**PROFESSOR 1:** So, yeah. So the publisher can demand, you know, I want to see this feature. EA demands that this game works with Origin. Ubisoft demands that U-- is it Uplay? Is that what-- that their universal sign-up system gets added on. Things like that, right? That happens all the time. Other stuff that's maybe money-related or market-related? Sometimes expectations just--

**AUDIENCE:** This feature in the past didn't sell very well, so you should add it to every--

**PROFESSOR 1:** Yeah. That usually is similar to the publisher thing. Someone's trying to push that. But somebody could come internally from the development team, right? They look at competing products and say, hey, this thing seems to be doing well.

You know, let's add hacks. Because hacks seem to be doing well for some games. So we can add hacks to our game. It's not going to necessarily change the way how the game functions, but it's still a new feature. It's still a new-- you've still got to come up with mechanics to unlock new hacks, and equip new hacks, and make sure they animate correctly, and things like that. Even if necessarily the gameplay decision making you make in the game doesn't change. Of course, hacks actually do change the way some games work.

The changes in marketplace in general. And sometimes they're making a sequel to a game, or sometimes you're adding number two, or number three, at the end of the title. And your players are going to expect something new. And so there's some pressure for you to sort of add new mechanics instead of simply, say, iterating on the old ones.

Why would you cut mechanics? Not necessarily cutting features, but cutting mechanics-- game mechanics. There's one good reason I've been drilling in to you all semester long. Yeah.

**AUDIENCE:** Simplify gameplay and decrease the learning curve? Or decrease whatever--

**PROFESSOR 1:** Decrease learning curve is a little bit separate from simplify gameplay. Both good points. Decreasing the learning curve is like-- testers just don't get how this game functions. We've got too much stuff going on simultaneously. You could try to tune those mechanics to make it a little bit more legible, or you could just try taking the mechanics completely out so that people can actually learn it.

Simplifying the game-- could be that, but it also could be just trying to make what's core about the game more obvious. You know, this game is really about generations and generations of heroes, you know? That is a game, right? Hero Generations? Yeah.

**AUDIENCE:** Yeah.

**PROFESSOR 1:** Yeah. And you've got, say, some sort of very elaborate weapons crafting mechanic or something in there. Well, the weapons crafting mechanic may not be really what your game's about. So you might want to take that out because people sort of forget about what the core experience of the game is supposed to be. So focusing back on the core of the game. What else? Say your game-- oh, yeah.

**AUDIENCE:** New technologies come out?

**PROFESSOR 1:** Oh. New technologies come out-- like what, physics? You know-- maybe?

**AUDIENCE:** Yeah. Or has something to do with things like-- well, they mentioned-- well, I don't know. 3D, for example. 3D comes out it changed--

**PROFESSOR 1:** Well, that's true, right? I'm trying to think of a game where that's a really obvious example. Can of think of a game mechanic that substantially changed when it went from 2D to 3D? I'm thinking Mario, but--

**AUDIENCE:** *Mario*--

**AUDIENCE:** Movement, because now, instead of moving left, right, up, down--

**PROFESSOR 1:** Just straight-up movement. Well, the way how the camera responds to movement as well-- that's a big one, right? In a 2D game, say a side-scrolling game, like *Metroid* or something like that, you don't have this situation where the place where you're trying to go to is obscured, unless it's actually physically blocked by something that you have to destroy or open up or something. But in a 3D space, that happens all the time. You know, you've just got bits of background scenery getting in the way of the camera.

*Mario* just changed the way jumps work entirely, when it went from the 2D*Marios* to *Super Mario 64.* And every iteration on top of that has added new ways to jump-- double jumps, triple jumps, jumping off the wall, and stuff like that-- now, the latest Mario games expect that you know all of that. Which is why I can't play Mario anymore, because I didn't go through that process. Although the games are not that hard, at least the first couple of levels are not that hard.

Let's see. What else? There's one that I've been always mentioning, this entire semester. It-- hmm?

**AUDIENCE:** Maybe just to try something.

**PROFESSOR 1:** Just to try something?

**AUDIENCE:** Yeah.

**PROFESSOR 1:** Oh, yeah. Yeah, just to see how it's going to change how the gameplay works, right? You know, let's take something out and see whether that was actually essential to the experience or maybe getting in the way of the experience. You don't know. You know, and the emergent results of taking out a mechanic might actually end up as a more interesting experience.

One thing that the book does mention is to take stuff out that might actually change the ESRB rating of the game, or the PEGI rating if you're selling something in the UK or in Europe. So for instance, if you're building on top of a game engine like Unreal or something like that, which has body location damage-- and then you shoot a head, the game knows that you shot the head. But maybe you don't want your game to be that specific. Maybe you're designing a game for kids or something like that, and it's like, well, rewarding kids for head-shotting each other is maybe not the feel that we want to go for. It might actually push us out of the E for everyone and into the-- what's the next rating, T?

**AUDIENCE:** And E--

**PROFESSOR 1:** Yeah. There's like an E-10 rating. At least there used to be. OK.

And sometimes, you've just got to cut mechanics because you don't have time to do them. They're great ideas. They would actually make your game better, if you had time to be able to polish them and make them understandable and smooth and everything. You just ran out of time, and you just want to be able to ship. So that's a perfectly legitimate consideration.

So that's this process of, like, generating ideas. Say, this is like an x or something, the number of ideas that you've got in your game. And mechanics in terms of either rules, and in a computer game, there'll be code. In these games, that's just the rules, it's pieces, it's boards, it's everything that you actually have to make your game work.

And you kind of go through this brainstorming phase where you get a ton of ideas, right? You've got some good ones up here. You've got some crappy ones down there. Actually, it's probably more like that. A whole bunch of crappy ideas.

And you've got a couple of good ideas. But yeah, there's a big explosion of ideas, where ideas turn into experiments. Some of them are thought experiments, that you're thinking-- just discussing them among your team-- what if we did this, what if we did that, how would you solve that problem.

They may be real, testable experiments where it's something that you can play, something that you whip up within an hour or half an hour, and you actually put it in front-- we've been doing that all semester. Some of the ideas may just come up off different ideas interacting-- and just, what if we had dogs in our game? What if we had solar power in the game? What if we had solar-powered dogs in our game? And things like that. You know, and then ideas just happen.

However, it's important-- so far I've sort of been describing the brainstorming process that you've all been through already. But one thing that we haven't necessarily asked you to do nearly as much is to come up with actual discrete experiments. You know, it's like, here's an idea. Actually come up with a criteria of success that you're looking for.

How do we know whether this particular idea, this particular experiment version of the idea is succeeding or not. And you're going to need that. Because you're going to need to cull. You're going to need to like reduce this back down, try to keep this, or maybe even build this up a little bit.

Actually, it's probably more like, you're going to cull a little bit. Then you want to bring this up again. And then each time this goes, you have kind of an increase in that you'll have an increase in bad ideas, but a decrease of bad ideas. But your total number of good ideas just keeps going up.

And by good, I'm using a very vague term. It could mean polished. It could mean actually good for game play. It could mean something that's been proven in a marketplace, something desirable that you want to keep in your game.

**PROFESSOR 2:** Things that could be-- like, it should be good for the core experience, but one of the problems could be, they are just-- they're good working mechanics, but actually don't help the core experience, or detract from the core experience.

**PROFESSOR 1:** Mm-hmm. Yep. Let's see. There are a couple of dangers, even in this early brainstorming process. So I'm just looking at this whole part here.

One is that you pretty much already think you already know what game you're going to make. You've got a single solution. Say you're taking the computer game version of this class and you decide that you're going to make an RPG. And your model is *Final Fantasy 6.* OK, so everything is going to basically follow the *Final Fantasy 6* template. So you're basically pursuing one single solution.

And the problem with that is that your implementation ability of actually making all of those things fit that template is already-- you are not Square. Right? You don't have all of the resources that Square had, even back when *Final Fantasy 6* came out. How old is *Final Fantasy 6?* 1994? 2?

**AUDIENCE:**    I don't know.

**PROFESSOR 1:**  Something like that. I mean, even then, they were a large team with many years to work on-- at least a year to work on that game. I don't know how long they actually worked on that game. There's a very, very good chance that your ability to actually even implement something that you know how it already works may not be able to meet that. So you will have a failed experiment and then you're kind of stuck. You don't know what to do.

So you're going to have to do lots and lots of experiments. But experiments are expensive. Everything takes time. Everything is going to need culling at some point in time. But you need a certain amount of time that isn't going to end up going into the game-- this chunk. Actually, it's a good chunk.

Which is why we've been trying to teach you how to prototype really, really quickly. It's a little bit of [INAUDIBLE]. Because we want you to try to be able to get on to experiments, come to a conclusion about whether that idea was good or not as fast as possible, so that you kind of reduce the cost of doing a single experiment.

And yes. It's true that you're not going to be able to conduct an experiment on every single idea that you've got. So you have to kind of pick and choose a select number of ideas to be able to take the experiment to the experimental stage.

Now, there are a couple of things that you can look out for to say, maybe we don't want to conduct an experiment on something. One is a brittle idea. And a brittle idea is that you kind of have to kind of twist it really, really, really hard to be able to make it fit with any idea that you've already got.

I'm assuming that you've already got a concept that your team was kind of building around this concept. Somebody comes up with an idea and a huge amount of twisting and turning to be able to even make it fit close, even though you think that on its own, this idea is a really good idea. You might actually want to see how many different ways could this particular idea be executed before you actually decide on the way-- on an experiment to conduct on that idea.

Two ways that you can do that. One is that you just specify, what is the goal of this possibly brittle idea? It takes a lot of twisting. Why do we need this idea in the first place?

Is it because the challenge in the game is too low? Is it because players have too much information or not enough information about what they're supposed to do? Is it because you want to introduce a little element of chance or maybe give some sort of strategic decision making?

Try to identify, what is this problem that you're trying to solve in the first place with this kind of unwieldy idea? All right? And once you've been able to identify that, then you come up with, all right, what are all the different ways that we could address that problem. As opposed to, here's this one solution that's kind of unwieldy, that might very well solve that problem, but maybe there are alternatives.

Has anyone encountered this in your own prototyping? Maybe in this assignment, maybe in a previous assignment. Like, somebody had an idea that was just kind of hard to use, but then what the idea was trying to solve was actually a good problem to solve?

No? Not yet? I thought I saw some nods. If you've encountered it, I'd love to get an example from one of you.

I can throw out examples. But I'd love to be able to hear examples from one of you. No?

**AUDIENCE:** I guess in our first game, we realized a couple problems with the game. So we came up with various ways of trying to resolve it, that didn't really work for the game itself, but solved the smaller problem.

**PROFESSOR 1:** Can you be more specific about what the problem was?

**AUDIENCE:** Yeah, I'm trying to think back to specific ones. One of them was that the way our game was with the set of the tic tac toe, it was really hard to see what sets were available on the board.

**PROFESSOR 1:** Right. This was the set assembly game. Sure.

**AUDIENCE:** Yeah. And so one of the things we tried to do was removing the center tile altogether, so that you could only play around the edges. Which did solve the problem of, now you could really see what everything was, so it was a lot easier in that sense. Except it was just a lot less interesting of a game, because there was a lot less strategy to it.

**PROFESSOR 1:** So you had a version of the game where all the squares were in play, including the center tile, and it just made it confusing for your players. And then you have a version of the game where there's just, like-- this one possible solution to this problem is just take the center tile out. But that made it trivial, or-- yeah.

**AUDIENCE:** So we tried to find a middle ground. And that's how we ended up with the idea where we had one single card in the middle. Because you could still use it, but you didn't have to worry about the lines of play going through the middle, and how incredibly complex it is to try and reason about that.

**PROFESSOR 1:** OK. So it sounds like, instead of a brittle idea, you actually had a good, robust idea. Which is, that center square is really problematic. It has hope-- it has possibilities. It has really interesting gameplay possibilities. But for learning the game and starting up the game, it's problematic.

So you found a way to say, all right. How do we deal with the center square, instead of the solution of just delete the center square. You know, let's find a different way to work it in the game. And your final game actually does that.

Something else that you can do is, if you've got an idea for your game that's kind of unwieldy, and you're not quite sure what to do about it-- discuss it among your team, but don't experiment on it. Don't work it into your rules. Don't try to implement it. Just get into everybody's heads.

Because that idea may come bubbling up later. And little bits of it might be applicable and slightly easier to implement as you make progress in your game. So in other words, put it on the back burner. But make sure you have actually had a chance to discuss it, so that everyone's kind of thinking about it, even if it's not something that you've decided that you actually want to do right away. It might be a lot faster to just have a quick discussion in that case than to conduct the experiment.

Yep. So those are a couple of strategies on how do you sort of focus your experimenting time. And this is probably not going to be very useful for assignment two, but it is going to be useful for assignment three. Because in assignment two, hopefully you're well past the creating and adding new features part and now you're just trying to get this out of the door for Wednesday.

So let's talk about this stage now. Well, yeah. In fact, I'm going to focus on this one instead. The sort of, like, cutting phase. Where you're both cutting good stuff and bad stuff.

But you're cutting stuff in general, because you have to. Maybe because you're running out of time. Maybe because you are trying to get rid of the bad stuff.

So, things that you need to do-- and this is something that I'd like you to do today, when you meet up in your teams-- is to determine what is the criteria for which you're going to cut a feature. For instance, come up with a quick description about what your game is supposed to be all about. You may have already written this in your rules. It might be in the very first paragraph of your rules, saying that our game is about getting entangled with other people, you know, or pushing them over, for instance. Which is it? And being able to identify it.

And then, when you look at all of your rules in your game, you can actually determine, OK, which chunks of our mechanics actually serve that goal, and which ones don't. You know, there are things that are there just to keep your game going. Those may be fine to keep, but those aren't directly contributing to how good your game is. Those are just basically there to keep the thing working. And maybe they can take a little bit of improvement, but they're kind of suspect.

And there's things that'll directly contribute to your core gameplay that you're trying to achieve with your game. And those are probably things that you'll want to spend a lot more time making sure that the rules are absolutely clear, and your players understand what they mean, that if there's any other mechanics in your game, they're supporting that-- you know, that those meet your criteria for what should stay in your game. Because once you know what your criteria is, it becomes a lot easier to be able to just look at everything that you've got, deciding what to keep and invest more time in, and what to cut.

Now, there's a couple of problems that people typically run into when it comes to cutting stuff out. A couple of game developers refer to this as culling. If you come from 3D graphics, culling is a very common way just to be able to just cut all the stuff out that you don't need.

So, for instance, if you don't have an explicit culling criteria, then you don't actually know why a certain mechanic exists, and you don't actually know whether you should keep it or not. So that discussion should be something that happens today. In the end, what is your game about, and does your game actually meet that?

You might have other culling criteria that aren't that. Does this game run in five minutes. What was the limit for this one? 10 minutes, I believe?

**PROFESSOR 2:** 20.

**PROFESSOR 1:** 20 minutes? OK. So if the game needs to run in 20 minutes, because we as instructors gave you that limitation, then that's probably a perfectly reasonable culling criteria for you to use. What in our game is sort of contributing towards this, and what's actually getting in our way of hitting that target.

Let's see. Something else that comes into play, and that affects culling, that affects this phase, is when you conduct experiments here, but your experiments are not very tangible. They're kind of floaty and wavy. This usually happens in teams that like to talk out all of the ideas, but don't actually like to do a lot of hardcore prototyping and playtesting.

I hope that that's not so much of a problem for anyone in this class by now. You've got in a lot of practice in that. But you may have a couple of rules and mechanics that have ended up in your game, that you never actually prototyped, that you've never actually tested out with real people. Make sure that whatever actually ends up in the version that you hand in on Wednesday is something that you've actually tested-- is something that you've actually gotten some real data about whether it's working or not.

This one happens with every student project. Students-- game developers in general always assume that adding more stuff is better. It takes a very, very, very seasoned game designer to realize that sometimes simplicity is good. And again, I hope this is not a terribly new revelation to the people in this class.

But I see it in student projects all the time. I see in student projects where they assume that more features and more complexity is going to be better. And in fact, I saw some of them in assignment one. And we're going to be talking a little bit about that.

More complexity doesn't necessarily always make for a better game. It may make it a more interesting intellectual problem. It doesn't necessarily always make it into a more entertaining game.

Let's see. One thing about culling is that cutting features and cutting stuff out kind of takes practice. You've got to do it fairly regularly. For assignment three, which is going to be starting on Wednesday, right-- we actually cut that on Wednesday-- what I'm going to ask you to do is try cutting a feature every week. You know, just say, what's on the chopping block this week?

Every Monday, or every Wednesday, or if you have a regular meeting time on Saturday or something like that-- just, like, what is the least useful thing that we've got in our game right now. And just try that. Because you can always add stuff in later.

Remember, this is a cyclical thing. You're going to be adding, then cutting, adding and cutting, adding and cutting. I think you've got in a lot of practice on the adding part. But you've got to practice a little bit more on the cutting part now. And you've got time to be able to do this for assignment three. You've got a little bit more time on assignment three than you had on assignment two.

So, start setting yourself a schedule, of all right, we're definitely going to cut something. I don't know what it is, but we're going to meet up as a team and we're going to decide that. And if it turns out that it was the wrong thing to cut, you can add it back later. But that allows you to focus your efforts on the things that you've deemed as more important for a short amount of time.

It can be hard to be objective about this process. Something that we've done-- in mostly our video game development, but you can do this in paper prototypes-- is to hand people very, very simple survey forms that basically say, how fun did you find this? How fun did you find that?

Say your game has different roles. You have a tinker, tailor, soldier, spy. These are your four roles in your game. And you just hand out the survey form-- how fun did you find being a tinker? How fun do you find being a tailor? Just one to five.

And that isn't terribly much information, but it gives you just enough information to help you concentrate your efforts on what's the most problematic thing in our game right now. You can go into greater details, like do you think that the game was too long, too short, too challenging, too easy, too luck-based, required too much thought. You can put these things on these five-point scales to be able to give you a little bit of feedback.

They don't necessarily tell you what the problem is. But it tells you where the problem might lie. And that will help focus your discussion and your efforts a little bit.

Yeah. And finally, the most important thing is that you've got to keep doing this over and over and over again. If you just do this once, you kind of end up in this situation where you've got a bunch of good ideas but the bad ideas are still there. I'm going to expand this for you.

What you eventually want to get to is a situation where the only stuff that's left in your game is the stuff that is worth keeping and that works well. And all the stuff that was bad, you've either completely cut it out or you've improved it to the point where it is actually a desirable feature. And you're only going to be able to get through that through multiple iterations. You're not going to get through that just by introducing something at the last minute and hoping that it works. Because you haven't had time to actually improve it and polish it yet.

Now something that's less of a problem for this class, but may be a problem in your other game design experiments outside of class, is that you may not know when to call it a day. You can repeat this process forever. And you've heard about games that have been going on for four or five years.

How long was *Duke Nukem Forever* in production for-- 15? At least 10. And that was actually one example of where new technology just ended up changing the foundation of which the game was built on, over and over and over again. It just kept resetting the game engine development and just took forever.

One thing that that game could probably have used is just some sort of standard criteria of saying, once we hit this criteria, we're shipping this game. It could be, once we've spent x amount of money. That probably should've been the case for that particular game.

But it could have been, once our informal internal surveys hit a rating of four out of five. You know, that might have been good enough. Some companies don't settle for that. There are companies that say, we're only going to release the absolute best games that we absolutely can.

But, don't kid yourself. Internally, they do have criteria on when these games have to ship. Even Nintendo does. And Nintendo's notorious for shipping late games. Because they have this mantra that a game is late for a short while and a bad game is bad forever.

So that's what they say to the public. Internally, they ship by Christmas. They get those games out to the market. Because they have a time box.

That's the other way to do it. And so it's like, well, we have to be able to get into the manufacturing process to print those discs-- less of an issue nowadays. But Nintendo still operates in disc fashion because they grew up as cartridge manufacturers. So they actually had to print it.

Now when it comes to publishing a paper game, that timeline can be even longer, right? Because you have to put ink on paper, you have to have it folded, you have to get all the plastic and stuff put in plastic bags, and put it into the box, and then put into pallets, and then shipped from China, usually, over to stores all over the United States so that they're in stores in time for Christmas. And that's, like, a huge timeline.

So that became the time limit. Sometimes you just set the time limit because you have to be able to meet ship dates. It's not very complicated. Now, when you're working on your own internal Kickstarter project or something, where you already know who's going to buy your game, because they paid for it upfront, before you even started manufacturing-- I think we'll hear more about that maybe in about two weeks, or about one week. April 9?

**PROFESSOR 2:** 9th.

**PROFESSOR 1:** Yeah. We'll have a couple of folks who actually work in the board game publishing industry talk a little bit about how they set goalposts and deadlines for themselves, what are the considerations for that. But in the end, it's just-- they had to set a goal, somehow. Either it's a quality goal or it's a time goal. And once you hit that, you have to ship.

Because you can always do-- if you have set a quality goal, you can always do better than that. The more time that you spend, theoretically, if you are sticking to this, the higher quality your game is going to get. But then you don't ship. You just have diminishing returns.

You'll never be able to move on to your next project. You're never going to get revenue from the project that you started. And your game may not be improved by that much more, just by spending extra time on it.

OK. So again, we're going to talk a little bit about things that are relevant to this from what we've observed from assignment one. And then for your team time today, start talking about, what is your culling criteria. What are you trying to achieve-- what's the simplest thing that you're trying to achieve with this game, and the simplest way to describe it. And then look at all the things in your game based on that criteria.

And whatever doesn't meet that might be something that you want to cut out, especially if it's already causing you problems. And maybe you'll just save some time-- free up a little bit more time to work on the stuff that's already working, and you can polish it up a little bit more. So any questions about adding and cutting features? OK.

**PROFESSOR 2:** So, taking all this and relating it to assignment one-- and then going into assignment two-- also starting with when you cut in assignment one. So, when it comes to advanced rules and basics rules, if you're going to do an advance and a basic rule set, make sure your basic rule set matches the core experience you're trying to make with the game. It's really, really important for assignment two, for this third assignment. Because that's what we said when set out, right, is we said, come up with something that's going to have a very particular experience that you want the players have at the end.

So if you're going to do advanced rules and basic rules, that criteria should be part of what your basic rules are. The advanced rules, as Philip mentioned yesterday, for some of them, it could add more interest. It could be more intellectually stimulating. But really, the basic rules should support everything the game has.

The advanced rules are just add-ons, actually. If you don't want to send us advanced rules, don't just cut it completely. Say, we're not going to use the advanced rules because we just could not get them to work in time. And that's absolutely, perfectly fine, so long as the basic rules still match the core experience you're trying to get across.

**PROFESSOR 1:** I think one thing I do want to add to that is that for the most part, your basic rules are pretty darn good. Much better than a lot of the advanced rules. Maybe because the wording in the basic rules had a little bit more care. Maybe you tested it. We found the basic games very playable and pretty enjoyable, was probably just fine for what we were expecting from assignment one.

So a lot of the stuff that you had in advanced rules, I do realize, some of those were things that came up in the testing process, and you thought, you know, this is a game that our game has evolved into. But once you realize that there is a core, basic game that works, and then that this sounds like a slightly more complicated version of the game, that slightly more complicated stuff is all candidates for cutting. Because you've already identified what the important part of this [INAUDIBLE].

**PROFESSOR 2:** When it comes to time and your deadlines, of course, adopt a common lingo. Adopt a common language. Stick with it. Proofread your rules. I think I have it written down in the very bottom. Proofread your rules. So I'll say that a third time later on.

And that means, like, naming conventions for the pieces, for what you've chosen to call things in your game. That could be naming conventions based on the rules, but also proofreading your cards, too. Making sure all your pieces are using the same language.

When it comes to talking about what the player does, we did notice some should versus may versus must, kind of getting mixed up. So when we're playing the game, we're not quite sure, well, if I should do this, if I may do this, why-- if it's may here, is it also may over here, or is it should over here? Just wasn't quite sure. Wasn't quite clear.

So I definitely recommend spending today working on your rules. Spend today having us play your game using your current rules, and then taking our feedback and then working on it for turning it in on Wednesday. And again, when it comes to cutting things-- when you are cutting, making sure that you are filling in the gaps. So if you cut something out, there might be a system, there might be a rule-- in this case, it generally was polish issues. So if something was removed from one of the rule sets, we saw a little bit of--

**PROFESSOR 1:** Remnants.

**PROFESSOR 2:** --remnants of it, spread in the rules.

**PROFESSOR 1:** So for instance, say you are one of those groups that had basic rules and advanced rules. And you took a chunk of the game that you moved into advanced rules because you realized that the core-based game was more playable without that. But your basic rules still refer to bits of the advanced rules.

And that gets really confusing. Because advanced rules are a completely different section of your rules that we haven't even gotten into yet. But we will. So do that cleanup work.

**PROFESSOR 2:** For setup-- so if you're going to have a setup section before your rules-- it's actually highly recommended-- have a setup section that is split out that is different from your rules section, so we know exactly how to set up the game. Include diagrams and photos. Where do players sit? What is the orientation of cards?

If you have a card that has all this information on it, what is that common information across all cards? If you think about a *Magic the Gathering* card, and all the different icons-- hopefully your cards aren't as complex-- but you'll have the same thing in the same place on all the cards, so you can then tell us what that thing is, and why it's useful, and how to use it as you're playing the game. And in particular, let your game teach us how to play, if possible. If you've got time. If not, it should still be in the rules.

What I mean by let your game teach us is, if you're going to have a board, and the board has a lot of empty space on it, you can put some your rules into the board. You can put some of the setup features into the board, if you've got time right now. Proofread your rules.

And turn in as many copies of your rules as you have players in your game. So, the assignment did say, it's a set number of two, three, or four players for your game. If it's a four-player game, give us four copies of your rules.

That's all the basic stuff. Yeah. And then we'll be getting your feedback, specific feedback for each individual game, tomorrow. Yeah. Yeah.

Tomorrow. I'm not sure what time tomorrow. But we'll get you that before you're actually turning in. That's it for me.

**PROFESSOR 1:** I just want to clarify one thing about should, may, must. I'm not sure, because I'm trying to think of a more concrete example. So it's like, if in your rules, you say, the player should move his or her piece from one square to another, we're not quite sure whether you're describing this as a good strategy-- as something that the player should generally be doing because it's going to put the player ahead-- or something that the player must be doing because the rules require the player to do this thing. Should is a very, very difficult word to figure out for us. And if you're to begin--

**AUDIENCE:** Would you suggest not using the word should?

**PROFESSOR 1:** It's possible. Sometimes context explains it for you. But it does muddy things more often than it helps.

I find should makes perfect sense if you're going to give an example. If you're going to give an example, make it [INAUDIBLE]. Make it obvious. Or even, like--

**PROFESSOR 2:** Put it in a sidebar, or--

**PROFESSOR 1:** Right. Do indents, or a sidebar, or something, to make it clear that this is not actually part of the rules. This is an example of how this set of the rules plays out.

And then you can use words like should. Right? You know, player one should do this. Because you're describing a strategy. You're not saying that the rules require this to happen.

Must-- say must when it's must, if a player must do something. And when you say a player may, be very, very clear of what else the player may do. Say, for instance, it's one out of five different options. You say, the player may do one of these five options-- bullet point, bullet point, bullet point, bullet point. And the last bullet point is probably, like, do nothing. If a game allows something.

**PROFESSOR 2:** Bullet points are really, really useful if there's any kind of option. Even if it's just, like, a player may do this or this, break up the or in its own line. Give it bold. Put a couple lines around it. Put the next thing that they can do-- the other thing they can do-- after it. Make it really, really clear it's one or the other, not both.

**PROFESSOR 1:** Yeah. Yeah. Yeah, that goes the same for not just your printed rule sheet, but also cards, stuff on the board. Because sometimes there are games where you have options on the card, right? And you may do this or you may do that. It's not entirely obvious if, say, the card does both of those things, or you get to choose from one of those things, and who gets to choose one of those two things.

So that's something to keep in mind. You only use numbered bullet points-- 1, 2, 3, 4-- if something's going to be sequential. It's good for things like, here are the steps of a round of game play-- first you do this, then you do this, then do-- do, do, do. Don't say the player may do one of the following-- 1, dot, 2, dot, 3, dot-- it's like, uh, now I'm not quite sure what's going on. OK?

**PROFESSOR 2:** Yeah. Yeah, and actually, this brought up rounds, stages, phases, turns.

**PROFESSOR 1:** That's the vocabulary thing again. Yeah.

**PROFESSOR 2:** Choose one. Stick with it. Proofread it. Make sure you chose it. Make sure it works. Especially if you're going to make any changes today, and you're turning it in in two days, have somebody who is not you proofread it and make sure you're using the same language throughout.

**PROFESSOR 1:** Rounds usually involves everybody around the table doing one thing in sequence, the same sort of thing, in sequence. Steps are really big. Phases-- it's usually describing large chunks of game. you know, things like opening [INAUDIBLE], mid-game, end-game phases, and usually not used in rules. But I know that some games--

**PROFESSOR 2:** Connect those games.

**PROFESSOR 1:** Yeah. Use those.

**AUDIENCE:** Sometimes they use other terms.

**PROFESSOR 1:** I know This is why it's maddening. Because it gets used in both contexts.

**AUDIENCE:** So what would you use to describe the individual pieces of a turn?

**PROFESSOR 1:** Steps. Especially if it's a sequential sequence of things. Like, this is my turn, in a round, and here are the three steps I get to do.

Step 1-- draw a card. Step 2-- choose a card from my hand. Step 3-- play it. Right? You know, things like that. That sort of works.

Phases-- I've also heard it described in things like there is round one is one kind of action, and round two is like a completely different kind of action, and round three is [INAUDIBLE] like phases. And it was like, oh my god. That's-- [INAUDIBLE] that, actually. Yeah.

**AUDIENCE:** In [INAUDIBLE], there are turns in phases within rounds in phases.

**PROFESSOR 1:** Yes. This is why the game is actually so much harder to learn than it is actually play. To play is actually not that hard. But when they desc--

**AUDIENCE:** The phases don't matter, but the turns and everything else was actually really, really good.

**PROFESSOR 1:** Yeah. So that's why phases complicate things. Sometimes it is, in fact, the right word, but it's just hard to be able to expect that every single player is going to interpret that word the same way. OK.

**PROFESSOR 2:** So you can break up into your teams, continue working on your games. This is playtest time. So, in particular, playtest your rules. If you want us to play your rules and give you feedback before you turn it in, today's the day to do it.

**PROFESSOR 1:** And attendance sheet--

**PROFESSOR 2:** Attendance sheet's right there.

**PROFESSOR 1:** Attendance sheet's over there.

[SIDE CONVERSATIONS]