: Write your name in.

AUDIENCE: I was pre-registered.

PROFESSOR: You're pre-registered?

AUDIENCE: Yeah, and I was there before. My name is on the list the first time. Now my name's not on the list.

PROFESSOR: Very weird, something weird. I'll take that straight outside, so write your name down. You did awesome, and I'll check it out during the break and see where to fix that problem. All right, so how many of you have seen this presentation from me before, in any of the 15 times I gave it in the past year? OK, all right, so about four people. This is a presentation that I give a lot. It's also something that ends up getting covered in one of our new classes this semester CMS301. This is probably gonna be the last time I actually give this presentation in CMS608 because it's kind of like a really basic skill, we're gonna be moving this into our intro classes in the future. But it's also the one skill that, if you learn nothing else from the rest of semester, but you still didn't come to class, please learn this. Because this is the core skill that we're going to be asking you to keep working and keep practicing and keep improving on, all semester long. This is the thing that you're going to be doing all semester long. So first of all, that's kind of jumping the gun. Let me take a step back. What's a prototype? It's in the reading.

AUDIENCE: Like a basic thing you just toss together to illustrate a concept.

PROFESSOR: To illustrate a concept, yep.

AUDIENCE: Isn't that supposed to be one mechanic and reiterate that one.

PROFESSOR: For game, it could be something that just tests one mechanic or concept, and then, you have to reiterate on it, sure.

AUDIENCE: Something you just put out there to get feedback.

PROFESSOR: Yeah, to just sort of like gauge how other people are going to respond to it.

AUDIENCE: In general, It's just an unfinished version of the game.

PROFESSOR: It is unfinished. It is not like your shipping product. At no point it should seem to be a shipping product.

AUDIENCE: I'd say it's that first version of anything, that was specifically built just to test that thing, rather than to have a product.

PROFESSOR: Again, just to test an idea, to test whether something could work, and, sir, you said first version could be an early generation of something.

AUDIENCE: I was gonna say like a minimum usable product.

PROFESSOR: Something that you can actually use, not like a sketch of a game. An actual game that you can actually play. Anything else? I thought I saw a hand back there. I think we are getting kind of a good sense of what a prototype is. You probably encountered prototyping in some other classes-- a lot of engineering classes involve prototyping. It is this unfinished thing. It is not meant to be an iteration of something that you're actually going to ship.

Now, in this class, a lot of things that you're going to end up building are building towards a finished class assignment-- the thing that you hand in that meets all of the criteria, that has all the rules, and if you put in front of someone who's never seen your game before, they should be able to figure it out. However, the very first assignment, it's pretty build a prototype. You're testing out one single mechanic. And the question that you should be asking yourself is, what are all the different things I can do with this one mechanic? And then, you can just deep dive into this one big open question. And you're gonna end up choosing that question for yourself, but as an investigation.

And prototyping is a tool to help you investigate something, that is going to help you build your final game in the end. So think of assignment 1 as an exercise that's actually going to help you build assignments 2 an assignment 3. Even though we are asking for things like rules that we can read, but for the most part, this can be very sketchy, very unpolished.

If we can play it, if we can use it, I think someone said minimum usable product-- to be able to see the ideas that you're working with, that's good enough. Do spell check your work, that would be nice. But some of the reasons on why you want a prototype-- where are my slides notes? Just one second-- I have no slide notes. OK. So some of the reasons for why you want a prototype-- we already talked about getting feedback-- being able to put it in front of people who may not necessarily have seen your game before, but may be part of your target audience-- to be able to get their opinion. Put it in front of instructors who may never have seen other version of the game before or maybe have seen the game, to be able to get their feedback. Or other designers or guests that we might be bringing in-- you want to be able to put something in front of everyone who knows something about games to be able to get their critique, right?

But the earlier and cheaper part is the important part. You want to be trying to get feedback, as quickly as possible, in your game development process. The earlier you manage to get new information into whether your ideas are working out or even appealing, then the cheaper it is to make changes.

So for instance, you've got a great idea for one of the game mechanics, and it turns out that everyone else on the team absolutely hates it. But you'd rather find out about that on the first day of meeting up with your team, than, say, two weeks into the project. That would be very useful information to sort of say, "OK, I can work with other ideas." Are you on a team? You are a team. Awesome, OK. So I don't have to worry about getting you on a team.

The other thing you are trying to do with a prototype is to try a lot of different approaches to the same problem, to experiment a lot of alternative solutions. One thing that often happens in design teams, professional design teams, amateur design teams in school or outside-- the people like to talk thier ideas out. They love to make sketches, they love to theorize about-- well, it worked in this game, so it should work in our game-- I really like that, or i really hate that in this other game. This offends me on some sort of primal way-- a game designer could explain it to you verbally in about an hour or so.

And that just wastes a lot of time, especially in a class like this when you have these very tight deadlines to be able to get something working. You don't really want to be spending a whole lot of time talking things out. You want to get actual answers as quickly as possible. And one thing nice about prototypes, especially on the multi-person teams is that can make multiple prototypes. It just tests a whole bunch of different ideas out. If solution a is better than solution b, well, instead of arguing about-- well, theoretically solution a is better than solution b, why not just prototype both of them and just play them. You'll get an answer very, very quickly among you, and all of your teammates can see. And it makes it [? special ?] and actually more fruitful because you're working with evidence, as opposed to working with just theoretical notes.

Another thing about prototypes is that, again, prototypes are not your shipping product, which means you should always be willing to throw it away. The less time you spend making a prototype, the easier it is to just discard it. With looks, the uglier and sketchier it is, the easier it is to abandon it. And you need to be able to abandon prototypes. You need to be able to say this just isn't working, and I'm fine with that-- the whole team is fine with that. You spent 30 minutes taking this thing, we can afford to lose those 30 minutes. Which means you need to be making stuff either really, really fast and really, really shoddily. You don't want to be spending a lot of time making a polished prototype.

Here are goals that I am setting up for you when you're going into prototyping-- today we're actually going to start prototyping the games that you're going to eventually hand in for Assignment 1. I want you to find the fun. All of these game mechanics, the mechanics that the teams have already chosen, and the mechanics that the teams will end up choosing-- all of them have fun and un-fun implementations. I can think of a whole bunch of ways to make building go at a really plodding pace, to make it so that you can never really get any progress, and things like that. And you can come up with really, really nice implementations that are just going to engage everyone around the table-- they're gonna have a good time.

Fun doesn't necessarily mean everyone's happy. It's like a game where, deception for instance-- it's like you're playing around with that mechanic, and you feel like you're just being an asshole to other people. But that's what the game's actually about. You are trying to be deceptive to other people-- maybe not necessarily without them realizing it. But then, [INAUDIBLE] experience, that engages you and sort of puts you in the persona of what the designers were trying to achieve. Then, that's engaging. That's good. If it's something that puts people off from ever playing the game again, then, you might want to re-evaluate that. But there is some value in games that you're only gonna play once. I'm not going to be very dogmatic about that.

What I want you to do throughout the prototyping process is figure out, of those game mechanics, what are the fun and engaging things that you can do, and what are some of the less fun implementations? If you don't find anything that's not working, then, I don't think you're looking hard enough. You need to be looking really hard to-- What you should be finding is whole bunch of things that don't work, and a few little gems that do.

The other thing that I want you to do with the prototype is use the prototypes to communicate to the rest of your team about where your ideas are going. As you work in your team, if you want other people on your team to understand the ideas that you have in your head, try making a prototype to communicate that. This is this thing that I did last night, I bring it into the team meeting-- together let's play this for like five minutes. And then, you'll understand what I think is interesting about voting, for instance. And your team may have a completely different idea about what they heard, about where they wanted the game to go, but this is a very effective communication tool.

When you're actually designing games, especially for assignment 2, and assignment 3, we are trying to hit some sort of desire, external aesthetic. The final assignment is going to be for our client's needs-- then, you are working on some sort of external spec. But then, you also need to communicate within your own team on how you're interpreting that spec. That requirement, that request, from external party in this class will be your instructors. Now, if I say make a game that is going to get you to a certain aesthetic, which is assignment 2, then how do you think the team should start even proceeding in that direction? Communicate that using your prototypes. It can be very, very difficult to get those ideas out otherwise.

The third thing that I want you to do with your prototypes is to take it outside of your team. Today is going to be very easy because we have a room full of people who are hopefully eager to help each other out. And they're going to end up playing each other's prototypes by the end of class. That's only gonna last for one class because, as of the end of today's class, all of you are going to know too much about each other's projects to actually be good testers in the future. So I want you to take it to people outside of class-- your dormmates, your friends and family. Email them stuff if they're at home-- take it to the libraries or the student center or something. Offer people free tacos or something to play your game for five minutes, that sort of thing.

It's very, very important to make sure that you're getting feedback from people who are not on your team. Of course, the instructors-- it's not just gonna be us. We're gonna grab people from the game lab to come in here and play your games and give you feedback of them. So occasionally, we'll just bring in new people who haven't seen your game before, but don't count on us doing that. You should be doing that as part of your homework. That is the process of prototyping.

OK, before I go down that shopping list, any questions so far about what you're trying to achieve with prototyping? OK. Again, Assignment 1 is going to be a lot more about prototyping than making the full game. Investigating one single game mechanic is something that you're probably actually gonna end up doing for both Assignment 2 and Assignment 3. Because it's like, hey, this is the way how I think we can fulfill the requirements of the assignment, and then, you're gonna investigate several different mechanics to get to building your final game. So think of Assignment 1 as the prototyping assignment.

Let's talk about what we've got for you today-- things that we recommend for prototyping include large sheets of paper. Here are a couple of preprinted maps-- there's a hex grid on one side. There's a regular horizontal and vertical grid-- is there a word for it? Cartesian?-- square grid. These are, I think, two centimeter squares, which is also the size of some of the wooden blocks that we've got. I think the hexagons are also two centimeters if you to take them horizontally.

So you can print these. These are all actually available in PDF on our class website-- the prototyping maps. There's a second version of these maps that has marks. It has just a bunch of lines in bold. It's exactly the same map, but it has a few things boldfaced to help figure out what a square in here actually looks like, if you want to use a counting track, or something like that. I like these better. I know Rick likes the other one better, so we put both up for you to download.

Let's see, what else do you need? Dice. There's a big box of dice, 6-sided dice, 12-sided dice. Has anyone ever used a 12-sided dice? Really? What are they good for?

AUDIENCE: D&D.

PROFESSOR: D&D does d12s?

AUDIENCE: Yeah.

PROFESSOR: OK. I know they do that d20s a lot. But, OK. 10-sided dice, 8-sided dice-- If you ever buy dice for yourself, a tip is to get them from kindergarten suppliers, rather than from gaming stores. Gaming stores will probably cost about 10 times as much. This box cost me about $12, plus the box. You get a nice little case.

Dice are good for randomizing, obviously. I do not suggest using dice to keep track of numbers. So say you've got a number that increments anywhere between 0 to 20-- sorry, 0 to 19, or 1 to 20. Don't use a 20-sided die to keep track of the thing because it's very easy to lose that stat just by a flick off your hand. Just grab a piece of loose newspaper or something, and just write down that number-- if you need to keep track of stats. Don't use dice to keep track of stats. But dice are good for randomizing, they can be used sometimes as tokens that move around in a pinch. So you can use dice to do things, like one die is the tens, and one die is the ones. So I rolled two 10-sided dice, and then, it's going to give me a number between 1 and 99-- no, 00 and 99. Yeah.

Index cards-- we've got white ones in here, we've got colored ones in here. They sell index cards that are a little bit closer to playing cards size, but remember the rule of keeping things big-- big sheets of paper. I'll go in a little bit more detail of why you want to do that. But if you're making like a card game, I would actually suggest using index cards. They are a little bit harder to hold in your hand. It's difficult to keep a whole bunch of index cards in your hand, like a fan.

AUDIENCE: They're hard to shuffle too after a while.

PROFESSOR: Yeah, they're hard to shuffle as well. But for prototyping, not only are they cheap-- we talked a little bit about doing things cheaply and very disposable. Those are larger because it's easier for you to actually see when you're prototyping. And I'll get into some value of why you want to-- a few more reasons of why you want to be keeping things as big as possible.

Post-it notes are not in here. They are in those boxes. Post-it glue and notepads-- by notepads, I mean stuff like this. Gets people to keep track of stats. We've got pencils, and we've got markers and pens in there, as well. Post-it glue- if you haven't see it, I'm pretty sure it's in one of the boxes-- it looks just like a regular glue stick, but it's blue. I might only have some in my office. Basically, it takes any piece of paper and turns into a Post-it. It makes it into sort of a restickable piece.

And not only is that useful for the brainstorming phase, where you can take index cards and turn them into Post-its and stick them up in a wall, they're really handy for prototyping because you can lay things out, say on a desk or on a sheet of paper. And it won't just go flying if somebody sneezes. You can sort of keep things in place. If you've never done prototyping for, say a user interface on a piece of computer software, it can also be really, really handy. Because you can cut out pieces of paper that are exactly the size of your menu bar or your window, or whatever. Just use the glue stick, make them replaceable and sticky, and you can place them anywhere on another sheet of paper.

Pencils, pens, markers, scissors, tape-- that should be obvious why you want all of those things. Gamebits-- some of you saw some of these in last week's games. These are sort of stackable counters. We have the cubes. You can also dice for gamebits. You can also use pieces from other games. You don't necessarily have to restrict your ideas to what we give you.

Those, I believe, are rubber animals-- often end up being used in tactical combat games, for some reason. I don't know why people always want to pigs to fight chickens. Ducks hate pigs, right. Ducks don't stand up, that's the problem. That is the problem with our-- we have a whole bunch of little rubberized animals, and they don't stand up very well. They do tend to tip over. We also have a whole bunch of rubberized vehicles, and those tend to be a little bit more stable. So keep that in mind before you decide, oh, I have to use this chicken piece. Kindergarten counters, things I use to teach kids how to count, make great, great covers for your design.

AUDIENCE: I think we [INAUDIBLE] or two.

PROFESSOR: Oh, you mean, like the ideas right there-- different colors on your side. So you can use them for currency in your game, for keeping track of points-- another way to keep track of a status-- just give people pieces that help them keep count. Your phone camera is extremely useful in keeping an archive of your work, of keeping track of your game and play, keeping track of who has what hand at any given time. It's really, really easy, and a lot of phones now have a resolution, that you can sort of reliably use them to keep a record of your work-- way easier than trying to start everything on the photocopy machine. So I used to recommend using a photocopier, but now, just take lots of shots with your phone camera while you are working.

So you want to be keeping your prototypes rough. You want to be using hand-drawn materials, trying not to immediately go to opening up a Google doc and creating a spreadsheet, or anything like that. Start writing things down. Again, you want to make a bunch of cards-- just like writing stuff down on index cards. So if you want to start making a map, just start using a marker and drawing it on the grid. You want to keep it sketchy, and you want to keep it large. And you don't want to be using too many inks. Just use one dark ink, and run with that. The reason for this is because you don't want people to be giving you feedback on how your game looks. If they do say something, wow, this looks like crap, that's fine. And just move on from there because that's not the feedback you were looking for in the first place. If you start using lots of different colors, everyone will start talking about your color scheme-- maybe there should be red, maybe there should be green.

If you start making things, say printed out from a laser printer or something like that, people are gonna ask, wow, is this final artwork? It looks not very good. Or if you take the trouble of using colored pencils, for instance, to nicely render an image on your cards. And people say, wow, this looks great. You hand-drew that, but then, now, if you wanted to alter it, it means you're going to have to go through all that effort again to draw a new picture. And that takes time. That takes more time than you need for this thing.

You want to keep things sketchy to sort of convey to your testers that this is a work in progress. If somebody sees something that actually looks very, very nice, they are going to think that you're close to final. And they are going to be a lot more hesitant in giving you drastic feedback. Things are going to, sort of, require drastic changes. But if it looks like you've spent half an hour, maybe 15 minutes, just sketching stuff on bunch of cards, you'll get feedback from testers-- stuff like, I just don't like any of this, or like, maybe this is the one thing I like, but everything else is just crap. But that's the kind of feedback that you want to get at the prototyping phase. And keeping things sketchy can sort of encourage people to give you that kind of feedback. I definitely have lecture notes, but they're not showing up on my screen. And so, I'm a little bit off right now.

Here we go, OK. So the other thing is-- this is actually a 608 class from way back. The other thing that I want you to do is keep iterating over and over and over again. Yeah?

AUDIENCE: So you said like, earlier on, we can't-- after this class, we probably won't be able to playtest each other becuase we'll know too much about the games. Does that also mean we should be changing our prototyping outside this class? Find a new group everytime we revise the product?

PROFESSOR: Absolutely. The next time we do a prototype-- a playtest in class, I'm going to specifically say if you try to find a game where you don't know anything about that prototype before you start playing the game. That's where the feedback is going to be the most useful. You don't want people to come in with preconceived notions based on people's prototypes because your prototype may have changed completely from the last time that they saw it. But then, they're going to come in thinking that your game is like some sort of natural progression from that previous idea. And they may respect your ideas of what kind of strategies are used. If they already understood the rules-- what you verbally explained to them on day one, then they can't give you any useful feedback on how well your rules were written because they already know the rules. So when they read your poorly written rules, they can't tell you it's poorly written because they already understand it-- that sort of thing. So yes, always try to find new testers.

So the purpose of iteration is to just repeat this over and over again. You start with a question that you're trying to answer. The broad question of Assignment 1 is what are all the different things that you can do with this mechanic? But say, I know someone suggested auctions on Monday-- what can you do with a Dutch auction? How does a Dutch auction actually work? --that sort of thing. You want a question that, not only is a clear thing that you can actually test, but also you can set criteria for what will be a successful test or an unsuccessful test. The question might be very specific, like this game is too long. Can we get this to run under 20 minutes? Can we get this thing to run under 15 minutes? Well, that's a falsifiable question, right? If the game play actually took more than 15 minutes, then it was a failed experiment. And if it took less than 15 minutes, then it worked. And you want to be able to go into the process asking, what is the thing that we're trying to solve? --before you start thinking of potential solutions.

How many of you have heard of axiomatic design? It comes from McKee, I think. The theory behind axiomatic design is that, you have a couple of-- for any potential solution to a problem, it needs to fit a certain set of criteria. So you come up with a bunch of axioms, which you just take for true. So certain axioms that you might come up with a game would be-- this game can't take more than five minutes, or this mechanic can't take a player more than 10 seconds. So that axiom could be something like, we don't want the player to have more than five cards in their hand. These are things that aren't necessarily always the right answer, but you're going to, sort of, set these criteria for yourself for a given test. And then, you start thinking about all the possible solutions to get you there. And if one of those solutions meets all of your criteria, or all of your axioms, then that's a successful test. If not, then it's an unsuccessful test. So a question in your game may be like, can a player execute this mechanic in more than one way? Or will a player execute a certain given mechanic in more than one way? We've given them three different ways to do it, but if they keep doing the same thing over, and over, and over again, then that will be an unsuccessful test. OK?

Once you've got a question, you can start designing for that, right? Maybe the designing involves something very small, like I'm just gonna tweak numbers off rules that we've already written. It might be, we've got to rewrite half of our rules, or we've got to throw out this rule. Or maybe, we're gonna rearrange the order in which these rules are going to be executed. That's all design. The trick is to do it fast. The word rapid is there for a reason. If you are spending a lot of time discussing about what the right answer is, just start designing two prototypes-- or more prototypes to sort of test out all the outcomes. Anything that takes a long time to kind of get bumped down in discussion-- it's actually wasting time for your team, when you should be prototyping. Because they're gonna learn a lot of things about your game on the side, besides the question that you're asking. If you have a discussion, you'll probably only-- if you do stumble across the correct answer, you're only gonna get the answer to that one question that you asked. Make more prototypes to answer your questions, rather than try to talk them out.

Then, you do a playtest. And that's the second part of my presentation, which involves the playtesting phase. But basically, you've grab a bunch of people who don't know how this game is gonna play out. First, you will probably end up playtesting within your own team, just to make sure that everything makes sense. And then you take it out to somebody outside, maybe someone else in the classroom, to see how they respond to it.

And then you look at the results of that playtest-- did it address the problem? Did it give us any information towards the question we were asking? Maybe it was inconclusive-- you need to do another playtest. Maybe it indicated that we were in the right direction, but the changes that we made weren't drastic enough, or maybe were too drastic and [INAUDIBLE] down. That's when you make your revision, and then you repeat the whole process again.

You can improve the quality of your question, be more specific. You might stick with the same question and just do a second version of design to it. You just want to be repeating this over and over again. The more times you get to do this, the more refined your prototype is, and the more refined your final games are gonna be. This is the same process, whether you're making a prototype or whether you're making a full-blown board game or card game or computer game. The more chances you get to iterate on something, the more refined it's going to be. You do want to keep changing.

Here are a couple of tips that-- actually, I will get back to this later after you've actually had a chance to prototype once. Those are like tips for how to get out of rut. So let me talk about this instead-- keep track of all of your rules.

Write your rules. You can write your rules on cards, which makes them very easy to rearrange, to discard, to say-- all right, we're not playing with this rule right now. But then, maybe you can reintroduce it later. So you can use the index cards for that. You can rearrange them to rearrange how they end up getting played. If you change a rule, update your card. It is something like, I'm going to just change a number. You can just do it right on the card-- if you're actually changing the way how a rule works, write it out on a new card. Take photos with your cameras, and try to simplify your rules to the point where you end up with like a minimum set, in order to make a certain prototype playable.

If you have too many rules operating at once, it can be sometimes very, very-- really, really confusing to figure out where everything is going wrong. It's a lot easier to add new rules than to take them out, which is why I place the emphasis on taking stuff out. Because if I remind you to take it out, maybe you will do it once in a while.

So that's going to be the process of prototyping. We're going to start handing out all of these materials. People who haven't figured out your teams yet should be having a discussion on what mechanics you guys want to work on, and how are you going to split up your teams. People who know what mechanics you're working on, or maybe are trying to decide between two mechanics, you can start splitting your team into two and working on two separate prototypes, for instance.

And the goal is to have something that somebody outside your team can actually play by the end of class, more accurately, by 3 o'clock. Because we are going to go into playtesting at 3 o'clock. And around 2:30, I'll go back to this slide-- to give you some ideas on how else you can change your designs to be able to help you get closer to your design goals. But right now, this is what I want you to do, so we're going to start handing out some of this material.

AUDIENCE: Do you want another table outside [INAUDIBLE]?

PROFESSOR: I'll be OK with the team moving up there.

[INTERPOSING VOICES]