

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR:

Schedule for the remainder of the semester. Today is a short week. Normal day today, slightly different day on Wednesday. I'll talk about that in the lecture today. Basically we hope to have the official class ended by 2:00pm. So 1:00 to 2:00 will be class on Wednesday. After 2:00, working in your groups or taking off early if you need to take off early for a flight or whatnot. So it's up to your teams, whether or not you're working in here or not on Wednesday.

This is also going to be the last lecture by us, I believe, on the schedule. Next week we've got two guest lectures. One, on December 1 we've got a panel from the Indie Game Collective. They're going to be talking to us about the business of making Indie games. Two people who have their own businesses, their own studios, and how their studios work together. So that should be interesting for you.

And then on December 3, on that Wednesday, we have Sean Baptiste from Fire Hose Games talking to us on marketing, communication, and community management for games. Because we're seeing you can't just put out a PR release and buy some advertising, you need to build a community if you're making games professionally. You need to find your players, because there's lots of games they could be playing besides your own. So he'll be talking to us about that.

The week after that is entirely based around your presentations. So December 8 is a rehearsal in class, live reversal with your game. If your game works on that date-- hopefully it will-- but definitely your slides and your spoken presentation. We will give you feedback, you will then take that feedback, improve your presentation, improving your grade, and give it as the final presentation on Wednesday, December 10.

At the end of the class on December 10 it'll be open play. So after all the presentations are over we're going to have some guests here. You'll be able to open your laptops and play each other's games and have guests play your games. So that should be fun and exciting, and maybe a little scary, which is good.

So plans for today then. I got caught on that-- not 5 minute, but 2 minute presentations. Basically short, informal presentation just like we did a few weeks ago, I believe. We'd like to know about the status of your game, what features you're working on, what features are not being worked on right now, what features might be dropped and cut, and I'd also like to see your game. So if you have got one person talking, maybe another person showing the game, maybe playing the game. This could be what the game looks like today. This could be the version of the game you tested last week, but have something show-able, something visual for us to see so that we can-- when you're talking about your game we understand what you're talking about. With no slides, of course, for this presentation.

For presentation on 8th-- on the 10th that is more formal, slides, all the good stuff. There's a description about that in Stellar and we'll talk more about that next Monday.

After that I'm going to give a short, about 30 minute, lecture on cutting features, some strategies on how to do it, why you should do it, things like that. Might be obvious, may not be obvious. After that is work in class. Pablo is here to give you feedback on your games, so we'll be giving you feedback on your presentations. If we have questions about what you just talked about, but also he'll be going around and available for each of your teams to talk about the status of your game and what's going on with your game.

That is class today. Any questions? All right. So it is 1:10. Are all the teams feel ready to give presentations right now or do you need a couple minutes?

AUDIENCE: Couple minutes.

PROFESSOR: Couple minutes? All right. So we're going to give you 5 minutes. 1:15 I'm going to come back up here and Snap is going to start.

TEJ CHAJED: All right. Hi everybody. I'm Snap-- I'm from Snap. I am not Snap. I'm Tej. And so we've made a lot of progress on our game and we've been able to run it a few times, but it has not changed a lot since probably anybody who's played the game has seen it.

So I'll just go ahead and play one round of the game just to show off what we currently have, and then I'll talk a little bit about what we're planning to do in the future. So everybody from the team, you want to play? And can somebody start the game?

AUDIENCE: I got it.

TEJ CHAJED: All right so the way this game works is we're going to throw out a topic. In this case it's animals. We're all going to try to enter as many animals as we can think of, and we'll get points whenever two of us enter the same animal. All right. And they can obviously all cheat, but let's-

AUDIENCE: Wait, we already started?

TEJ CHAJED: Yes, we started. Obviously I would not be showing this normally. This is a little bit easy to cheat on.

AUDIENCE: [INAUDIBLE]

TEJ CHAJED: Although we've had some other forms of cheating. When Andrew was here, they just entered A, B, C, in a row next to each other.

Anybody want to help?

AUDIENCE: Cow.

TEJ CHAJED: Cow. I guess this is also cheating.

AUDIENCE: Lemur. Snail.

AUDIENCE: Scorpion.

AUDIENCE: Amoeba.

AUDIENCE: Jaguar.

AUDIENCE: Chicken. [INAUDIBLE].

AUDIENCE: Pig.

AUDIENCE: Paramecium.

AUDIENCE: Veal.

[LAUGHTER]

TEJ CHAJED: Veal is not an animal. I guess baby deer is.

AUDIENCE: Geese. Human.

TEJ CHAJED: All right. So I think that gives you an idea of how this game plays out. Can somebody stop the game?

AUDIENCE: I got it.

TEJ CHAJED: So right now we have--

AUDIENCE: [? Yes. ?]

[LAUGHTER]

TEJ CHAJED: So right now we have the game completely manually controlled. And this is because current use case-- in our current use cases for the client we've been playing the game with a limited number of words input and without a time limit. So we let the client just implement whatever time they want by starting and stopping the game appropriately. But we plan on also supporting an actual strict word limit as well as a natural time limit that can be displayed on the front end.

We initially went with Phaser because we had a much different idea of how we would represent the game. But we ended up focusing more on words and, like, a more bare-bones representation. And so we're actually going to remove Phaser. Right now this is a Phaser window, but it's not really helping us at all. We're just entering words into a list. So we're dropping all of that and moving to just normal CSS, HTML.

We are also planning on-- we've experimented in our thought processes about what sorts of ways we want to show scores and activity. We want to be able to show not just what words you're scoring and submitting, but also when are other people submitting words, when are other people scoring, how am I doing relative to everyone else. So we have a couple ideas around that.

We also would like to show you a word cloud in real time. We currently have a word cloud for the back end. So this is the game we just played. And this is a-- and this is something that you can watch actually as a moderator, while you're playing, in real time. And then here's another game which is an actual game that the client ran in Berlin a week or so ago. So that was kind of cool that we actually were able to do a real playtest with the actual client in the real world.

But in the future we definitely want this to look more interesting and to feel more competitive.

We want people to be able to see what else is going on in the game, whereas right now you really can only see your scores. Any questions?

AUDIENCE: [INAUDIBLE] questions or comments, [INAUDIBLE].

AUDIENCE: What's your biggest risk right now? Like, what's the thing that's most catastrophic?

TEJ CHAJED: I think since we're switching the front end over to sort of new technology we need to make sure that a, that transitions over well, and b, that we have enough time to finish something new and useful rather than just reverting back to what we have now.

AUDIENCE: How does *Snap* happen?

TEJ CHAJED: So we have a server that is receiving all the words. And whenever it gets a word it adds it to a list of words for a player. And then every time it gets a word it also checks against all other lists.

AUDIENCE: So it's basically any other player types the same word?

TEJ CHAJED: Yes. In the same game. So for example, the game is now over. If you started a new game, that's an independent word list. Yeah?

AUDIENCE: Just for curiosity, so let's say you put in "dog" and [? Priam ?] put in "dog" and then Sam put in "dog"--

TEJ CHAJED: Everybody gets one point right now. We're thinking of changing that and letting people get more than one point, or maybe even making it configurable. Most likely-- intuitively it seems that you should get more if more people have entered the word.

PROFESSOR: Thank you.

TEJ CHAJED: All right.

[APPLAUSE]

PROFESSOR: Next up is Awesome Cholera. Yes, that is not the real title.

PABLO: While Awesome Cholera or other real title shows up, we will have a long conversation about details, but I'm very excited about the fact that it's already being used and working. And now the question will be how many features you want available to the [INAUDIBLE] player. And that

selection process will be based on both client need, playtesting, and also your ambition on how much can be accomplished in this time. [INAUDIBLE]

STUDENT 1: Do you want to do the talking?

STUDENT 2: Yeah. So this was the last build to the game we playtested. We hadn't changed a lot. We tried to add instructions so that people kind of knew what was going on. And this is running, right? Can you click Start? So right now this is what people saw. they saw their villages, and some health bars. It was still kind of unclear.

And over here you can see, like, the text is still really small. And when people clicked something to do education or something they either thought that, like, when they clicked that button it instantly did it. They didn't know that was a button-- like Educate About Soap was a button they could click. Also we had this really terrible pop-up bug that we're still trying to fix, but hopefully that'll be fixed.

And so because we were having so many problems with the GUI we finally sat down and tried to redesign it. And so I haven't actually-- we haven't done this yet, but here are some really great drawings-- A plus, I said-- of what the new GUI will have. So above each locality we're going to have this little box. I kind of stole it from [? Civ ?] and a few other places, but-- So it's going to have the name of the locality. And it's even worse because it's going to look different from what this is because we've already talked about how we want to change it.

And so behind the work locality is going to be a bar, and the bar is going to have a percentage green and a percentage red. And the percentage green will be the percentage of people in that locality that are healthy, and the red will be the percentage of people that are unhealthy. Above that-- sorry. Above the word Locality, there's a little arrow that represents the infection rate, which is something-- a lot of people used that word and thought that was the information they were getting, but they weren't. They were getting the percentage of people infected.

And so a lot of people wanted to know how's the infection rate being changed. And so we're going to have an arrow. And right now it's just going to be, like, up if it's positive, down if it's negative. But we want to also change it so that they can tell the magnitude of that as well. But we're also trying to keep it so that people aren't getting too many numbers thrown at them. So we're still trying to iron that one out.

Below the Locality are a bunch of symbols representing the different things you can do. What

we thought we were going to do is if you've implemented them they'll be there in color, and they'll be counting down so you know how much longer that will be in effect. And they'll also have a gold ring around them or something to say, like, if you've educated [? them ?] about it which means it's more effective.

And then the other screen is the screen you get when you click-- there will be a little thing that says Aid Locality above the Locality thing for each of the different localities. And over [? on this ?] screen-- [? which ?] [? got kind of ?] messy-- but it'll show all the things you can do instead of just the ones that are unlocked, because a lot of people didn't know that things unlocked over time. So we'll have it showing everyone, hey, these are all the things and they'll eventually unlock, either over time or if you do a certain thing.

And there will be two columns, one for educate and for prevent. And as you mouse over everything on the right-hand side of the screen, the description will come up telling you about the effectiveness and the duration, and giving you a description of what it's doing. We're hoping that this'll make it more clear as to what everything's doing, and it'll be easy to read chunks.

So if they want to know more about it they can mouse over it and just kind of read it. Or if not they can just kind of quickly say, hey, this is pretty effective but the duration's short so I'll just use this for a bit because it's cheaper and then maybe I'll use the thing that's even more effective and has a longer duration when I have more money.

The other thing on the map that I didn't scan yet is, on the map, on the very top of it, we're going to have a bar that tells you about what day it is and when you're getting money, because a lot of people were like, why am I getting money? And they didn't know time was passing.

So we want to make all these things clear. And so I think the GUI redesign will help with that. Hopefully we should have it done after Thanksgiving. But yeah. That's where we're at right now. Any questions?

PABLO: I have one, but you guys first. Anybody?

STUDENT 2: Yeah.

AUDIENCE: Is your goal for people to win this game and then have done the right things? Or is it for them to lose the game and then, based on that know what the right thing they should have done is?

- STUDENT 2:** Right now the game's really hard, and it ends up you kind of lose if you just leave it alone, which is not what we wanted.
- STUDENT 1:** As you saw, earlier, it just went to game over.
- STUDENT 2:** We want them to realize the different things they can do. I don't think it's going to be superbly challenging right now, is what I'm imagining. Because you don't want people to do these things and then be like, oh, these things obviously don't work because I keep losing the game. We want them to know that, like, hey, if I do these things it will help. And so that's what we're trying to get across.
- STUDENT 1:** Another idea I think that we're not communicating effectively but [? is ?] [? an ?] idea we also want to communicate is that the position on the map is important. So like if the village is-- upstream villages-- stuff you do there compounds to the downstream villages just because, like, the water, you know-- if you're not treating the water correctly in an upstream village that water is going to be contaminated down to the downstream village. And at the end we also want to communicate through this game is that, like, if you focus on the upstream villages you'll have better results in the long-term because--
- AUDIENCE:** So I guess how would I discover that? Is that just playing the game many, many times?
- STUDENT 1:** Exactly. So that's something we have yet to communicate. And we should figure out a way to communicate that to the audience.
- PABLO:** So thank you. Good luck reconfiguring the things that need to be reconfigured. My intuition is that you would be better off [? amputating ?] choice, and leaving somewhere very visible [? other ?] that says, these are other things that could be done but are not represented in this game. Because otherwise it will be just too [INAUDIBLE]. And unless you make a really awesome game that people want to play and play until they've tried everything, it can be too intimidating.
- STUDENT 1:** OK.
- PABLO:** One of the options for your consideration-- because right now you have the options of soap and the latrines and vaccination and so on-- you could reduce it to the things that the player, the macro-decider could do in terms of public awareness campaigns. So it could be about [INAUDIBLE] the water, about washing your hands campaign and maybe one more thing. So limited to three things that are of the same kind-- you know, radio advertisement or sending

people to [INAUDIBLE]. And maybe less villages.

Again, just reduce choice. Reduce features so the gameplay experience is more engaging. I just want you to know this is also applicable to the other Cholera team. The Ghana team is now really interested in promoting hand washing because it's beneficial both to cholera and to Ebola, which is a serious threat in the area.

So you are [? already know ?] doing what you're doing and if hand washing does not fit in the game, that's fine, OK? You are doing what we asked you months ago, not what we say now. But if you find a way to make hand washing fit as one of the choices, it's much better than having three vaccination choices and two moving waste choices and stuff like that.

STUDENT 2: OK.

PABLO: Good. And good luck.

STUDENT 1: Thank you.

STUDENT 2: Yeah, thank you.

[APPLAUSE]

PROFESSOR: *Hello Waves.*

STUDENT 3: Our game is called *Hello Waves*. We don't have the title screen in yet, but the idea is still forecast-based financing. So it's a little washed out on there. But if you can see there are these five different sand castles with workers associated with each. And up here in corner we have a forecast of what the water level will be like in a couple days.

This is the water level right here, and throughout the game it'll rise and fall. And the idea is not to let your workers get drowned by the rising water. The other aspect of the game though it that you have a certain amount of supplies and whenever you move someone from their castle in order to prevent them from getting drowned, they will consume supplies from that stock. And if you don't have enough supplies then they all take damage.

So as you can see, right now everybody's in their own castle. If we click Next Turn the water level will change and we'll see that the forecast will update. We have a range of values on this forecast, a high prediction and a low prediction, and if you mouse over any of the castles you'll see a red line appear on the forecast just to help you orient what heights are on the forecast.

Or if you're looking at the forecast, what it is in the real world.

So you can see the water move. For these couple days right now the forecast is pretty low. If we see that's kind of getting close and we're worried about someone we can click and drag them to another castle, and on the next turn they'll move over.

You can also click people to toggle between building and gathering. It's very hard to see the status text right now because it's small. We threw that in quickly for right now. But he is set to building and these three are set to gathering. So we'll see that each turn these three will gather one supply each. This teddy bear will consume one, so we'll see supplies go up by two, and victory progress will go up by one.

What we have planned for the rest of the game is we've made a lot of progress in terms of the intuitiveness of the game, once you understand it, but right now, there's a lot in the game that isn't explained from the get-go. And so if I give that spiel to anyone they can play the game fine and they do pretty with it. But if you just open up the game, you're lost. So a lot of our work is going to be on the UI, of making things more self-evident and making sure that people can understand what's going on and what their actions will do without me having to stand there and tell them.

PROFESSOR: Any questions? All right. So I'm going to ask you again, what is the biggest risk going forward?

STUDENT 4: I mean, I think in terms of our technical implementation we've already got good playtests of our work done and we have, like, all of our major graphics in and things like that. So I think our major risk going ahead is not being able to explain the game properly. And so even if we have finished product it may still not be playable if we don't write good introductions.

PABLO: All right. So on that front think of incremental additional features. So the games with the super simple, even maybe too boring choices, but then a new choice arrive. So the learning be playing can be staggered in a way that is [? enjoyable ?]. Good. Good luck to you, and we'll talk more about choices later. I should say that this high versus low prediction is something that I wish scientists did, like you're doing. Because most people say one value, then it's not that value [INAUDIBLE] earlier that [INAUDIBLE] forecast. Good. Thank you.

[APPLAUSE]

PROFESSOR: Heat Wave, come on down.

STUDENT 5:

So the main things that Heat Wave is focusing on right now are playability and education. As you can see this is our start screen. It doesn't have any instructions. It's very lame. We're completely redoing that. We're also redoing our end screen and our forecast screen because we want visual consistency and just overall improved playability by making the gameplay states more clear and more connected to each other.

The other thing we're working on is we don't currently have an explicit end condition. You kind of just play for however you want to play, and we're going to add in like a something happens that eventually too many people faint, get heat stroke, the game's over, so you have more of an incentive to play well.

And then finally something we're working on is user investment. So we are actually-- where's my mouse? Sorry. We are currently working on redoing pretty much all of our character graphics because we think that having better character graphics will make the game more intuitive.

So one of the ways you play this game, if you haven't played it before, is you have a bunch of characters-- I'll show in a second-- and you click on them based on different characteristics to try and offer them water or tell them, maybe you should go inside, it's really how out today. And different people will react differently to you.

And right now you have a bunch of characters-- where's my mouse? And I'm going to play this. They all look pretty much the same. And we have labels--

[LAUGHTER]

Uh, you're already laughing. Did something--

AUDIENCE:

There was a flare.

STUDENT 5:

Oh, yeah. So I'll talk about that in a second as well.

[LAUGHTER]

So we wanted to increase emotional investment in this game. So one thing I talked about was an explicit end condition. The other thing is-- we know that setting on fire is extreme and, at this point, kind of funny. We all just laughed. It's not meant to scare you, but it is meant to make you care.

Before, we just had people disappearing. And, number one, people didn't know that people had disappeared on the screen. Players were like, what happened? Suddenly there are less people. And then they'd get to the newspaper scene and they'd say seven people have fainted. Nothing. So we added the fire just as a somewhat comedic element, somewhat just emotionally investment element. I don't want my people to get set on fire, you know? So that has to do with playability.

And also the same thing with the graphics and the fire, is education. We wanted to make this stick with people long-term and we thought that maybe having, like, a more surprising reaction to them not doing anything would make it stick with them.

Something else we're working on [? with get ?] education is we need better and more dialogue. So we redid the backend of our system for dialogue, and it doesn't-- uh-oh. Sorry. I'm really bad at this. We redid the backend so that it would support better choice trees. We haven't actually added the content yet, which is something we're working on right now, but we have the whole backend working so that you can make decisions that actually really do effect what happens next. Something people complained about throughout is they felt like their decision weren't having an impact. That's going to be a lot more clear now.

And then finally we're implementing on setting up longer-term choices. So right now you can click on people and you might talk to them and they might not accept the water, they might accept the water. You might have a discussion with them and make them drink. But we wanted to have longer-term choices.

So what if you wanted to set up a public water fountain or something like that? And another thing that we're talking about is setting up umbrellas and it might be worth more of your time to just, you know-- the forecast for the next day it might say it's going to be 108, but today it's only 80, so let's set up a water fountain today so that tomorrow all the people can just get water and be a little healthier.

So that's the two things we're focusing on, playability and education. And we still have a lot of work to do, which is obvious, but we've done a lot of the backend work already and we think it's going pretty well. Any questions?

PROFESSOR: Questions anyone?

STUDENT 5: Yes?

AUDIENCE: [INAUDIBLE]

STUDENT 5: Oh, Julia did this. Isn't it pretty?

AUDIENCE: That's dope.

AUDIENCE: Thanks.

STUDENT 5: It is dope. I agree.

AUDIENCE: [INAUDIBLE]

[LAUGHTER]

STUDENT 5: Any other questions?

PABLO: Thank you. Quick comments. I agree with the idea of making something like the flame that is visually compelling. You may want to think-- if that's what you end up, fine. But beware that the idea of fire is very much elicits a hazard, something external, whereas the nature of the heat wave risk is a combination of the hazard like the fire and the vulnerability of the player. How susceptible are you, which is different from an elderly asthmatic person versus a healthy sports person.

So if you can brainstorm about how to depict that, which can be more evocative of the fact that it depends on the person and not just some external thing.

STUDENT 5: OK.

PABLO: I have no idea how. Maybe it's not doable, but if you can, that would be [? an invitation. ?]

STUDENT 5: And even if we can't do that we're going to try to implement that same concept through the dialogue by making the dialogue very specific, based on the person.

PABLO: Excellent. Great.

STUDENT 5: Yeah, you had a question?

AUDIENCE: Yeah. We found the fire funny. Do you think everybody would find it funny?

STUDENT 5: So you have to think about the audience for this game. The audience for this game is the Red

Cross worker. Now I don't necessarily think that all of them will think it's funny, but they'll definitely elicit some reaction, and I think that's what's important. Because before in our game we weren't eliciting any sort of reaction, so it wasn't memorable.

The fire might be funny to some, it might be scary to others, thinking about people going on fire, I don't know. Me, personally, I think it's a little terrifying. But either way it elicits a reaction. That's really what we're going for. Any other questions?

PABLO: One minor comment.

STUDENT 5: Yes?

PABLO: Most of the rest of the world uses degrees Celsius as opposed to Fahrenheit--

STUDENT 5: Oh.

PABLO: So consider having a choice for [? that. ?]

STUDENT 5: That's a great suggestion. Thank you. We'll take that into consideration. You had a question?

PROFESSOR: And your biggest risk?

STUDENT 5: Our biggest risk is-- so we're not adding any features, but from our product backlog it's just we're redoing a lot. So just the sheer amount and the breakdown of work-- it's just we're redoing a lot.

PROFESSOR: And that's content work, or is it front and backend work still?

STUDENT 5: So we redid a lot of the backend already, but in terms of the adding of objects, that's more backend. But in terms of the dialogue and stuff, that's more content. So we have work on both sides.

PROFESSOR: OK.

STUDENT 5: Thank you.

PROFESSOR: Thank you.

[APPLAUSE]

Next up is Saving the Animal Village.

STUDENT 6: OK. So I'm just going to start talking while Justin sets up. We are a subset of Saving the Animal Village. And our game is trying to teach cholera prevention through a series of minigames, three.

And then the story is that the mayor needs help deciding whether or not cholera is what is plaguing the city or if it's this monster that you encounter in the beginning. And so the player will go through, explore the town, and try to convince the mayor that it is, indeed, cholera.

The most prevalent things that we need to fix, I guess-- the prevalent problems that we need to fix are-- when we playtested with Pablo some time back he noted that our minigames-- for example, the filtration minigame-- wasn't really something that would connect to our audience, which were children who were in affected areas, like Ghana, from the ages of 8 to 13.

And we had this, like, factory setting before and it was not something a child would encounter ever. So what we're changing it to is, it's going to be inside a home and a mother's going to be the one that's bringing the water. And it's going to be basically things that children that will be playing this game can relate to. And we're still in the progress of changing that.

And we're also in the progress of adding conversation UI that's functional and that we can use while not alienating the lower range of our age range. Because Pablo mentioned that the children in Ghana [? who are ?] 8 years old might not be able to read as well as we want them to. So we're going to have more visual instructions as well, which is important. And that will probably be the deciding factor of whether or not this game will be successful.

I would say the thing that is keeping us back the most right now is the art, which we have to synthesize to match Ghana culture and communicate our message as well. We're also attempting to implement a way to integrate the minigames more through our main scene, but we'll see how much progress we get through [? and ?] [? we'll have ?] [? to correct ?] that feature. Any questions?

PABLO: Just one clarification. When I referred to matching, I like what you just showed way better than what was there before. And I think if this is the end aesthetic of that minigame, that's fine. My inclination is to think not so much matching to Ghana culture, but making it more adequate or suitable. So if it looks like science fiction to a Ghanaian child, that's fine.

There must be someone from Ghana around the Boston area. Send an email, talk to taxi drivers, I don't know. Just like you're doing playtesting with kids, try to find someone from West

Africa, if not Ghana so that they can tell you, no, cats are actually spiritually poisonous or whatever they may say. I have no idea. But you want to get some kind of feedback. I love cats. I hope they're not spiritually poisonous.

Good. And then there were many, many games. We will converse later, but this core idea I think is a very useful one. I'll be keen to learn more about how the game can be adjusted in terms of speed of learning and the speed of-- as you master it and it becomes more complicated because of things moving faster or some kind of additional complexity. But good. Well done.

PROFESSOR: So biggest risk? Are you going to say it's art? You mentioned the art.

STUDENT 6: Yeah.

PROFESSOR: What about the art? Is it the UI art or the representational art that we talked about?

STUDENT 6: Yeah. So it's mostly making sure the representational art aligns with-- what we think of it aligns with what our intended players think of it. And also just, like, we have so much art to produce. Like, getting it done.

PROFESSOR: The gameplay is there, all the buttons work, it's just getting stuff onto that?

STUDENT 6: Also the conversation UI, but our biggest risk, I think, is the art.

STUDENT 7: It's mostly functional. There's a few bugs to iron out, but the structure's all there.

PROFESSOR: OK. Last questions? Thank you.

[APPLAUSE]

All right. Any generalized feedback you want to give the whole class?

PABLO: Yeah. You're now in a place where you have on one hand more to do than probably you think you can, but also more temptation to do more things, new things, explore new ways of representing, et cetera. It will be a balancing act. In order to get a passing grade you have to deliver something that works well enough. I'll let them define that. I invite you to remain inquisitive as to those simple-to-implement new ideas that can help something be really better.

So don't try to do something really better that require starting from scratch. That'll be suicidal

at this stage. But that balance is a balance that you will be experiencing throughout the rest of your lives, whether it's designing a video game for the Red Cross or designing, you know, rockets. Good luck with it all, and we'll have more personalized conversations after the half hour presentation. Thank you.

PROFESSOR: Thank you.

PABLO: And I'll be outside. You let me know when.

PROFESSOR: We're done? Do you guys need to take a-- should we take a break? Should we just keep going?

AUDIENCE: [INAUDIBLE] 10 minute [INAUDIBLE].

PROFESSOR: Yeah. So we're going to take a quick 10-minute break. It's 1:50 right now. So 2 o'clock. If you want to talk about and synthesize the feedback we just gave you, do that right now.

On the presentations you just gave. You did really well. You knew the problems that you had, which is good, because I was worried on playtest that you might not know some of the issues that you were seeing in the playtest. So that's good. We're really glad to see that.

The big challenge I see for your future is planning for fixing those-- responding to those problems that you know. There's just a lot of unforeseen stuff that can happen between now and the end of class, and that is what my talk is going to be about, is how to plan for disaster. And part of that is, just don't have the disaster. Cut it from the get go.

So let me just make sure this is up. Next. All right.

You have deadlines. This is the creative process. Deadlines happen at the end of the process. Hopefully work happens before the process. In this class your deadline is the last day of class, right? Double check your Stellar, you might be surprised.

In the real world some of your deadlines are going to be demo days. E3. Anybody heard the term E3 demo? And soul-crushing-E3 demo, and studio-closed-after-failure-of-E3 demo? Basically, show your game off to a bunch of press and hope it works. Usually that means don't allow any press to play it. You just play it for them. And by playing it for them, that might be just showing them the video while you manipulate the controller. You don't get to do that in this class.

Other terms you might hear from the industry. Going gold. It's not just for disco. It is-- the Gold Master there on the left side, that's a game that went gold and actually was canceled. It got all the way to Gold Master, and then they said, no, we're not going to ship it. We're not going to produce it. It's just not good enough. That can happen.

Ship dates. Back in the day games came in these things called boxes made out of cardboard with printed graphics that were in interesting-- Eidos would make them in these trapezoidal things so they wouldn't quite fit on the shelf. So you would see-- that's an Eidos product.

Thief-- were you on this one, Thief, The Dark Project?

SARA VERILLI: Yep.

PROFESSOR: Yep. *Tomb Raider* as well. Again, you don't exactly have a boxed ship date, but you have a give-us-a-playable-version-of-the-game ship date.

So how much time do you have left? December 10. You have 17 days left between now and then to finish your game. Well, let's assume you don't work on Thanksgiving. Anybody going home to see family or anything? A good portion of you.

Are you going to work while you're with your family? Maybe you're going to try. Maybe you're in a turkey coma, or veggie turkey coma for me. But what about the presentation week? So December 8 and 10, you're actually in class working on your presentations. Hopefully your game works by the start of class on December 8. Hopefully.

We're actually going to recommend that you do not change your game on December 8 and December 10, and you do a thing we call "soak," which I'll talk about later. It's totally up to you to do it. I offer it every year. It's almost never taken. But I'm offering it.

So actually, was that 13 days or was it really 30 hours? You've got 12-- this is a 12-hour-a-week course. I took out the time that you're actually spent here in lecture, that I'm stealing from you. Sorry. I like talking sometimes. So you really have less than 1 work week to get all this stuff done. So think about that when you're thinking about your projects. Can you do all this in one regular professional, less than 40 hour work week? So that's scary.

How do you meet your deadline? Do you crunch or do you cut? Who's heard the word crunch before? Just raise your hand if you've heard it. All right. Raise your hand if you do crunch right now, if you are actually in the middle of crunch, not even on this class but in other classes.

Yeah.

As students, you already know how to crunch. You do this. This is your daily life. And it's expected in some cases, it's some problems with the system, guess what? The same problems in the game industry. There is a way to do crunch-- to do a viable crunch. And that's actually what next week is probably going to be for you, is your crunch week. So ways you can do crunch and ways crunch is viable in the actual, real world industry.

You spend a limited time period in crunch. So if you hear about death marches, that's not a limited time. That can be months, years spent working 80-hour days on some studios, unnamed, in Australia.

We're recommending spending one sprint. If you do a planned sprint we say, in advance, that we are going to crunch on this week. That means we're going to spend that week picking up the slack and getting through a bunch of stuff and we're going to work a long period of time. And it's going to be awesome. Because guess what? It's voluntary. It is team-driven. We decided, as a team, to do this. The management did not say, you are going to crunch right now.

And in fact, a lot of game industry would say, that's not even crunch. You're having too much fun. That can't be crunch. For them crunch is compulsory, unpaid over-time. Three words that I despise.

So if you are going to crunch and it is voluntary and you plan for it in advance, you would follow it by deep soak. Deep soak, in this case, means test, play, itemize bugs. It means you are playing your game, you are understanding every subtle nuance of the user experience of your game. You probably have a very good understanding of the programmer experience of your game, of the artist, of the producer, of the person on the team. You know how your game ticks from your point of view.

But you don't yet know it from the players point of view, even after focus testing, even after seeing other players play it. You probably haven't played your game enough to really understand what's going on. What it actually does from the get go. So you play it a lot, you itemize bugs, you prioritize bugs, you write bugs down, but you don't actually fix them. You just see what they are, see how bad they are, decide whether or not afterwards you want to fix them or not.

All right. So that's how you do crunch. I give that to you. You will probably do it. That is OK. Good luck.

But, advanced course. If you want to cut features, this is how you cut features. So from now until December 8, ask yourself, can you ship your game right now? That's part of the aspect of sprint, right? At the end of a sprint you should be able to ship your game. It's a viable product. I can send it over the wall, someone can play it, and it is good and ready to go.

In order to do that-- and in this talk I'm assuming that, in order to do this, you're going to need to reduce the scope of your game because you need to ship it soon.

So what can you cut? Some developers say these are the only things you should cut. Features that are not fun and features that do not advance the game's goals. So play the game a lot. Decide whether or not those features are fun, collect evidence for it, have a theory about what is fun or what is not fun, identify those features that go towards that theory and remove them. Just don't even continue working on them. In this case you're thinking about the game from the idea of your player and your stakeholders. These are basically features that if you spent more time, it would not be time well-spent.

But there's some other ones. There are features that are too big to be finished in time. Really take a realistic look at your product backlog right now and decide if you can finish it in time or not. Could you cut those features? Just give it a moment's thought. Think about it.

Features that just don't work. Why bother fixing something if it's just going to stay broken in two weeks? Can you cut it? Can you just remove it? Does it change the game or not? Can you even fix these in time anyway? You just have to look at it and you have to analyze it and decide.

So how would you actually go ahead and do that? Hopefully you can say yes to these things. Is your product backlog currently prioritized right now? Did you estimate your features well? So did you estimate the size and the time it takes to create the features? And did you create good blocks of work in your sprints? And my guess is, probably not. And that's OK.

We talked to you about Scrum, we talked to you about Agile. The thing is, Scrum is not a process. Scrum is people doing a process. You could be the best person, the best programmer in the world and be put onto a team, and still have all the problems that you're having right now because the team doesn't know each other that well.

That's one of the things we try to do on the first couple projects, is get you used to working with the other people in the class so you can find people that you've compatible work styles with. So that when you're in project four you have a chance at forming a good, solid team and having some team dynamics.

And remember when I talked about team dynamics earlier in the semester? Would anybody say that they are in the performing mode? Like, do you feel like you are performing well? Do you have an engine going? Do you see tasks get done? Do they just fall off the list because, oh, that was done. That was verified. That was awesome. Has that happened yet? Probably not. That's OK. Yeah. A lot of head-shaking. That's OK.

So we need something else we can try out, because that's not going to work. And even in the ideal world, you're still going to have to cut. Like, developers cut features every day. Those are the tools they have to use them, but when those fail we have other tools.

So based on what I read from your project three, some people think cutting features means dropping features. That's slightly different. It's not exactly the same. The difference between a cut feature and a dropped feature? A cut feature has intention. You've planned for it. You're going to tie up loose ends around the feature after you've removed it from the feature list.

You'll actually even delete the feature from your product backlog, and feel really good about it, and not have to ever worry about it again because it's gone. It's no longer going to be in there. Dropped features are those things at the bottom of the backlog that just don't get into the game at the end and could cause major issues if they're connected to other systems.

So how do you have that intention? You think through the dependencies. Scrum does not have a good tool for identifying dependencies. It's actually one of its weakest points. It assumes that, if you do all the other stuff your dependencies will just fall into place because you've created really good backlog items and you've organized them well.

So this is one of those things where, if you had a couple more weeks maybe you would want to switch to waterfall or go into a Gantt chart or figure out a way to graph every little thing that's going on in the game. And I'm going to say, don't do that. But think about that and get in that mindset.

So take a look at your code and backlog for dependencies. And here's one really easy thought experiment to take. look at your biggest unfinished feature and just think what would happen if

you just didn't do it. So in one of the games I saw-- this team here. One of these teams. You two teams over here. What's your biggest unfinished feature right now? Either team. Somebody just say it out loud. Maybe Matt, while he's chewing. Or somebody else.

AUDIENCE: Our time to gather all the clues in our endgame.

PROFESSOR: Time to gather all the clues in the endgame. That's your biggest unfinished one? What would happen if you just didn't do that?

AUDIENCE: There would be no purpose to the game.

PROFESSOR: OK.

[LAUGHTER]

So is it still the same game? Not really. Would the other features feel its absence? Hells yes. All right. So trace your dependencies, take a look at that thing, go down the line and see, what all the other systems that are tied into it? This is where graphing could be helpful. A little mind map, node map. If I wasn't crunching and actually made this presentation this morning, I would have had that graph.

But can you identify the loose ends? And then if you've cut those loose ends off of that feature, is your primary feature usable, understandable, and polished? You don't have to do this for every single feature on your feature list, but identify one or two really big ones and just do a little thought experiment and see what would happen if they went away. Because they're going to go away at some point in the next week and a half.

So you've identified a feature and you're going to say, we're going to cut this feature. We're not going to drop it, we're going to cut it. What exactly does that mean? So think cut plus. So cut the feature, replace it with a different, smaller feature. Key word there is smaller. Based on new knowledge you know your game better now than you knew it when you first made that product backlog. That fits a similar purpose.

This is an OK solution. It takes a little bit of time and you actually might be too late to do that right now. An example could be, you've got a first person shooting game and you have rocket launchers, but you don't have animated missiles. Cut the rocket launcher.

You have-- what else do you have? You need another weapon. You need another big

weapon. Maybe you have a ray gun. Maybe you can already animate things on top of another character. So you have a ray gun that if you press the button, anybody in front of the button has a little animation that vaporizes and dies. You've got a very similar feature going on. It's a cool weapon. It's damage heavy. You don't need to do that animated bullet, that animated missile, and you save yourself some time.

And you haven't hurt the game. The player doesn't even know that that wasn't even there. You didn't give the player a rocket launcher that didn't fire missiles, and just made people randomly blow up. You've given them something different and new and exciting.

Next one, cut and redesign. So take the same feature and then shrink it. Redesign it for a smaller scope. Again, based on new knowledge. I'm playing the new Pokemon right now. Pokemon, that really outrageous, complex, matrix of strength and weaknesses. When you break it down it's rock, paper, scissors.

I'm trying to make a game and I want to out-Pokemon Pokemon. I'm going to have modifiers for everything, I'm going to have even longer lists of different enemy types. Maybe I just need to go back to rock, paper, scissors, do a thought experiment, create a short little prototype. That's one thing you can do right now is you can create paper prototypes on your own. You've got big enough teams that one person can say, let me do a really quick paper prototype of that feature to see what it would look like.

Let me look at all the previous features we have, make sure it matches, and come up with a little paper version of it and see, is that going to fit the same thing or not? And if it does, great. You've saved yourself a ton of time. [? Today in ?] content, as we saw in the presentations today. Content is ginormous. Churning through that is really hard.

Lastly, cut and amplify. Take an already developed feature and make it better. So did you promise 32 guns for your shooter? Could 20 just be enough? And then spend that time you have with those 20 guns to make them better and more polished. So I already went back to guns. That's just where my head was today.

If you think, if you had one good, really, awesome weapon-- double sawed-off shotgun in *Quake*, maybe. That was pretty cool. I used that one a lot. And then a bunch of poorly developed ones-- BFG. I don't even understand why they even put that in there. Was that in *Quake* or was it just in *Doom*?

AUDIENCE: *Doom.*

PROFESSOR: But either way, though. You rarely saw it. It was cool for one second. Maybe if that was a hugely important, expensive thing, maybe cut it and the player doesn't even feel it.

So in all three of these things when we're cutting the feature what we're thinking about is, what does the player see? Does the player have-- what did I say? Expected feedback. If you drop a feature might the player still expect some feedback that they're not getting because that feature's no longer there? If you cut it the player should never know that it wasn't there.

Here's a game where they dropped features. This is one of our favorite examples from 2003, *Big Rigs*. Where's my mouse? Anybody played it?

[VIDEO PLAYBACK]

-Hey, kids. Strap yourself in for some action-packed racing. It's *Big Rigs*. 18 wheels of thunder, and we got trucks. Yeah, trucks. *Big Rigs*. Off-road traction. More power for non-stop driving action. *Big Rigs*.

PROFESSOR: It's *Big Rigs* off-road driving. You can drive off the road.

-Who knows? *Big Rigs*. Never lose a race again. You're always winner with *Big Rigs*.

PROFESSOR: You always win, every time you get through it. There's no timer.

-Phasing molecular mechanics to pass through solids--

PROFESSOR: Whoa!

- --so as not to interrupt--

PROFESSOR: Is this the future?

-Nothing stands in your way when you're *Big Rigs*. Rear-spinning tires with warp-drive velocity for inter-dimensional exploring. Leave the game behind and exceed the boundaries of existence!

PROFESSOR: This game shipped. This game was put on a disk, was put in a box, was delivered to your local game store and Walmart. People bought it, I think.

-Big mother-fuckin' Rigs!

-*Big Rigs*.

[END PLAYBACK]

PROFESSOR:

Yeah, that was advertised as the trailer. I quickly found out it wasn't, but I kept it anyway. Humor. Oh, where's my mouse? No! [INAUDIBLE] speaker notes. [INAUDIBLE] back. Apologies. There. [INAUDIBLE] here, down there.

All right. So that is the ultimate expression of dropped features. And-- oh, and I forgot to copy this. All right. So another example of cut features. This is *The Sims 4*. This is a quote from the executive producer, Rachel Franklin.

Basically what happened was, when the game shipped-- well, right before the game shipped, players identified that through whatever they did-- I think through watching community forums, listening to what the developers were talking about the game-- they identified 89 features missing from *The Sims 4*. 89 features that were not in *Sims 4* when it shipped. How did they know about those features? Well, they were in *Sims 1, 2, and 3* but they didn't put them in *4*, so they should be in *4*, right?

Basically, she says "I want to put everything we've ever created in the last 14 years into a base game, but we simply can't. What we have done instead is to create an incredibly rich, robust experience. If we didn't build the core foundation but I gave you pools, that wouldn't be the right thing to do."

They didn't quite meet the player experience. There could've been some things that-- maybe there were some problems going on the production, but they decided that they still needed a core game that could work because they could always add some of those features in later. There are some people on the forums who say, no, that's actually not going to be working in the game engine because I'm the nerd on the internet knows everything and I know you can't do that, developer who actually made the game.

So that is a problem that they knew they were going to have. They know that they have it when they ship the game. They have to come up with a community response to figure that out. And actually, that's something we'll talk about next week when it comes to community management, is when you're creating this game and people find out about this game, how do you then talk them down? But actually, in the most part you'll be talking about bringing them

up.

All right. So we're going to cut some features. We're not going to drop them. So who decides which features are going to cut? Who decides what and when, and then who's going to implement it? How are we going to do this?

So the people who decide which features get cut usually are the stakeholders. In the professional world that's going to be that executive producer. For you that could be your client. That could be your product owner, if you have a product owner on your team, if you have someone you consider a vision owner on your team, it could be that person.

It could default down to producer or Scrum Master, if that's the person we're saying that we're following their lead, they're going to make that decision. It could be your game designer. But it's going to be somebody on your teams. One, maybe two people on your teams who are coming up at least with the what and the when. Not everybody on the team knows the schedule as well as maybe the producer does.

But the who implements and how it's implemented-- that's the entire team's job. This is why we have those long, boring Scrum meetings at the beginning and the end of each sprint. It's because that's the time period when we have all the team together to make these kinds of decisions.

It's definitely not the producer and Scrum Masters' role to decide how you're going to cut that feature. They're there to facilitate it and make sure that everybody who's attached to that feature can talk about it, can explain why, can give some argument, can give some defense. Rather than giving the entire team the need to decide everything, one or two people decide the initial part and then the whole team focuses on just how we're going to cut those things.

And that's because how you implement it requires knowledge of both code and content. It's more than a feature list. A feature list can't do this for you. It can do the top, it can show you the what, but never the how.

All right. Last little section. So you were working on a feature and your feature was cut. You spent a lot of time on that feature. Your roll was to implement networking. And this is actually an example from two years ago. But was cut. In that case, it wasn't cut because of these things.

I don't want to cut it. I want to see this finished. I put some commitment into this thing, dammit.

I want to see this all the way through. I feel guilty about cutting it. I really let my team down. I was just working on networking the entire semester, and it's not going to make it into the game. What have I been doing? I feel sad about cutting it. It's not fair. How dare you cut my feature? You're going to see a lot of that, if you haven't already. But that's OK.

One thing to keep in mind when you're cutting a feature-- and this is one thing that we told that person-- was, your work needed to happen because otherwise you wouldn't know that we needed to know about your work. You described this better, by the way.

SARA VERILLI: I'm trying to remember exactly what I said, but it's something to the effect of, sometimes the work that doesn't make it into the game had to happen anyway because you had to do the work to explore networking to find out if it wouldn't work or not. Just like you had to do the work to explore several other features that did go into the game.

But you didn't know that the-- I don't know-- you didn't know which of those features were going to work out well. But you were hoping that some of them would. Some of them did, some of them didn't. The fact that one particular feature-- the one that you worked on-- is not a reflection on you. Not at all. It's a reflection on the game, and the project needs to go forward.

PROFESSOR: Absolutely. So yeah. Work that gets cut is never wasted work. And that's actually a problem I have when game developers do their postmortems and post in the Gamasutra is they always talk about it as if it was wasted work. But really, again, they wouldn't have had that information had they tried it.

Work well-cut is exploratory work. It means that you cut it early enough that it was exploratory. You got your feelers around it. You figured out that was probably a direction you didn't want to go into, and you pulled back. And actually you did that in the beginning with your paper prototypes. I don't think your games now actually are one-to-one relationships to those paper prototypes you created at the beginning of this project. And you threw those away, and it was OK.

So why is this important? Because the team is more important than the product, which is, unfortunately, more important than you. But that's OK, because the team equals you plus others in the same situation. You are actually part of the team. So just keep that in mind. This is where I go to my *Goodfellas*-- was it *Goodfellas*? There's no I in team, baseball bat? You are on a team. Everyone's contributing to it. Everyone's doing good work for it. All right. so

that's my spiel on cutting features.

Here is my-- what I'd like to do on Wednesday. So Wednesday is going to be a short class. We're just meeting from 1:00 to 2:00. You'll have the remainder of the time to work in your group, in your teams you as you like, but if you want to take off, you can take off.

What we are going to do-- and it's going to be me, Phillip, and Drew, because unfortunately Sara won't be here-- is we're going to meet with each of your teams individually for about five, seven minutes tops. What we'd like to do is look at an updated vision statement. So between now and then take a look at the vision statement you turned into Stellar, compare it to the game you have right now, and modify it.

The vision statement is your publisher contract with us. That vision statement that you actually finally turn in on December 10 with your game, is the thing that we are judging your game by. If your game has all the features that are described in that vision statement, then aces. If you created a vision statement that doesn't have the features are in the game, or if you don't describe the features well in the vision statement then we're going to use that as an understanding of what could have gone wrong with that game.

This should be a really quick thing you can do. You can probably do this in five minutes today with your team. You have everybody here right now. You just had your presentations about your products. So if you can take five minutes today, reassess your vision statement, take a look at your product backlog, change it. We're going to talk about that.

You can re-turn that in to Stellar or you can bring in a paper version on Wednesday, so long as we can see it when we're sitting with you on Wednesday. That's all we ask.

And again, like I said earlier, plan as if nothing is being done over Thanksgiving, just in case. If work happens over Thanksgiving, cool. You did good. That's bonus work. So just keep that in mind. Any questions?

All right. Pablo is back in the room. And you're working on your teams for the rest of the day. Thank you.

[INTERPOSING VOICES]