PROFESSOR 1: So we want to maximize time today, both to make sure that we're doing-- all of the presentations get good feedback, but also so you have time afterwards to make some changes. So, if you haven't signed in yet, come on down sign up while I talk. Just get that out of the way really quick. Schedule for today is, we're going to probably be done with rehearsals by 2:30.

After 2:30 we are going to have Luke from OpenCourseWare interviewing teams and producers from teams about the class. So, we're thinking about how many minutes per team?

LUKE: Not too long. 10 to 15, casual chat.

PROFESSOR 1: Yep, casual chat is optional, but I would appreciate it if at least people who did production and Scrum Master roles would talk to him. But it could be the entire team talking with him, and maybe one or two people dominating the conversation. That's OK. The goal for this is for people who are looking at the OpenCourseWare site, particularly teachers who might want to use these materials in their classes, to find out from the student perspective what students got out of the class. Because it's kind of like student evaluations. We're going to be out of the room for that period of time. So that means after 2:30 we're out of the room, Luke's doing interviews, and your teams are doing whatever teamwork you need to do to get ready for the demonstrations and presentations on Wednesday.

PROFESSOR 2: Whatever you say to the camera, we will not even get a chance to see it before we give you grades. So you can whale on if you'd like.

PROFESSOR 1: Please do. Or at least, please be honest-- let's try for that.

I should also note, student evaluations are available online. Please fill those out. We're going to-- actually it's a great period of time from 3:00 to 4:00 in class today to just open up your browser and fill out your student evaluation, right here and now so that you get that out of the way. And again, that feedback's going to be really helpful for us. It's going to change how we teach the course in the next year. We change it every year based on feedback we get from students. So topics you thought you wanted to hear more about, topics you thought we covered too much, whatever feedback you have for us, do let us know.

Last announcement we have a guest here, [? Leigh, ?] from Odyssey magazine.

GUEST SPEAKER: It's a kid's science magazine for kids aged 9 to 14. And we're doing an issue devoted to gaming. And [INAUDIBLE] graciously allowed me to come and observe you guys as you rehearse your presentations. And I'm going to be taking some pictures, and I may ask some questions. I have some photo releases I'm going to need you to sign if I use your photos. So I'll pass those around. The good news is, if you don't want me to use a photo of you, don't sign it. Because I can't use your picture without it. So that's just an easy way to handle that. So, I'm just going to pass a pile of these around. And if you're willing to let me use your photograph, fill one out. Thank you.

PROFESSOR 1: OK. All right, so for our order of presentations just for today, not necessarily for Wednesday-- we'll figure out the actual order by the end the class today. We're going to start with Heat Wave, then Hello Waves, then SNAP!, then saving Gora Gora, then Cholera Control.

What we're asking you to do is come up, plug-in the computer that's going to be used to demonstrate your game and make sure your game works on the projector. The display resolution the native display resolution is 1280 by 1024. If you're doing a 4 3, or 1280 by 768 if you're doing widescreen. It shouldn't make a huge difference for your slides, but it might be a problem for your game. So, make sure you're coming up, doing a quick two- or three-minute check to make sure the game works. Afterwards, plug-in your slide presentation, and we'll just go step by step.

One thing I'd like to know from each team is how many speakers do you think you're going to have-- how many main speakers you have in your presentations. So, starting with Heat Wave, how many speakers do you have? One. Hello Waves? Two. SNAP!?

AUDIENCE: Two.

PROFESSOR 1: Saving Gora Gora?

AUDIENCE: Five.

PROFESSOR 1: Five. And Cholera Control?

AUDIENCE: One for now, maybe two. Not sure.

PROFESSOR 1: So, for Heat Waves, Hello Waves, SNAP! And Cholera Control-- actually for all of you, whoever is doing the most speaking, strap this little thing on. For everybody else, when you speak, make sure you are standing on the normal spot in front of this, so that the microphone can capture you. If you have-- for volume control, it is over here. Hopefully you've used this before. It's set at about midway. Take a note of all the different settings you are using when you're plugging in your stuff to the projector, so we can make things go fast on Wednesday. Any questions? All right take four more minutes in your teams to get yourself ready, and Heat Wave, please come up at 1:15 to plug-in your game.

AUDIENCE: [INAUDIBLE]

PROFESSOR 1: No, so time yourself with the clock. Take a note of your own time. We're not enforcing time limit right now. We're concentrating on content.

GUEST SPEAKER: Well, hello, my name is Mary [? from Providence ?] and today I'm representing team Heat Wave. We made a game about heat waves, not surprisingly. And these are the people on our team. I'll talk a little bit more about that in a minute. So, first of all, our game was about heat waves, and the main goal our game was to educate Red Cross workers about what to do when a heat wave was upcoming or was currently in progress. How do you prepare for it, and also what do you do to deal with it. And while we focused on these goals, our design goals were more specifically usability, play--

AUDIENCE: [INAUDIBLE]

GUEST SPEAKER: OK. How did I get that far without someone pointing that out to me?

AUDIENCE: [INAUDIBLE]

GUEST SPEAKER: OK, we're going to start over if that's OK. Do you mind? I really didn't get that far in, so that's OK. Heat waves-- oh, but now I can't see the slides. OK. Heat Waves, this is our team. Heat Waves is a game to talk and explain to Red Cross workers what to do in order to both prepare for heat waves, and also what to do once you're in a heat wave. And that was the purpose of our game. And while we had those goals for our game educationally, our goals overall for the design process were playability, usability, and education. Usability was first, because if the game isn't usable, no one's going to use it. Same with playability, if you can't play through the game, then it's kind of useless. And finally, our main goal was education, because we really wanted this game to serve a purpose, which was to educate people.

For our design process, we did the same process that most of the people in this class did. We started off with brainstorming. Then we went into team formation, responsibilities breakdown, and then just constantly updating, using Trello and different Scrum features. I'm going to talk about all of these in a little bit more depth.

So first of all, brainstorming, we started off right on the get-go, drawing on the board. What do we want this game to look like? What are going to be the main parts of it? And we actually settled on a somewhat complicated design, with the main screen a dialogue system, and then kind of an options menu.

And once we had this, we actually formed our team. This is our team minus-- I don't think Joe's in this picture because he was sick that day. And one thing you notice is we had a team of nine people. So it was a lot of people doing a lot of stuff. So we had to really deal with that. And one way we dealt with that was using Scrum. We had a lot of emails. But the way we managed that is, we learned from previous times that if you have one giant email thread it gets overwhelming, you don't see everything. So we separated it, had a couple different email threads based on different topics and subgroups.

And then also we updated our Trello. One thing we realized really quickly is that Trello is not so good for listing every single Scrum. So you kind of have to have a to-do, a done, and a working on right now, just for future reference. And that's kind of how we organized ourselves later on. And then just because it's Scrum, it's an iterative process, which is the picture.

Next, we had a lot of group meetings. And something we spent not as much time in class because we spent a lot of time outside of class. We had a bunch of meetings where we had snacks, and we would all just sit around for like three, four hours, and work. And those were really productive times for us, because it was a time where everyone who was there was really dedicated to fixing bug after bug, issue after issue, getting things done. And they were really what made our game come together as much as it did.

So, focus test, we ran four focus tests. And each of our focus tests had a different theme. So once we had our team together and we were building the game, there were different things working at different times. So for the first focus test, we had a low-fidelity prototype that with digital, and it was a Python game. And we've tested concept-- I'll talk a little bit more about that in a second. The second one we tested our initial unity game, and we looked for over-arching issues. The third test, we worked on visuals, just how can we emotionally resonate with our audience. And the fourth test was game balancing, because at the end of the day, once you have everything together, it's really easy to ignore if the game is balanced or not.

So the first test we ran was a concept of general mechanics. So as I said earlier, our general focus was education. It was our big goal, and we wanted to make sure the type of game we were building could really convey information and teach the audience about something. So if any of you played our Python test game, you would remember that you basically had a bunch of people you could choose to talk to. And if you talked to them, you could offer them water, tell them to go inside. And they'd either listen to you or ignore you. And you just went that way in loops until either everyone fainted or everyone went inside.

And what we learned from that was, first of all, it worked. People were noticing, OK, I'm going to focus on the drunk guy, I'm going to focus on the older person, try to get them water first. Because they faint more quickly. And the other thing we noticed right away is something we did try to try to make this game more lively, is we had fun dialogue. So for example, the first quote up there was like a more factual, drinking water even when you're not too thirsty is important in this type of heat. That's a very true fact about heat waves. You're just supposed to drink more water than you need to. And then the second one is just more fun trying to make it more lively, hide your wife, hide your [? kill, ?] it's heat wave. So that's something we experimented with early-on, having fun or dialogue to kind of increase the excitingness of the game.

So, the second focus task we really were looking for overarching issues, and it was our first unity-based game, so we are working with issues with our dialogue system, issues with how is the playability. Does this concept of interacting and thinking about education still work in this unity-based game, where it's less of like a, oh, there's just these old person or young person, which one do I choose? Instead now you have these figures. And we tested this out a little bit. We found the educational aspect was still coming through, but we had a lot of playability and usability issues. So, we needed to majorly revamp that.

So then we got our focus test three, where we focused on the visuals. So once we went through play test two, we realized, OK, we can fix a lot of this usability stuff. And now we want to see, can we make this game emotionally resonate with the players. And make it feel like less of just kind of like, I'm sitting there looking at this red screen and it's like, argh. So we had this, and what we were looking to talk to people about was the different characters, were they still prioritizing in this scene. And it turns out they weren't, because those labels didn't move with the people. And even if the labels had moved, people said the labels were too small to read, they were confusing-- really should just get rid of them, and instead either add little icons or caricatures of different types of people in order to choose who to help. So those were the kinds of questions we asked. And we actually showed them a sample artwork of a redone homeless person drawing. And people said that, that was a lot more informative and that way we didn't have to have like little text that no one could read. So focus test 3 was really about the visuals, and we really re-did a lot of our visuals based on that. Sorry, this mouse is a little non-responsive.

Finally, focus test four-- I think you all probably had one of these-- was just game balancing. The end condition of our game is basically when people faint, how many people have fainted. And if a certain number of people have fainted, then we stop the game. And the problems with that was that it was, at first, there was too many people, and everybody was dying right away. And then we instead of ramped up the number of people, and then there was an issue where, you were fine, you were fine, you were fine, and then suddenly everyone died on the fourth day. Because there were suddenly 100 people, and you could not help them all. So focus test four really focused on that, and trying to balance the game out.

Along the way, we made some bad choices. Every team does. And the main ones we made was, we decided to depend on external code. And while we thought it would allow us to make a more complicated game, we didn't really foresee how much we would have to adjust ourselves around the external code. It's a two-way street. And so it kind of impeded our progress sometimes. It added a lot more complexity, things that we weren't expecting. And then the other thing we made a mistake on, which I'd like to emphasize, is we focused on education, which was a great thing to do, but we forgot to focus on fun. And one thing we did to compensate for it, as I mentioned, is add that dialogue, that banter, that makes something that's kind of dull a little more interesting. But even so, I think one thing that if we went back and we did things differently, we'd focus on fun as a number-one priority, along with playability, usability, and education. So ours became a pretty standard learning game, as we talked about them.

Good choices, we had a lot of group meetings with snacks, great idea. Turns out people come if you have snacks. Just if you were wondering. And then we prioritize. So everybody prioritized. But one thing we had to deal with, because we had such a large team, was not everyone was available at the same time. So it was really prioritizing not only based on what needed to be done but even if something kind of had already been assigned who can actually work on this right now and get it done, and like when are they going to get it done. So we really worked with that and kept iterating and figuring out, based on what needed to get done right now, who was going to do it.

So the game itself is composed of really three screens but a couple of them have sub-components. So, the Start screen with an instruction screen; the main game, which has the daytime screen and a newspaper scene in it, which is like the end of each day you see that; and then the end screen, just you can take a look at all of these. So originally, this was our start screen. Main feedback with that was, it was not pretty, and it was not informative. So when we redid it, we added an instruction screen, which is actually still having a little bit of bugs with the readability. But what we did was we added a way for players who didn't play the game before to find out what they actually needed to do. And then the actual game, the original, we had-- I showed you on our focus test two and then our focus test three with the people moving, but the labels not moving. We really iterated the main game to make that as usable as possible.

And our final result-- well this is actually in progress. So one thing we updated in progress that I almost forgot to mention, was that we saw in focus test two and focus test three that visuals really important. And one thing we weren't displaying at all that was very educationally important, that Pablo actually mentioned, was that as people are about to faint, they usually get sick. If you see somebody who's about to faint, they'll like start sweating, and turn red, and all this stuff. And in our game, they just kind of keeled over and died. So, we needed a way to implement that. But we couldn't necessarily like draw a million different character art. So, we decided on the simple solution was just to add a little-- that little circle, like the thing above their heads, you can all see it with the exclamation point. And that was something we did in progress.

And finally we integrated that with the new character art, and this is the final game-- I don't know what those squares are, they're not in our game. But you can see the people are a lot more descriptive. I think that's pretty obviously that's an old person, they have a walker. And we had to rely on these caricatures of people, which really worked out in terms of like getting visual feedback when we talked to people. They were much more responsive to this than like labels or even like little symbols. They really liked the art.

And finally-- not finally, but the other part of our main game was the newspaper scene. And initially we just had a lot of information displayed, and no one actually read it, because they didn't think they could do anything about it. And they also just didn't like looking at it. It was confusing. So we re-did that, and so made it a little more visually appealing. And something we did that I didn't mention on the previous side about the main scene, is that we actually added more longer-term options. So this was more relevant as you were playing. Because if it was cooler on a day, you could start installing umbrellas. And then you would actually be like, OK, well, it's actually 115 today, so I'm not going to install any umbrellas, I'm just going to try to give people water.

And then finally-- so originally at the end of our game, as you can see, there was no end scene. And that's kind of important. So, people really wanted feedback as to-- they didn't just want the game to restart, they wanted to know how they did. Even if it's educational, part of that fun that was something we obviously didn't focus on, was just knowing, oh, I actually did pretty well. This many people-- like I lasted this many days, or this many people like went inside because of me. And that's because we weren't focusing on fun, we didn't even think about the importance of an end screen. Right? So, when we re-did it, we added this end screen, and it tells you how long you survived. So, two hours, five days, you can get a sense of how long can I keep it up and make sure everybody was working out well. So that's our whole game.

And looking back at this whole process, something we've discussed as a team that we would do differently, is passion. So one thing about heat waves we all thought initially, oh, it's such cool topic, heat waves, it gets hot. And then we read about it, and it was actually a surprisingly simple phenomenon. It gets hot, people drink water, go inside, and that's about it. There's not really a lot of complexity to the issue. And so a lot of us lost our passion very early on. And I think if we were going to do this again, we'd somehow incorporate something in addition to the heat waves that would make us more passionate about the game.

The other thing we'd do differently is accountability. Just making sure that-- with such a large team you have to really keep track of everyone doing everything. We really focused on prioritizing who was like actually had time to give and making sure they got tasked on-- which worked out really well. But at the same time, there were people who sometimes were kind of lost and didn't know where the game was going. Because we weren't always on the same page with such a large team, nine people's a lot. But it still worked out really well. And I'd say that given everything, we did pretty well with it.

And if we were going to do more features, something we talked about was more long-term options, people like those. They made it more fun, because you had more of an optimization problem if you had more long-term choices. Those were kind of hard to think of, because in heat waves there aren't that many. Like you can install an AC unit, and then maybe a water fountain. But we were thinking maybe we can make this a bigger scope game, add multiple environments so you have more long-term options. And really build up the complexity over time, so this game can be more realistic of a community as a whole instead of a single environment. And that's really what we wanted to do. Thank you so much. Are there any questions?

[APPLAUSE]

PROFESSOR 1: Oh, so comments about presentation. Anybody up there? Yes, so can you load up your game make sure your game works? Can somebody come down here from the team and do that? And you can answer questions while we-- so feedback from us then about the presentation. The main question I had-- like you covered a whole bunch of stuff, which is really good-- the things we're asking for, what went right, what went wrong, you did in a different format, but you got to it, and that was fine, that was good. The one thing I would have loved more information on, if you can fit it, is the external code issue. Specifically, what was the external code, why was it in there, what did you think it was going to help you with? And then what did end up like-- just a couple lines of those details about

GUEST SPEAKER: More specific.

PROFESSOR 1: More specific.

GUEST SPEAKER: Yes, definitely. [INAUDIBLE]

PROFESSOR 1: And really what is it giving you that basic unity wasn't giving you.

GUEST SPEAKER: OK, absolutely. [INAUDIBLE]

PROFESSOR 2: Let's see, I like the structure, as [? Rick ?] said, you got to everything. But actually the way how you structured it also works really well for all the points that you intended to hit. If we are following this particular order of presentation--

PROFESSOR 1: We're not necessarily, we're going to figure out what the order is after we do the all.

PROFESSOR 2: Whoever goes first will probably need to explain who Pablo is to the audience.

GUEST SPEAKER: Oh, sorry--

PROFESSOR 2: No, no, no, it's not something that you would necessarily have expected. But whoever goes first on Wednesday, if you talk about any of the clients, you have to assume that your audience doesn't know who the clients are. And different teams had different clients that they were talking to, too. So do just explain [INAUDIBLE]

PROFESSOR 1: We'll introduce-- basically we're making games about the topic, but your specific topic--

GUEST SPEAKER: If I have to mention Pablo, I should say who it is, or I shouldn't mention him directly.

PROFESSOR 1: Yes.

PROFESSOR 2: Yes, or just give his name and who he is, that will do wonders. Something for anybody who's using Google Presentations, Google Slides or whatever that thing is called, I think the Ask button brings up your notes. So if you have notes-- I wasn't sure if you were relying on--

GUEST SPEAKER: No, I just wanted to be able to see the slides. That's why [INAUDIBLE]

PROFESSOR 2: You also get a little preview slide. So if you hit the Ask button, you bring that up. Also it automatically brings up the, "Google Drive is using full screen, click this to make it disappear." And that's really distracting. So whoever is using Google drives, make sure you do that. And you may want to use the keyboard, because the mouse was not--

GUEST SPEAKER: It was flipping out.

PROFESSOR 2: Yes, it was there was not cooperating.

GUEST SPEAKER: I brought it for the actual game play, but we're not playing the game today.

PROFESSOR 2: OK, so recharge, and if it starts to give you that same problem, for the slides at least, you may want to just avoid using the mouse.

PROFESSOR 1: Actually, putting it in mirror mode, too, instead of two screens, would at least give you a full-screen version of the slides. The speaker notes give you a very tiny version of the slides.

PROFESSOR 2: Yes, it was very hard to see, actually. So mirroring your screen back and forth will also make it so that you don't every have to do that.

GUEST SPEAKER: Can you write that down?

PROFESSOR 2: I can email this actually, at least these notes. At the beginning of your presentation. If you can just-- when you introduce that your game's about heat waves, before you go into the things that you did, if you can just very briefly talk about what Red Cross workers have to do in heat waves.

GUEST SPEAKER: OK, yeah.

PROFESSOR 2: Yeah, of course, if that's after you demonstrate your game, then that may not be so necessary. But just to give people an-- just to make sure that before you start talking about this is what your team did, just give people an idea of this is what you're going to do in the game.

PROFESSOR 1: And you all have a choice of demoing the game before the presentation or after. It's whatever works best for your presentation.

PROFESSOR 2: So if you're showing the game first you may not need to describe this, but it if you're showing the game after your talk then you definitely need to explain that. When you say, "over-arching issues," as a bullet point, that tells me nothing. It's like, you had a big issue. But then later on of course you went into detail and talked-- oh, it was about dialogue, it was about learning, and stuff like that. I think the first time you introduce a bullet point, say over-arching issues, and you say, such as, the dialogue, bugs, and the learning, very quickly, just so that bullet point means something.

[? Rick ?] had already mentioned the thing about external code. When you talked about how you swapped your art assets, and you said, we talked to people about the art. Did you the mean play testers, or did you mean your client, or did you mean your team members? Right? They're all people, so which people? Yeah. I was assuming play testers based on the context. And finally, this has nothing to do the presentation. Does the game still use Fahrenheit?

GUEST SPEAKER: Can we change that?

PROFESSOR: What was the question?

GUEST SPEAKER: Fahrenheit? I don't think we have changed--

PROFESSOR: The game is still on Fahrenheit.

GUEST SPEAKER: It's still on Fahrenheit.

PROFESSOR: OK.

GUEST SPEAKER: We can change that today.

PROFESSOR: It's a bit risky to change it at this point. You might want to list that on your things to change in the future maybe, just add [INAUDIBLE].

PROFESSOR: That's enough.

PROFESSOR: Yeah, that's enough. I wouldn't touch the code.

PROFESSOR: Don't touch the code. Please don't touch the code.

PROFESSOR: Yeah. Just acknowledge that, because that was one of the big pieces of feedback, is that only the US uses Fahrenheit.

GUEST SPEAKER: Yeah.

[INAUDIBLE]?

PROFESSOR:

PROFESSOR: I'm not following, so he got [INAUDIBLE].

PROFESSOR: Sorry.

PROFESSOR: No.

PROFESSOR: So make sure it actually plays. So you're going to demonstrate it in unity rather than as a web hosted version?

GUEST SPEAKER: We might do a [INAUDIBLE] version, but we haven't had luck yet.

PROFESSOR: OK, cool.

GUEST SPEAKER: We're meeting after class.

PROFESSOR: Great. But it looks like whatever you're going to run it in, it's going to work at that resolution. And is there audio for it?

GUEST SPEAKER: Yeah. There is. But it's not playing right now.

PROFESSOR: OK. Oh, there's some.

GUEST SPEAKER: One of the things got lost, apparently.

PROFESSOR: So at least it's not muted.

GUEST SPEAKER: I'm sorry.

PROFESSOR: It looks like heartburn. It looks like it ate something bad.

GUEST SPEAKER: Thank you so much.

PROFESSOR: Thank you.

[APPLAUSE]

PROFESSOR: And thank you for being first.

PROFESSOR: Yeah.

PROFESSOR: I'll email you the notes.

PROFESSOR: "Hello Waves," come on down.

GUEST SPEAKER: So it's a little light on there, but our game is "Hello Waves." It's a game about forecast based financing that we're developing for the Red Cross. The idea of forecast based financing is using the idea of a forecast about the future in order to make decisions that are more effective than just reacting to disasters when they happen. So I'd like to show you a play through of our game. It should open up. Great. And let me turn on the sound.

[CLASSICAL MUSIC PLAYING]

So this is "Hello Waves." Here are the instructions. We would have done a tutorial if we had enough time, but this gets the idea across, and designing a good tutorial that actually teaches the player well is hard to do in a logical flow. So we have these instructions that explain how to play, what the point of the game is, and a couple of tips about how the game works. So if we go into the play screen, you see something similar to what we showed you a couple weeks ago, where you have all these different toys out there, castles.

And as you go through the days, you'll see a forecast of what the water level's going to be at over time. And so you can see that right now. They're gathering candy. The water level changed. This character is underwater, and so he takes damage. And so what you actually want to do is you want to move toys out of the water so that they don't get damaged. However, they can only move one castle over at a time. So because we didn't think ahead well enough, we'll see that this truck will actually take damage on the next term, because he's underwater.

Actually, he ended up not underwater. We got lucky there. But theoretically, he would have. And so as you go through it, the end goal of the game is to build a castle. And so every toy when they're home can either choose between gathering candy to feed evacuated toys or building the castle to reach your end goal of the game. And then the idea of the forecast is that you want to use the forecast in order to know how much candy you're going to need over the next couple of days and to use it to know when you're going to need to evaporate various toys from their variously hidden homes.

So back to the presentation. Full screen. So we had a couple of challenges in the design process. The first and the major one is really that we were trying to teach about forecast based financing, which was a bit of an abstract topic. It's a little different than just thinking long term, because you have to use the idea of there's some information we know about the future that we want to use to make optimal decisions, or at least decisions based on some idea of the risk that's out there. But we also wanted to avoid things like just pushing a button to win, where you have all the information you ever need, and there's just one option that you know you're going to pick every time. And the game has no thought whatsoever.

And we also needed to think about how we were going to communicate the forecast to the player so that they could then use that to make decisions. One of the problems that we also ran into related to this was that we focused a lot on the idea forecast based financing as the topic, and then tried to build a game built on top of that topic instead of building a game that used forecast based financing. So that held us back a lot in the beginning when we were trying to come up with ideas. We also had a really difficult initial target audience of policymakers-- 50, 60-year-old government officials, or people at NGOs who were going to be using forecast to make decisions of some sort.

And it was supposed to teach them about how they can use forecast to make these decisions, except this was a really difficult audience, because they don't generally play games. And they don't have a lot of time to learn about this kind of thing. And so we couldn't really expect to get them to sit down for a long period of time and play around with our game. In order to deal with these problems, we came up with a couple solutions. The first thing was that we had no idea what kind of game would work, or would make sense, or anything like that. So what we did is we just broke our team up into multiple groups and came up with a bunch of different prototypes.

On paper, we had two different prototypes. One that focused on the idea of managing a city, and its resources, and its response to disasters, versus focusing on individual people, and how you're going to move them around to keep them safe from the disasters. And then we went into a whole bunch of different digital prototypes-- one that was a text based game about managing a city. We had one that looked like this, where you had two different cities, and you were specifying what workers were going to do, or when they would leave the city in order to stay safe.

And then we would take the different ideas that we were learning from both of these prototypes, combine them together, take things out. And our final prototype actually uses ideas from all of these. Another solution that we got lucky with is when Pablo came to play our game, he told us that we actually shouldn't focus on the policymakers, because he wasn't confident that he could actually get them to play the game. And so we switched our target to being grade schoolers, which is why you saw the cutesy art with the beach and the toys.

This actually made it a lot easier for us, because we could target people who probably had some experience with games, or at least wanted to do something fun, and would be curious to learn about our topic.

GUEST SPEAKER: So for our development process-- or I can just speak here, right?

Yeah. You can speak right from there. That's fine.

GUEST SPEAKER: So another big issue, another set of challenges that we ran into was through our development process. And so we had initially issues with communication and facilitation. Our team had a wide variety of experiences and backgrounds. Some were hardcore gamers, some mostly mobile gamers. And so there were initially a lot of disagreements on what level of game we wanted to create, and what sort of game, casual versus hardcore that we wanted to create. And so we needed to overcome challenges of facilitation and communication within our team.

Another major challenge in our development process that we had to face came from our design issues, where for a very long time, we had a vague vision of what to do. We didn't know what kind of game we wanted to make, and so we purposely tried to keep our game ideas vague as we built prototypes. But then we ran into issues where we would have solutions. But no consensus on which solution was best, and where we went for long periods of time without having a clear direction of where we wanted our final game to be.

So our solutions for the challenges proposed by development were a team structure. And so we structured our team loosely into three subteams-- a production subteam, which would take care of production, like the deliverables, and making sure that all the game ideas are being communicated properly. A technical team, which worked primarily with the code and making sure the game got done, and then a user experience team, which handled art, UI, and sound. And so we kept our responsibilities flexible.

So as team members got busy over the semester, or as changing conditions, lots of different people contributing, responsibilities were able to easily flow between teams and team members. And additionally, another idea we started with in the beginning with the idea of sub-team leaders. [INAUDIBLE] the two people marked with the I's. But that was an idea we later abandoned in favor of just having a more flexible team structure, or a flat team structure. Another solution for helping our development process was the use of good code practices.

And I cannot emphasize enough that this really helped speed up our development, because we didn't run into trouble with code. It was mostly with design. So we used Yeoman, which is a JavaScript module system. And basically, it allowed our code to be interoperable. We can write one module separately from another module. So that solved a lot of issues with dependencies, or people working in parallel, because it allowed people to work in parallel without overriding each other's code.

We also used good code practice with state machines and NBC, which is motor vehicle control. And so we had a very object oriented code, very modular. And when we did need to change our code, rip it all out and put it back in, it actually didn't turn out to be too painful, because we just had to switch a couple objects around. And then one final solution that we used for our development processes were Slack and Scrum.

So Slack is like a modernized IRC chat room, and it's very feature rich. It has a lot of integrations with GitHub, and Google Drive, and things like that. And so that real time communication actually made it so that we didn't really have to meet outside of class too often. If someone was working, we would just email out saying, I'm on the Slack. And then people could meet on the slack. And it was full featured enough. We could send attachments and things like that.

Most of our communication in person communication could be done in class. And in class, we adopted a daily Scrum format, where we simply said what we had done since the previous class and what our goals were until next class. So in the end though, we did have to cut some features. These features mainly were the idea of again, a tutorial, or multiple levels, simply because they'd just be too much content that we need to play test in order to make sure that it was a consistent quality got our message across, and also, trying to add more individuality to the toys that you saw.

They have different graphics, but that's as much as we could do, given the time constraints. So kindly just bring it back. And our three worst decisions were first, we did end up spending a lot of time on code and assets that never got used. We maybe used 10% to 20% of our final work in our final project. Actually, maybe that is a bit overkill.

But maybe 30% of our final code and assets in our final project. This really was due to this second bad decision, that we kept the game and its direction too vague for too long. We always were holding out that oh, maybe we'll be able to come up with a better game idea. Or maybe we'll find some magic solution to how we can make forecast based financing into a game. And because of this mindset, we'd spent probably the first half of the project just staying too vague, and that hurt us in the end, because we spent so much time going in all these different directions.

And really, the decision that captured those first two though is the fact that we tried to make a game on top of forecast based financing. So we had forecast based financing. And we were like, how can we scan this as a game? Whereas once we switched that mindset and thought, let's have a game, and how can we put forecast based financing into it? I think that was the moment that we then came together as a team and really started making the final game that we wanted.

So our best decision.

GUEST SPEAKER: So one of our best decisions was that we chose good tools at the beginning, which meant that as we went through all these different prototypes, we actually didn't have to completely rewrite our game. We could pull out the way that workers worked. In one game, we could pull out the view that we were using in another game. And then we could just combine them together, and that allowed us to move quickly whenever we were changing our prototype.

We also weren't afraid to trust each other, both, in terms of what everyone was working on, but also, in terms of the decisions that we were making. And so when we said that we had to throw something out, we all understood that it was for the good of the project. And we didn't have a lot of complaining or hurt feelings when something didn't get put in, or when we decided to throw out our assets or code. And when we got to the end, we had been through enough of this vagueness that we were all frustrated with it. And we realized that we had an idea that we all liked, and we really got on board with it and made it happen.

Once we started working on our final idea, we saw that it worked. Basically, every decision from that point on was how do we make this game better? And we were all on board with that vision.

Thank you. Any questions?

[APPLAUSE]

[? PROFESSOR: Anyone in the ?] audience have [INAUDIBLE] first before--

AUDIENCE: We can't see [INAUDIBLE] across the [? room. ?]

GUEST SPEAKER: Yeah. We'll have to make all of our slides darker.

PROFESSOR: Yeah. Yellow and white.

PROFESSOR: Yeah. Like the yellow and white on your early screen, I was doing this on [INAUDIBLE]. I realize that's the game, not the presentation. But wow.

PROFESSOR: It's also a little faded in the game itself. It could be the display resolution you're using, so after class, after we're done with everything, try different resolutions. See if it makes a difference.

GUEST SPEAKER: Yeah. That's be great.

PROFESSOR: The music in the game development was too high. It was overpowering your voice. So I think you were almost hollering just to be able to be heard. It doesn't need to be that loud. So you can just crank down the volume right down on the computer or on the controls over there. Something that I had a question for. You don't have to answer it today. But you might want to put it in your presentation.

How long was the design of what you eventually established on the table before you decided that this was the thing you were going for? Because I'm assuming it was one of your vague-- it came up during your vague idea phase. And you're implying that you were in that mode for too long. But I don't know how long it was on the table, or was it only something you figured out at the end of the vague idea phase?

Because otherwise, you wouldn't have been able to switch to this idea if it was not on the table in the first place. And how did you decide this was it, that the same household game was going to be-- I understand-- I think you very clearly explained the benefits of deciding this was it. But how did you come to the conclusion that it was? That's stuff I wanted to hear more about.

But you don't have to answer it now.

PROFESSOR: Yeah. And just a little bit of specificity. You mentioned you spent too long on that design phase. I wonder how long that was.

GUEST SPEAKER: OK.

PROFESSOR: You said Pablo switched your target audience for you, telling us whether that was because of the game that he saw, or because of something else. If you know that information, throw it in there. If you don't know that information, that's fine. Knowing why you dropped the team lead-- you mentioned you dropped it, but you didn't exactly say why again. Really quick, we weren't getting [? blah ?] out of it, or the flexibility was more, whatever.

And defining your terms. You were saying things like casual versus hardcore. Give a little bit a definition of what you mean by that. It means different things for everybody. So what is your use of that?

PROFESSOR: I think he specifically said hardcore and mobile. But there are hardcore mobile players out there.

PROFESSOR: And casual can be considered a version of hardcore, just in a different way. So just be a little more clear, because I think you were talking about the target audience, and the kind of players, and the kind of games they might play. So just be a little bit more focused on what exactly you mean by that.

GUEST SPEAKER: OK.

PROFESSOR: That's it for me.

PROFESSOR: So we can start testing for observations in the young man who's not relaying [INAUDIBLE] module frame [INAUDIBLE]. It's just a system to generate the project, and it posts in the front [INAUDIBLE]. So again, just show that.

GUEST SPEAKER: OK.

PROFESSOR: So that's a terminology issue then.

PROFESSOR: Yeah.

PROFESSOR: Clearly, it worked out for your team, so we're not saying, don't mention Yeoman. It was clearly a good thing. But just check your definition of what it is, because it'll be more helpful for other people to understand what it is so that they can think about whether the want to use it in the future.

PROFESSOR: So your demo-- you spent about three minutes doing it. We are going to have a player from [INAUDIBLE] play your game live, without getting a lot of help. You can talk over it. You can, after a while, start helping them. We want to see them at least just start playing it on their own. Decide when you're going to put that demo in. You could actually combine them both together if you do it at the beginning or end. But if you do that, give the player a little bit of time before you start talking.

GUEST SPEAKER: OK. Sure.

PROFESSOR: That's all I got.

PROFESSOR: Great. Thanks.

[APPLAUSE]

PROFESSOR: Snap, come on down.

GUEST SPEAKER: Hi everyone. I'm part of the Snap team, and this Sabrina, who will talk about the front end aspects. I'll talk about the back end aspects. So I think we actually wanted to start with a demo of the game, and we'll do a real demo right now. So could everybody go to snapgame.org?

GUEST SPEAKER: You should turn on your volume, as well. It'll be interesting to see [INAUDIBLE].

GUEST SPEAKER: Oh, nice. My screen's hidden. And I will also play the game. Everybody ready who wants to play? So the way this game works is I'll explain it again, in case anybody hasn't heard this before. We're going to give you a topic, and then you're going to enter words related to that topic. Whenever you enter the same word that somebody else entered, you'll get a point, which we call a snap.

And the topic is algorithms. And you can cheat by looking at my screen, but please don't. I have no idea how I'm doing.

AUDIENCE: Oh, no. What happened?

GUEST SPEAKER: What happened? Oh. Oops.

AUDIENCE: We found a UI bug.

GUEST SPEAKER: I think it's this resolution. Probably. I did test this. I'm sorry. I never entered this many words. I can't scroll. I think we're going to stop this one. Wow. That was impressive. Justin won, followed by Rodrigo, and Rachel.

[APPLAUSE]

This is what you guys submitted.

[LAUGHTER]

This is a very MIT heavy. Domain.

GUEST SPEAKER: Someone put [? socks. ?]

GUEST SPEAKER: So that's our game. So the way we did this project was we split up into back end and front end teams from the very beginning. This was a bit of a challenge for new features, because initially, we both were really highly interdependent. But eventually, it became really nice, because it was simpler to manage a front end and a back end team individually, and they generally didn't need to talk to each other once the back end was off the ground.

I think that structuring our team with more individual roles, besides just having generally back end and front end people would have been helpful. UI design was something that we didn't think about early on enough, and that's why the UI is not as polished as it could have been. We could have had somebody just dedicated to taking information between the back end and front end, say, documenting the back end and making sure everybody knew how it worked. We also definitely had problems with paperwork, which would have been nice to have somebody who was really on top of that.

And then the game design, as well, was something. In addition to UI design, we also could have had somebody just looking at the game and thinking about the core game concepts, things like scoring and what we're going to display to players. We did a pretty good job with planning. We used Asana and Asana worked really well for us. It was quite flexible, let us manage independent task lists for front end and back end, which we think worked well all the way from the beginning to the end.

So separating out the task lists we liked, Asana we liked. Our project had a lot of dependencies, and this was something that Scrum inherently had a problem with, and our tools also had a problem with. It was hard to know, for example, when a task on the front end and back end were dependent. There wasn't a good way to encode that and see that the back end needed to get something done more quickly, because the front end could use it. We also had trouble just creating tasks.

I think that there was a diffusion of responsibility as to who would be creating tasks and who would be assigning them. Once the task got assigned, then it went pretty well. But we had trouble just getting all our tasks listed out. I think the core of all these problems was really getting people invested in the project and making sure that they treated it as a priority and really spent some effort on it. Want to talk about this?

GUEST SPEAKER: Sure. So we also used Slack for communicating as a team. And I think it was really helpful. I liked the fact that we could have multiple channels so that people working mainly on the front end didn't have to get all the communication going on with the back end. And then we also had channels for our Asana and GitHub so we could see what was going on with that. Some problems though where I feel like people weren't as active as they could have been-- I feel like maybe everyone should have downloaded the mobile app, or at least signed up for notifications so that everyone's aware constantly of what was going on.

And if someone mentioned them in a post. They were quickly able to respond and get back to them. And that's the main thing that I think we could improve upon if we use this in the future.

GUEST SPEAKER: So going a little bit specifically into the back end, what went right for us was definitely our technology. We used node.js as the server and socket.io to do the two way communication so that you could receive snaps from other players. And we deployed on heroku. All those three things worked really well. We had no problems with them. And the design of the back end also went pretty well.

It's a pretty simple server to implement, and we did a good job of figuring out what it needed to support. We didn't document as well as we could have, in terms of what is the published API that the front end should be using. So in socket.io, you send all these events, and the names and the formats of the events were not well documented for a long time. There wasn't a lot that I would change on how the back end was structured and how it went. But I think maybe we had too many people working on it for the small amount of work that it was, and it would have helped if people more people had worked on the front end.

That's also part of finalizing and freezing. If we had just finalized and frozen the back end, then we could've stopped worrying about it and just moved on. Networking was explicitly forbidden in the rules of the class, which we did break, and we intentionally broke it quite early on. Networking really didn't give us very many problems. We had a couple heroku issues, where we had to quickly do a deploy, or there was a bug, such as the one you just saw. But really, heroku went really well for us, and you could easily roll back.

And heroku gives you good documentation on what code exactly ran every single time. We actually have been deploying to the client. The client has been using this for a while now. These are some of the tests they've run. And you can see that they're quite different, in terms of how many teams there were, how long they went. I'm not really sure what the really long games were like. We still need to talk to Pablo about his feedback for those.

But indeed, there were words being submitted for those entire three hours. And just to give you a sense, I'd actually like to show one of these word clouds. So these are much more serious topics than algorithms, where actually, looking at the words will give you a sense of what people are thinking about a topic that you care about. So for example, I think I'll just include screenshots of those, because that doesn't work very well. So here, the topic of this conference is zero poverty, zero emissions within a generation.

And you can see, education seems to be a big theme in this conference. I can tell that just by looking at the word clouds, of several of them, actually. And a lot of our effort ended up being focused on the client, as opposed to the class. I think that's part of what made our paperwork fall behind, is that I've been working a lot on supporting these runs that happened in the past four days, as opposed to the deliverables of the class, because I thought that this was important.

It's part of why we have snapgame.org.

GUEST SPEAKER: Developing the back end, we were really successful. But I felt when it came to the front end, we struggled a little bit. I think a main challenge was the game is game that was intentionally made to play person to person, and have that communication. And representing that in a digital game was a big challenge that we had to do. So in our first design, we decided we were going to use Phaser, and it looked something like this. And obviously, Phaser wasn't even really being used, so what was the point?

So we dropped Phaser, and then we went to just using Bootstrap. Here, we're just using Bootstrap in JavaScript to handle all the input from the users. And then in our final design with as you saw, the dots and the drawing, we're using a JavaScript library called 2JS. So what went right was I think we had really good people to do UI design. But we didn't really begin thinking about that until later. And it took us a really long time to be able to represent the game in a really good way to the user. There were a lot of challenges we faced in doing that.

So I will come back to this. Some of the first ideas that we had for how we're going to represent the information to the player was by using a word cloud that we used on the end, but blurring out the words until they actually submitted a word. Then we would display it. But this is actually really, really challenging, and so we didn't want to spend our time doing something like that.

Then we considered having some sort of line where you're racing against the other players to get to the top. But this isn't entirely interesting, so we considered including both of them. But then again, the word cloud, we ended up deciding that was too difficult. So we came up with the idea of everyone's a dot, and your score would be represented by how big your dot is.

But if you're playing with 100 people, this is just unreasonable. And so then we finally came up with this last idea, which is nearly complete, where the height of your dot represents where you are in relation to the max score. And the lines are representing snap connections that are going along with the other players. So basically, I think for future projects for the front end, we definitely need to think about design sooner. This process happened really slowly and really close to the end.

We were very focused on the functionality of our game and the mechanics, and not a lot about the player's interaction with the game. It would have been awesome if we did some prototyping early on, because we had some other ideas. And we just never put them into fruition, and we never tested them out. So it would have been interesting to make those and see what users thought of them.

GUEST SPEAKER: So I think working with a large group was definitely difficult. And splitting it up was a good idea. But I think that having more responsibility and roles within those people would have been very helpful. We also should have thought about design sooner. I think also, from Miriam's presentation, the idea that we should have thought about the game and making it fun was something we should have focused on, as opposed to just thinking about porting a game that we had seen.

We really should have focused on what makes this a fun game, and what could we do to improve it? Because the core game can be done very quickly, and within the first couple weeks, we had the core concept done. So I think more quickly, we could have figured out, what feedback should we give to players? How can we make it more competitive, which I think it makes it more interesting as a game, as opposed to as a tool to collect data?

That's all we have. Are there any questions?

[APPLAUSE]

PROFESSOR: Anybody else from the class? Any suggestions? No? Well, we've got some. You want to go first?

PROFESSOR: So one thing that I noticed was there was a screen where you had the [INAUDIBLE] didn't future. That made no sense to me at all. I stared at that slide for a while. And then I heard what you were talking about, and then ah--what worked, what didn't, what are we going to do in the future. I can see how you might do it, because it fits with the snap theme of having just a single word.

But wow, that was really confusing.

PROFESSOR: Oh, that's why all [INAUDIBLE] are in small letters.

PROFESSOR: Yeah. It took me a while to realize what was going on. I'm guessing that. I don't know for sure.

PROFESSOR: Is that what you were going for?

GUEST SPEAKER: Not really. It was just for consistency.

PROFESSOR: OK.

PROFESSOR: Slightly more clarity would be helpful. You talked about the task diffusion, having a hard time getting tasks assigned. But when you talked about dividing your team into groups, you didn't talk about what the structure was running those groups or anything. You didn't have any of that was what was going on. But you might want to be maybe more explicit about that.

PROFESSOR: Yeah, like how did somebody take a task from Asana, by the way.

PROFESSOR: Yeah.

PROFESSOR: Was there a process for that? Did that actually happen that way? Or was it an assignment?

PROFESSOR: Yeah. Was there one person generating tasks? Or did the groups generate tasks collectively? That sort of thing. I think that's most of what I have.

PROFESSOR: So first of all, the demo for the game probably is what we're going to have to do on the real day itself, because it's such a large scale game. So it's not going to be like two people playing the game. So we can probably run the demo pretty much the way that you did.

GUEST SPEAKER: [INAUDIBLE] that's what we were [INAUDIBLE].

PROFESSOR: Is that the way that they do it when they run it?

GUEST SPEAKER: It depends. So Pablo has been actually doing a limited number of words. And only at the World Bank are they doing it like the way we are doing it, where everybody's submitting. And that, they haven't done yet. That's actually this afternoon.

PROFESSOR: You might want to talk a little bit about the history of this game. Everyone in this room right now has played the original game.

PROFESSOR: Absolutely.

PROFESSOR: Yeah. Similar to some of the things. I have it-- why are all your subheadings in small letters? Because it really, really threw me. Now, maybe it was unintended to make it look like your game [? snap, ?] but it actually does make it hard to read, your slides. All caps would be even easier than all small letters. There's one slide where you talk about adding roles for specific game design, UI, and everything.

But the heading just says, future. The big takeaway is more specific roles. You'll see it. So that should be on your slide. Your slides move really fast on average. And some of your slides are more word-heavy than others.

GUEST SPEAKER: OK.

PROFESSOR: So for the ones that are word heavy, them up a little longer, or make them less wordy. One or the other. Otherwise, I'm trying to read everything, and I can't even listen to what you're saying before the next slide changes. When you described the mobile app, I'm assuming you meant an Asana mobile app?

GUEST SPEAKER: The slack mobile app, yeah.

PROFESSOR: The slack mobile app, OK. Just be specific, because it wasn't entirely clear what you meant by that. So yes, put screenshots into your presentation instead of switching windows. That will just make things easier. There are maybe people in the audience who actually haven't used Phaser or know what Phaser is. So when you say, Phaser clearly isn't doing anything useful, well, they don't know what Phaser does.

So just explain. Phaser's a sprite-based game engine. We have no sprites, something like that. Two points about the game itself. First of all, how do you tell which dart is yours?

GUEST SPEAKER: It's the green one.

PROFESSOR: OK. I think because of the peripheral vision, I really did not even notice there was a green dot.

GUEST SPEAKER: Yeah.

PROFESSOR: So maybe it's just my eyes or something.

PROFESSOR: I didn't see it either.

PROFESSOR: OK. How many people actually noticed a green dot?

PROFESSOR: [INAUDIBLE] mine was so small.

AUDIENCE: There was a random green dot in the screen, and I couldn't do anything to it, so I kept [INAUDIBLE].

PROFESSOR: There is a green dot on screen, and it's really hard to see a green dot. There's something about the way how the human vision works on that.

GUEST SPEAKER: I think it's also because there are so many people.

PROFESSOR: Yeah.

GUEST SPEAKER: We've never tested with more than five people, and there are a lot of dots on screen when I don't know how many people were in that game--

PROFESSOR: I think it's cool that there is a dot that is yours, and that it has a meaning to it. But just being aware that that dot even exists and is different from the dots is more challenging than you might have realized.

GUEST SPEAKER: Yeah.

PROFESSOR: And then finally, one thing about the networking thing. it's actually really, really neat that [? Heroku ?] and [INAUDIBLE] seems to have solved problems that we traditionally, in this class, have just derailed entire teams. So it's a possibility that you just have an awesome team.

GUEST SPEAKER: Yeah. I think one thing I forgot to mention is that we have a lot of experience in all of these technologies. And so that helped.

PROFESSOR: What happens if somebody who doesn't know anything about node.js or hiroku decides, they're going to make a networking game?

GUEST SPEAKER: I have no idea.

PROFESSOR: So you might want to talk about that.

PROFESSOR: Mine, to reiterate-- client support versus class deliverables was a good point. Speak more about the client, and what you did for them, and what their role was on the team. It sounds like they had a more active role. And dependencies-- when you talk about dependencies and scrum, having difficulty with that, absolutely true. What did you do to get past it? Explain that.

Or if you didn't do anything to get past it, say that. But say whatever it is, rather than-- you do a lot of [INAUDIBLE] does this. But you don't go into the how. So go a little bit into the how for at least that point. Thank you.

[APPLAUSE]

PROFESSOR: Would you need a break? Or are we good on time?

PROFESSOR: No, let's just hammer through them.

PROFESSOR: All right. "Saving Gora Gora?"

PROFESSOR: "Saving Gora Gora," come on down.

GUEST SPEAKER: So we're the "Saving Gora Gora" team, us plus some devs in the audience. And I'm Liz, and I'll be starting you guys off. So quick overview of our game's goals coming into the project. We wanted to inform players about a cholera, and specifically, the causes, and the systems, and the prevention of the disease. And our audience, we wanted to narrow that down really early.

Specifically, we were more interested in targeting the younger age range of 8 to 13. And we had specific issues of that, which we'll talk about later on. And mainly, we wanted to have all these education goals and audience goals; be a part of a fun game, basically, like we wanted our game to be fun, and not one of those educational games they play, and then you fall asleep, and it's basically a poster.

A quick overview of what the game ended up looking like. This is "Saving Gora Gora." It's a children's game, and you are visiting a village of animals which is falling ill. And as a player, your goal is to help figure out what is causing the illness and trying to defend the character in the game, which we'll talk about now. A big part of our game is narrative, which is why we're going to do the demo later, because we wanted to ease you into the narrative without having you read all the text.

Everyone in Gora Gora is sick. We have this monster that is being blamed for it. He is totally a cool guy though, and he wants you to help you prove his innocence. You're Kojo. You're a little bunny kid, and you need to work with the town to prove to the town that it's not Sal that's making everybody sick. It's actually cholera. Basic gameplay is you're the main character.

This is Kojo. And you'll unlock minigames by exploring the map. Each minigame will give you a clue to prove that Sal is innocent, and your ultimate goal is to convince the mayor that Sal is a cool guy, and he shouldn't be in that cage that you see in the background.

GUEST SPEAKER: So our first minigame was about water filtration. And the basis of the minigame is that friends and neighbors in the village are bringing water to Kojo's mom, because they don't know whether it's safe to drink or not. So the goal of the game is to have the player decide whether each water source is safe to drink from. And along the process of developing that minigame, we had to actually iterate over the physical design of the minigame quite a bit, because our first idea was since it's water filtration, we wanted to make it perhaps a facility.

But we realized that wasn't relatable. It wasn't culturally relevant to all the kids who would be playing this game, so then our final design ended up being outside a home, as you can see in the background. So a couple other challenges we had. First, we had it working in real time, where the player had to rush to get things done in the minigame. But we thought that that took away from the overall purpose of teaching the kids, so we made it a lot more relaxed, a lot less time constrained, and allowed player to play at their own pace.

And lastly, we wanted to make sure that the game was clear. The game plan was clear. But we didn't want to make it too text heavy. And we considered making a tutorial, but we didn't have enough time to actually implement that, so we added some instruction text boxes on an overlay to just try to get the player a brief overview. And then during the game, they're also able to go back into these instructions and reread it so it becomes more clear.

GUEST SPEAKER: So the next minigame is a water collection game, where Kojo, the main player, needs to help his friend find a safe water source, since you have to go look for water and figure out where to drink, or where to get it. He basically helps his friend choose where to get the water from. So they basically show you a village map, and you would choose which places on the village map you would want to get the water from.

In terms of challenges and how I started developing the game, we wanted to make sure it was "fail-safe" so that even if the player chose the wrong choice, then they would get feedback. They would learn why that wasn't the right choice, and then be able to continue playing the game until they made the right choice. Originally, we did have an excreter. You had to find where you would also excrete.

But we cut that feature pretty early on, since we thought it was less crucial, and we wanted to focus on water. And then other challenges were, as Kevin said, we had some cultural issues, in terms of design, so we revamped our images. And also, we wanted to provide more feedback to the player as they were playing. So throughout our development process, we added things like helper texts and more explanations about why you did or didn't do things correctly.

GUEST SPEAKER: Our third minigame is focused around understanding the symptoms of cholera and understanding how important it is to act early on these symptoms. So in the game, the doctor of the village, Dr. Quasi, is very busy, and he can't go out into the village to help patients, because he has so many patients that are coming to him already. So he asks Kojo to go help him find sick patients to go talk to his friends and see if they have any of these symptoms, tell them to come to the clinic so that they can get treated.

During the development of this game, we initially started off with the idea that you'd be helping the doctor diagnose patients at the hospital. But then we realized that this would quickly turn into a quiz and wouldn't be as engaging or as fun. And it would essentially be the same as reading off of a pamphlet. So we wanted to add more of a game aspect to it, so we decided to make it such that you are going out and trying to find people, and suggesting to them that they go to the doctor's office. And then another one of the cultural things that we ran into is initially, you're playing as a kid. But we initially had it such that you're going around and talking to some of the adults.

And as Pablo pointed out to us, there's a much stricter social hierarchy in Ghana, and it might seem appropriate that we are suggesting to kids that they demand that their parents go to the doctor's office, because that's not something that would necessarily happen in the Ghana social structure. So we decided to change that around to you're talking to your friends, who are the same age as you. And you're talking to your friends, and you're having them go to the doctor's office. Some of the challenges we ran into, since this is a very conversation based game, making sure that the dialogue UI was understandable, and that you knew when you had choices available to you, to choose, and when you were talking, versus the other person was talking.

This was something that we had to integrate over multiple times. And finally, making sure that the dialogue was age appropriate. Since these kids don't speak English as their primary language and they're children, we wanted to make sure that the dialogue was simple enough for the children to understand-- but also, still had the meaning behind it so that they realized what actions they were taking through their dialogue. And now, we're going to start our demonstration.

GUEST SPEAKER: So it starts out with the monster hiding. And the monster is hiding from Kojo. And Kojo's like, why are you hiding? And so you see, people are blaming him for everyone's sickness. And so you say you'll help and you know who to talk to. But then when you bring him to the mayor, he's locked up by the mayor. And you're trying to convince the mayor that he's not the actual monster.

So you can see the beginning dialogue really sets up the scene for the story. And now, as Kojo, I'll say, I don't know, but I'll figure it out. And so now that you're in the town, you can explore the town. And you might be able to talk to some of the people. And so the people around town will give you things that will help you. So this dog is giving you matches.

And you might figure out later, oh. And now, you see of matches in your inventory, and you might be able to use that later. So for example, we can also click on the different buildings. And now, we realize that your friend is missing a bucket. So you tell them you'll go look for it. And we see that oh, the monkey has a bucket.

So then you can get the bucket from the monkey. And then now, you can go back and tell your friend, oh. You found the bucket. And so this brings you to one of the minigames, the collection minigame. And you can see, you can drag the bowl to different locations. And you can also click on the locations to see what they are. And you might say, I'm done. But it looks like this water spot is rusty.

So that's where the fail safe part comes in. And if you eventually drag it to a good source of water, it'll say, thanks. And you've acquired a clue. So then the next minigame, as Kevin was talking about, is you have to purify water. And so as you see, there's a lot of help or text to help you get started. And you can just drag the water. It says it's discolored, so I think we'll have to boil it. And eventually, the game, you have to figure out what to do with all the water and collect enough water.

And so I think we'll just cut that short, since we're a little short on time, and then we'll show the rest of the game for the final presentation.

GUEST SPEAKER: So in terms of like too technical challenges we ran into first is like the states and stays there, obviously, there are six of these are and we have to find a way to work around keeping a lot of paired data, in terms of clues and which games you had completed, what dialogues you were on, which dialogue should show up. So that was something-- that was a little bit of a challenge that we figured out. And then additionally, balancing difficulty with accessibility.

So we mitigated that by using things like helper texts, and using a hand cursor when you're allowed to click on things-- stuff like that.

GUEST SPEAKER: So one of the other biggest problems that we had, or I would say, the biggest problem we had is that our target audience and [INAUDIBLE] are completely two separate groups. We're targetting 8 to 13-year-olds in Ghana. We're college students in the United States. There's this complete different culture. And so we had to do a lot of research to make sure that our game was connecting to this audience. I think we did a good job. I think we could do a better job. We don't know.

We didn't get to play-test with our target audience. So unfortunately, we aren't 100% sure that this game's going to connect to these people. And another challenge is playing meaningful-- again, just because [INAUDIBLE]. So things that went wrong. Our communication was really good, but we never talked about deadlines for things. We would say, oh, we will have this done by the end of the week.

But then somebody would need [INAUDIBLE] assets, and it wouldn't be done. Sorry. Or we would want to play test the game at a certain date, and there wasn't a running game at the time. So I think one of the things that we had a problem with, just making deadlines, making sure people were doing things on a timely manner. And then we were having good communication, just not constant communication.

We were using several tools, but unfortunately, I think that's the next slide. We were using Trello, meeting weekly. We did use Slack. Unfortunately, not everyone was on Slack. Not everyone was using Trello, and not everyone was able to go to the meetings, and so sometimes, we would communicate with three fourths of the team, but there would be people that were completely out of the loop.

So things that went right. We did separate our team into basically two different sub-teams. There were four people coding, and then me and Liz were doing the art. She was doing character art. I was doing background art and all of the other assets. Then we'd have someone doing the [INAUDIBLE].

And so just by separating the team, it made things a little easier. You didn't have eight people touching code and breaking things or five different people making the art for something and having completely different styles.

We had a meeting every week, pretty much. Not everyone was always there, but it was very useful to just get up to date on what everyone was doing.

And then something that we did early was just cut a few features. We cut a fourth mini game that was picture matchmaking, which didn't really make sense, so we just completely cut it. Just cutting things like choosing where to dispose of your waste, something that we did early on that helped us.

So then moving forward, things that we would want to do is just test the game more. We would have loved to find someone familiar with our target audience or to play test with our target audience just to make the game better. Then changing things like difficulty and making sure the game is culturally relevant. I know I would want to play test more and then change some of the elements that I don't think are too great right now.

And then even going further, just expanding the game and making it more interesting, having more mini games and having more, an interesting mystery, rather than little paragraphs. That's all.

[APPLAUSE]

RICK EBERHARDT: Thanks. Any questions? Comments? I didn't think so.

OK. The demo that you did mid-presentation, you might want to rethink how you do that. Again, you're going to have somebody playing the game anyway. Maybe you could put some of that description while they're playing the game. Maybe you just demonstrate the map and say, here's a mini game, and then you'll see these mini games in play.

Or if what you're trying to do is help us understand what the games are, just give us a screenshot. That's probably enough. I don't think we needed it when we were watching the presentation to understand what you were talking about. I think you did a pretty good job of that when you were talking about individual things.

And then my big question, what I'd like to know more, is you mentioned the big thing as being the research you did for cultural appropriateness, what was it? Tell us what you did in the presentation. Maybe even give it its own slide. I think it's pretty important.

ANDREW GRANT: The social hierarchy example was a very good, specific example. We need more of those. So what changes you actually had to make once you realized that was a problem. You've got it in this umbrella category of cultural issues, but the specific stuff is really cool.

RICK EBERHARDT: That's it for me.

ANDREW GRANT: Sara?

SARA VERNILLI: No. I think Rick actually got most of what I was--

ANDREW GRANT: You might want to move the Tools slide before What Went Wrong because all the way things worked out, you just covered that first. The water filtration picture, where you say you can see it in the background, no. We can't see it the background. The background for that particular slide was totally invisible on this projector, so increase the contrast on that image or actually put it in as an inset.

You two need to warm up before you present. Your voices get louder as you speak, so practice outside of something. Warm it up and you're more audible.

This is more of a game design thing. There is a bucket right next to the well while you are trying to find a bucket, and every player is going to say, why isn't that bucket just good enough? Stick a hole in the bucket or something, whatever is the easiest way to solve that problem. That's pretty much all I had.

RICK EBERHARDT: OK. Thank you.

[APPLAUSE]

All right. Last up, Cholera Control.

STUDENT 2: Hi, everybody. We're Cholera Control. We're going to make this really simple. Our game was about cholera and it is a strategy game. You have to sort of manage many villages and try to tell them what they should implement in order to help prevent cholera within their village. And so we're going to just go through what went right, what went wrong, what we learned, and any future changes we would make.

So firstly, what went right. What happened over the course of the class is we simplified the game, and that was one of the best decisions we made that was really hard to do. Adjusting scope is really challenging because there are a lot of things you want to have in your game, you want to happen, but when you realize you just can't do it, it's like a load off your shoulders and you can really focus on the things that are important.

One of the things we did simplify is what the player can do. They have less actions, and we'll show that when we have some screenshots, but they're more meaningful. What the player is doing has a bigger effect on the game, and so they don't feel like they're just clicking buttons. They're actually thinking. They're strategizing rather than just, oh, I need to click these as fast as I can. And it's also more fun for the player. There's not too many. There's just enough.

The iteration was really one of the other things that went really right. It was the best experience in the class as a whole for changing the game is when you do that iteration, it feels really good when you look back and you see, wow, look how far our game has come. Look at the changes we've made. And comparing those really helps you realize the choices we made are good. Some of them are bad, but most of them are good. And we also had design meetings that were occasionally a struggle for everyone to attend, but when they did, they were really productive and really helped the group know what was going on, know what had to be done.

And lastly, the huge design change we did halfway through the eight weeks, this is what it originally looked like when we had a full digital prototype. It was really confusing. There are four villages. They had little bars over them. People didn't know what that meant. There's text in the top corner of the screen that you can't read and you couldn't even read when it was screen that told you your money and the population of each village, and then those two buttons that people weren't really sure were buttons, so that was really difficult.

This is what it looks like now. It's much clearer. There's a nice bar at the top telling you your money, how many days, and a nice little pause menu, and then there are only four villages. They take up the whole screen. It's very obvious that you want to click those. And also, there's a lot of indicators as to what's going on in each one.

Before, you had to click on each village and then up in the corner, that text would change to tell you what's going on in that village, and now, you can see it. It was a quick overview when you're watching. And so you have the population that's sick and that's not sick, and the green bar is healthy and red is infected, and then an arrow telling you the direction in which the infection rate is going, whether it's going towards more people getting healthy or more people getting sick.

And then you also have those little circles beneath each locality. We call them localities. It's a better term than "village." We wanted it to be very neutral, and that's what our clients told us we should use. And so below each locality are those four little bubbles that show what you've implemented there, and they count down. You might not be able to see it. It's a little small on this screen, but they count down and become less colored in or a little bit darker so you know how much longer it'll last.

And then for the menu of when you clicked a village, that's what it originally looked like. It was really hard to read. There were a lot of choices that you could do. We had eight, I think, is there, which at first, we were like, yeah, this is a strategy game. It's going to be really hard. There's going to be a lot of choices, a lot of things you can do, and this ended up being a very poor choice.

And so this is what it looks like now, where it's much clearer, first of all, how to close it, how to get back to the main screen, also what you can do. When you hover over something, in the Description box, its description pops up. If something's implemented, which nothing is in this locality, but if something was implemented, that circle would also be colored in and counting down to how long until you can implement it again.

So then what went wrong. First thing was coding. Working together is really hard. We learned this over the project. So we thought, OK, we'll divide everything up. We'll make it so everyone can work independently and make sure there aren't any real dependencies that are going to be roadblocks. But even doing that, it's still hard when you have to bring everything back together and there were just inconsistencies. People had different coding styles and merging just took a lot longer than it should have taken.

Motivation was also a really big problem. At the beginning of this project, people weren't really passionate about cholera or making a game about cholera, and even as we came up with designs and interesting choice and things we thought would be fun, that passion just didn't grow. It was always just like, I don't want to do this, which is really hard. I don't know how we could change that.

Also, the topic was really restrictive. It's something that probably that I'm guessing must happen a lot in the real world that sometimes you work on things that you don't want to work on, but you have to find a way to struggle through that.

We also had crunch problems. Our team was just a lot of crunches. There wasn't constant work. There was a lot of crunches. And so we had a lot of crunching right before a play testing class. We had a lot of crunching after Thanksgiving before now, and the work on the game was really low priority because everyone has other classes that they need to do things for. And so it is hard to tell people, no, you need to do this because they're like, I have a thesis project due. That's way more important than a project for this class.

And another thing that happened over time is that people weren't adhering to the communication we agreed on. We said, OK, we'll do emails and we'll use Gitter, which is sort of tied in with GitHub, and then people weren't on Gitter or they didn't reply to emails. And so whenever you sent out a message, it was like throwing it out into the ether and then hoping people would respond, which is really challenging. Sometimes you forget to log onto Gitter or you forget to look at an email until later, so it's difficult. It's something that happens, but it's a hard thing to deal with.

So then what we learned. Within game design, there's actually a lot we learned from this project. The biggest thing is that more complicated and more choices does not equal more fun. It can, and if you're making a really big strategy game like Civilization, I could see where you'd want to have the ability to do a lot of things, but in something simpler, you don't need to be able to do as many things. But if those things have meaning, which our choices now do, it's a lot more fun.

Play testing revealed that a lot of people didn't like having so many choices. They didn't know what to do. It was real time, so people were dying and they were like, I lost the game before I even felt like I started.

Design is very restrictive. You need to pay attention not only to what the client wants, but also who you're making this for. How accessible does it need to be? Who's the target audience?

And also, it's really hard as a designer to realize you need to put both the player and the client's needs before your own. You as a designer know, I have this grand vision of what our game should be, but then play testers say, we don't like your vision. You need to change it. It needs to do this. And your clients say, we don't like your vision either. It needs to change. You need to do this. And so doing that is very hard as a designer, but we learned to do it and the game was better as a result of it.

Some other things is none of us knew anything about cholera when we started, so we learned a lot about cholera. And it's surprising how easy it is if you just wash your hands. That's it. So if you're in Ghana, just wash your hands, guys. And probably don't drink infected water sources, but the big one is wash your hands.

And that also applies to a lot of other things. Pablo actually asked us to emphasize the washing your hands because Ebola also, it benefits if people wash your hands. It helps you so you don't catch other diseases.

And then about coding is tweening and other animation APIs make a game look more polished for little effort, which is something we didn't really implement until the end, and then when we did, we were like, wow, that looks a lot better.

And then future changes. So for just within the game, things that we never got to implement but we really would like to are subtle effects that would force people to change their strategy because sometimes it can get really easy, and so we'd want to make it so people who quickly figured out a strategy and then think, oh, we can win is we wanted to make some things that their strategy doesn't work anymore. So we originally we had the idea of a change in season, which is something we would like to do, but time constraints, or random events where there's maybe not a change in season, but there happened to be really heavy rain one day so everything flooded, so now all the water sources are more infected.

Also, multiple levels of difficulty. We did like simplifying it, but there are still some things we'd like to add, at least one more action and maybe a few more villages. So that's something that we would like to add if we had more time. And also, more advanced infection models. You can talk about the infection models if you want.

STUDENT 3: I'll chime in there. It turns out to be really, really hard to balance the spread of infection of a disease. It's almost impossible to make the game relatively simple at the beginning but then the right level of difficulty at the end because the nature of infection is that it's exponential. You change a parameter by nothing and it just swings the whole thing over and makes the game go from very easy to completely impossible.

So that made it really, really hard to balance. If I had more time, if I had a few more weeks to work on it, probably the things I'd want to work on would be work on something more advanced, like some more advanced mathematical models, as opposed to just a simple exponential model that we currently have, because that could make game balancing much easier. I had a really hard time balancing infection.

STUDENT 2: And lastly, some things we'd change for the class is we wished that we had something that enforced the sprint tasklist. We submitted it every week but we were never held to actually doing what we said we were going to do. So it felt just like, oh, it's something we have to do. We have to say we'll do things this week in order to be working on the game. So we wish there was something like the instructors asking, OK, can we see your completed sprint tasklist?

The restriction to the Red Cross topics was a huge constraint for the creative freedom, especially for the last project. So because all of us didn't really like the topic to begin with, it was a really long eight weeks. So we wish that we didn't have to do that, but I understand for this semester of this class, that was what happened.

And then another thing is that we wouldn't work on an educational game regarding an unknown culture for the final project because it was just really hard to understand, OK, what do people in Ghana find normal? What would they want in a game? Do they understand how this works? It's something that's even hard because there are a lot of assumptions people who have played games make about games that's really hard to get over. And so doing that for the final project that also has to be an educational game that also has to be about cholera, it's just like there's a lot of things piling up that made it really hard to do.

So any questions? And we'll also show a demo of the game, but if you guys want to ask any questions first.

[APPLAUSE]

Should we do demo first and then questions?

RICK EBERHARDT: Yeah, go ahead and demo.

ANDREW GRANT: Both on this screen and with some of the earlier teams, many of you are running off Mac and you are using a screen resolution that's better for your screen than for the projector. So it's actually rescaling it to fit the projector, which means actually, every pixel that you're drawing on screen is actually too small for the projector and things get blurred out. In the Screen Resolution Settings in System Preferences, you can actually select whether you are scaling it optimized for either display. I guess in this case, you are optimizing it for display.

RICK EBERHARDT: Which resolution are you using?

ANDREW GRANT: It looks like you're using 1600.

STUDENT 3: What's the best one?

RICK EBERHARDT: It's on the board right behind you. There's a wide screen and a four three version.

ANDREW GRANT: So check that because you're using a lot of visual detail when you're scaling it like that. It makes your games look worse. That looks way better.

RICK EBERHARDT: And the color scheme is going to be better, too.

ANDREW GRANT: Yeah. Your colors do look better as a result.

STUDENT 3: Cool. Awesome. So we still have a little bit of things to iron out, but the idea of the game is that you manage these villages. Cholera is spreading and you have to use various prevention measures to stop it from spreading further.

So when the game starts out, there's just one village. People were having difficulty figuring out that you could actually click it, so we made it do like that so you know you can click it. So you click it and then there's various options that you can use to stop the spread of cholera. So it's just the first village, so I'll just go with soap, which is the cheapest option, and that usually stops. I'll make it go faster.

The first village is sort of a tutorial. Once you've figured out how everything works and you're able to stop the infection there and you cure it, then a second village pops up. As the game goes on, more villages pop up, up to a total of four villages. Basically, at the beginning, it's quite easy. You can't really lose on the first village unless you sit and stare at it and do nothing for three minutes.

But the second village comes up and now you have to think about things, then you get an idea of how the game is actually played. By the time the third village comes up, you really need a strategy or you're going to lose. And the fourth village, it just gets really, really hard. Generally, most people actually fail in the third village, don't make it to the fourth one. I guess if you guys want to see more game play.

RICK EBERHARDT: Is there sound?

STUDENT 3: So we don't have sound right now. We're going to add that in today. So the three different options, basically, soap is the cheap, all around option because we want the takeaway of the game is soap is overpowering. These two options distinguish themselves in that water containers are good for stopping an infection from spreading but they don't cure anymore, and electrolytes don't stop anything from spreading but they immediately cure just a percentage of the infected population.

What we found is that I thought electrolytes weren't actually going to be fun, but then they are, because at the end of the game when the infection is all over the place and everyone's dying, you just keep spanning electrolytes. It has a cool down so you can't keep spanning, but you span it as much as you can to desperately stay alive. I don't know how far to actually take this.

RICK EBERHARDT: That's fine for us. A live player is going to play it on their own, so they'll be figuring stuff out. Again, give them a couple minutes, and then you can start talking if you'd like to talk over.

For the presentation itself, when you started out, we could use a little bit more context about what you're talking about. You kind of just throw us into what went right and one of the things that went right, so maybe telling us what you're going to talk about might be a way to give context for the full presentation.

STUDENT 2: Would starting with the game give a better idea?

RICK EBERHARDT: That would be good, especially if you're talking over the game and talking about some of the design. Then we can kind of call back.

SARA VERNILLI: Another context related issue is you refer to your client and you refer to the game and you refer to making the game about cholera but you don't actually explicitly state who your client is, what your game is for, what the goal you were trying to create your game for, what specifically it was. You could gather it from context, but it's a nice thing to have upfront.

ANDREW GRANT: So the early slides, they're pretty wordy. Both of you are actually really good speakers, so I feel like you can get a lot of those points in your actual verbal-- you did, in fact, get all those points across in your verbal things. The slides don't need to also restate every single thing. So stick to high level headers, but the way how you're actually presenting these points are coming across very clearly verbally and the words are starting to distract a little bit.

RICK EBERHARDT: Yeah. Each one of those could have been just one slide plus an image.

ANDREW GRANT: That becomes a backdrop to the things that you're actually saying, which are actually coming across quite clearly and you don't really need all of that additional visual noise, especially when you have sub-bullets, like you have a heading and then you have tiny little sub-bullets underneath. Those sub-bullets, just say them. Don't represent them.

The image of selecting which updates you have for each locality, right now you have the blank rectangle with mouseover text in it. Nothing is lit up. If you just replace that with a screenshot of something's lit up, something's not lit up with the mouseover text, that will be way more informative, that one slide alone.

When you talk about balancing the infection rate, that's a really good detail case example. I just want to call that out. That level of detail is neat because you're talking about really something very specific. Other things that you can talk about-- I'm trying to remember the way how you had set up a communication system of using email and Gitter, and this is what Gitter is, but people weren't responding.

That was good. That was very specific. That was like, this is the way how something specifically broke down. You can talk about how things eventually got resolved would be nice, but maybe it didn't get resolved, and you can say that, too. That level of specificity was nice.

SARA VERNILLI: One more. Specifically on the slides, you talked about how you did iteration. You talked about you made changes and you stayed very generic there. I think some actual specific examples we really would like to hear. What are some things that you dropped? What are some things that you had in the beginning but you changed a whole lot? They're in your end game but we don't recognize it from the beginning game because it got changed [? in right now. ?] You don't need a whole lot of examples, just one specific ones would tie it back.

ANDREW GRANT: Yeah. So if, in that particular section, you just added, here's a specific feature that got changed in this particular way, that's great because you clearly have that information.

RICK EBERHARDT: All right. Thank you.

[APPLAUSE]

So thank you all. Everyone, sorry we ran a little bit late, but timing wise, everything looks like it went out. Tech wise, it looks like we've got most of our stuff figured out.

A couple points of order. This did go long. It's probably going to go about full two hours on Wednesday. Do you want to take a break midway through? Would you like an intermission midway through? I'm seeing some nods, so we're going to put an intermission midway through.

Lastly, based on the number of speakers that we've got and the topic matter that we're talking about, I think what we're going to do is start with Snap because everybody will have their computers out from the get go and then we won't need that again afterwards. Just take care of that right there. You're going to have people playing live in the audience.

Then we're going to go to Hello Wave. Basically, we're going to swap Heat Wave and Snap and Saving Gora Gora and Cholera, swap those two. And for that, because we're going to have lots of speakers at the so that will just make it so that you're doing all the moving around. Cholera Control, you're going to have two people for that, so we're going to make sure we have two microphones for all of that.

We'll have the intermission right after Heat Wave. So that's about an hour and a half, and then we'll have another hour, and then that's the end of the class on Wednesday.

ANDREW GRANT: So that's the order of presentation right there.

RICK EBERHARDT: Yeah. One, two, three, four, five. There we go. Cool. I'm trying to think if there's anything else that I should get to before we release you. No. Student evaluations, web.mit.edu/studentevaluations. I'll write up the real URL right there. Please do that in class today if you could. Thank you so much. You actually are doing a really good job. I really like these presentations. I'm really excited to see what they're going to be like on Wednesday, and I was really happy to see working games today, so aces on you for that.