

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

PROFESSOR: So to all of our guests, thank you for coming. This is the final presentations for CMS.611 crossed listed as 6.073 the Creating Video Games class. In this class, we have 40 students, learning how to make games and learning about the production process of making games.

So a lot of the things we're talking about in class, a little bit about design, but often also about how do you communicate as a team. How do you get work done on time and on schedule, as I overheard a lot of things happening this last week.

Every year, we choose a client to make games for. This year, we were very blessed to have the Red Cross/Red Crescent Climate Centre, in Pablo Suarez. As our common client, he presented a number of different games, ideas, and topics ideas to the class. The class formed teams about those for the final projects. And that's what we're going to be seeing today.

Our order of presentations are going to be over here. So we're going to have three presentations. Each presentation is about 20 minutes plus another 10 for Q&A and then just tech switches.

We'll have the intermission after *Heatwaves* is finished. And then we've got two more. *Cholera*, I did not write down the full name for your game. Your team, Lauren. What is the full name for your game?

LAUREN: *Cholera Control.*

PROFESSOR: *Cholera Control.* Make a note of that. So they and [INAUDIBLE] will be last. I also wanted to mention to all the students here, if you're interested in continuing your education on game development, as you can see, spring is loaded with game courses. CMS.301, which is taught by Sara and Michael in the back-- hi, Michael.

MICHAEL: Hi.

PROFESSOR: It is an introduction to game design methods. In particular, it's unpacking the idea of prototyping [INAUDIBLE] for digital games. And the lessons learned there can be applied to

pretty much any kind of designer, but they're focusing on game design methods.

CMS.608, Phillip and I teach, is board game and card game design. It's structured, actually, very similar to this class with a few small projects and then a final project at the end that you spend about half the semester on. And those, of course, are non-digital games, so board games, card games. We've had people do live action RPGs and tabletop RPGs, role-playing games for that class.

The next two courses, CMS.610, Art, Science, and Business Video Games, and CMS.617, Advanced Game Studio, also built on the team building aspect but not as harshly as we do. You won't be in eight-person teams, unless you really want to. Teams tend to be about four to five people for those classes, and actually as low as three in some cases for 617.

Right now in the catalog, they both look like they're being taught on Thursday. We are trying to move 617 to Tuesday. So I'll send out an announcement about that.

And then lastly, if you liked creating the specific games you made in class this semester, these kind of educational games, or do you want to know more about how games are used in education, how to make games for education and how to simulations in particular for education, CMS.590 cross-listed I think-- I didn't look up the number-- but my memory said 11.129 is Computer Games and Simulation for Investigation and Education, tion.

That is taught-- last year, it was taught by Scot Osterweil and Jason Haas with Eric Klopfer is the faculty on that course. A really great course, I highly recommend it. You can look up the syllabus in OCW if you want to know more about it. And with that, we are going to start with *Snap*.

TEJ CHAJED:

All right, hi, everybody. I'm Tej, and I'm from the *Snap* team. We're going to start off with a demo of our game. So the way our game works is it's called *Snap* and everybody participates from their own computer.

We'll give you a topic, and then you'll enter words related to that topic. And whenever you get the same word as somebody else, you'll get a point. So it's pretty simple. So we'll invite you to join us at snapgame.org.

We'll be showing the moderator interface. So normally, you wouldn't be seeing this. The moderator would do it on their own. But we'd like to show this, because it is a part of our

game.

All right, and the topic we'll be doing today, to mix it up from last time, is video games. And we'll run the game for about two minutes this time.

All right, so I have no idea how long it's been. But I'm going to give a countdown, and I'm going to stop the game. 10, 9, 8, 7, 6, 5, 4, 3, 2, 1. And that's a game.

Congratulations, Julia, and Kevin in second place. This is what you guys invented. Thank you, guys, for playing *Snap*. That was [INAUDIBLE].

AUDIENCE: Luigi [? book. ?]

AUDIENCE: What was that?

AUDIENCE: Luigi.

TEJ CHAJED: All right, so as you can see, this gives you an idea of what you were thinking about, what you think about when you think video games, which not surprisingly right now is *Snap*. And with that, we've given a feel for the moderator interface. And hopefully, you got a chance to play the game and see what the player sees. And we'll start the presentation.

So this game was made for the Red Cross. Pablo actually helped co-develop this game. And he presented it to us as a face-to-face game that they play at conferences. He provided feedback and guidance.

And in our project, in particular, the client was extremely helpful in giving us an idea of what they wanted and also helping us run the game in their actual settings. So we were able to see what this looks like with the players that we expect to play with.

For our organization of our project, we split up into backend and frontend teams. The backend worked on both the server and also the moderator interface you saw here and to word cloud, for example.

This split was challenging for new features, especially, because features would be interdependent or frontend couldn't do anything until there was a backend. And there was a communication gap throughout. When we needed to add a new feature, the API between the backend and frontend wasn't necessarily well-documented.

But overall, we think that it led to simpler management within the teams. We didn't have a lot of organization within each team. There wasn't a lot of explicit organization. But it did help to know that a smaller set of people would be working on the frontend or the backend at any given time.

In the future, I think we would have gone even more specific in our roles. Like a few roles that we thought would have been important include a UI designer. This would have made sure that UI design saw some first class love from the beginning and would have made sure that we actually focused on UI from the beginning.

It also would have helped to have somebody communicating between the teams. This person may have done some documentation or may have simply just made sure that the frontend knew what the backend was working on and the backend knew what the frontend needed, so we could prioritize well between the teams.

It also would have helped to have somebody dedicated to paperwork. This would have helped both in submitting class paperwork and also just doing the documentation that we needed, both for ourselves and the class.

And finally, somebody working on game design would have helped us develop more iterations early on and think more about the game aspects that go beyond what we already knew the game would look like based on the game of *Snap* we played from the clients.

As far as technology for our organization, we used Asana to organize our tasks. Asana worked really well for us. In particular, it let us split the task lists for the frontend and backend. This was useful, because generally, the frontend didn't need to touch the backend tasks and vice versa.

But on the other hand, we had a hard time encoding task dependencies. This was particularly a problem for a feature that crossed both the backend and frontend, where some API was needed on the backend and then was also needed on the frontend. And you couldn't do one without the other.

We also had trouble with task creation. And this basically meant that people didn't create tasks. And part of this was that we didn't have a specific role for creating the tasks. And so, because of the diffusion of responsibility, nobody would go through and list out everything that needed to be done.

To improve that in the future, we'd like to get people more invested. Part of this was that people had a lot of other classes. And it would have helped if people both explicitly said how much they were going to commit to the class and also we made it easier for people to contribute with the small amount of time that they had available.

**SABRINA
DRAMMIS:**

So for communication, our group used an instant messaging application called Slack. And this was really great for us, because it allowed us to separate into different channels. As you see here, we have frontend and backend as two different channels and then a general channel where everyone can communicate.

And this let the frontend people work on their thing separately from the backend, not needing to know exactly what was going on. But they could go look, if they needed to reference some conversation there. The bad part about this was that people weren't always online and responsive right away.

And I think we could of made this not a problem, if we required everyone in our project to download the mobile application. That way, they would get a notification immediately when they were mentioned. And they would be more likely to respond quicker.

TEJ CHAJED:

On the backend, generally things went really well. Our technology, in particular, worked really well for us. We used node.js for the server, and socket.io for two-way communication with the browser, and Heroku to do deployment.

We were explicitly forbidden from using networking in this class, which we violated from the beginning. But I think that our experience with networking and all three of these technologies really helped make that possible. We already knew how to use them. And that helped us both debug issues and understand what we were working with.

On the other hand, the backend it seemed had too much manpower. This manifested itself as a problem on the frontend, where sometimes we felt like we didn't have enough people. We could have had fewer people working on the backend, which would have made sure that it was a small unified code base, and we could quickly finalize the backend and just move on.

We actually deployed the application, and the client has been using what we showed. This is actually a quite early deployment on November 18 in Berlin with some students there. And you can see the words that they generated.

Here are some other deployments. One of the things that made this class-- what was challenging for us is that we had to manage both client expectations and the course expectations.

So because they were asking us to do these runs, it seemed important to us to support these runs. And it was hard to balance doing this and also during the tasks that the course required of us and needed for us to do.

**SABRINA
DRAMMIS:**

So I'm going to talk a little bit about our frontend design and that process. So initially, when we started the project, we had this idea that people would be represented as a car and they would be racing other players. So we thought it would be a great idea to use Phaser, which is a sprite-based game framework.

But as you see here, we didn't go through with the car idea. And we had no sprites. But we still used Phaser to make this, our first design iteration. And we realized we didn't really want that type of look for our game. We needed something more professional and more abstract that could be used in any sort of setting that Pablo, our client, may use it.

So we completely scrapped Phaser. And we redid our frontend design, deciding to just use Bootstrap and CoffeeScript. And this was our second design.

As you can see here, it's just very typical Bootstrap styling, if you've ever used Bootstrap before. So we tried to add a little bit of flavor. And we also needed a better way to kind of represent how the game is progressing to the player. We currently had no idea for that in this design.

So then we came up with the final design that you guys played, our theme of having blue. And then we created this visualization to the right, where each player is represented by a dot. And snaps are represented by lines being drawn to dots. And your height is representative of your score in relation to all the other players.

So how we kind of got to this final visualization idea was an interesting process. First, we thought it'd be really cool to have a word cloud on the frontend, as well as the back end. But a player would only be able to see the words that they had submitted. All the other words would be blurred out.

But this didn't really convey to the player where they were in relation to all the other players. So there wasn't any sort of competitiveness. So we came up with this idea of having a race

line, where all the players would be on the line and the height would represent their score and their position.

And then we thought, well, we can put these both together. But we investigated the word cloud. And we realized it'd be extremely difficult to blur out these words. And it wasn't really going to be the best use of our time.

So then we came up with this idea of using dots to represent players. And we thought it'd be cool if the size of your dot represented your score. But the use case for our application could involve 100 players or even more.

So a lot of growing large dots could really clutter our UI, so we needed to rethink this. And this is when we finally ended up deciding on height representing score.

So I think we had really good UI designers. And they were really quick. Once we thought of ideas, we were quick being able to implement it. But the bad part was that we were really late in coming up with design decisions.

We didn't start thinking about how we wanted to convey the game to the player and how we wanted to make the player feel about the game until really late in the project. We focused more on what the client needed as far as mechanics and not really what the player was getting out of the game.

So in the future, definitely we need to think about design right from the get-go. And we need to start prototyping all the ideas that we have and testing them out more thoroughly.

TEJ CHAJED:

So in summary, the high points of our project are that we did make the client happy. And they've been using our application and the game. And they really like the game.

We're glad that we focused on fun. The game has been fun throughout. And players seem to really enjoy it. And I think maybe this is just because of the core concept. And we're glad that, even though this is an educational game, it is intended for purposes honestly of data collection, people really enjoy playing it.

Unfortunately, we had slow design iteration. This prevented us from really fully realizing all the ideas that we had and figuring out which one would be best for players. In the future, we'd like to get people more invested to try to solve these issues and make sure that we best use our time and create the best product possible. And with that, we'll take questions.

AUDIENCE: [INAUDIBLE] Is there a maximum number of players?

TEJ CHAJED: We did some stress tests to see how much the backend on Heroku could support. And that seemed to work OK up to 150 players or teams. We are not really sure what the frontend might look like at 150 real players. That might be actually the limiting factor.

AUDIENCE: Is that 150 people playing the same session or in the multiple sessions 150 people?

TEJ CHAJED: We currently don't support multiple sessions. This is actually a request that the client repeatedly gave us. But we thought that it was just much simpler to avoid that for now, because it wasn't strictly necessary.

Most likely, the load on the backend would be the same, whether they were split into multiple sessions. But this is a solvable problem by getting a beefier backend.

AUDIENCE: I see. But right now, the backend supports 150 people logged in simultaneously.

TEJ CHAJED: About 150. We can obviously pay money to get better servers and then run with many more.

AUDIENCE: OK.

TEJ CHAJED: Any questions from the audience?

AUDIENCE: So you got some feedback from Pablo specifically from the moderator point of view. And I'm wondering whether you got any indication about what the players felt outside of this class plus your [INAUDIBLE]

TEJ CHAJED: Sure, I think we can both talk about our focus tests, where we played with players outside this class.

**SABRINA
DRAMMIS:** Yeah, so one of the main things, I played with maybe a group of at most 10 people. And they knew each other, so they wanted to know who they were snapping with. And they wanted to see the names of all the dots, so they could know who they were beating or who they needed to catch up to.

But this would be unreasonable in a large environment, especially when all the people at the conference probably don't know each other. But I think that would be a really cool thing to do, if we continued working on this in the future.

TEJ CHAJED: And I also ran a similar test with people that knew each other. And they actually used it as part of their strategy. They knew what other players would be submitting for that topic. And they could use that to their advantage.

So, for example, we did a topic of systems papers. And we had a professor in the group, so his paper showed up a bunch. And I would echo what Sabrina said, that players really wanted more feedback around the actual players in the game, which we specifically didn't do, because the client didn't care. And in conferences, people don't generally know each other.

But I think that if we were to use this as a more popular game, then definitely focusing on players and getting more feedback, especially after the game is over. People wanted to know who they were snapping with. They wanted to see that graph, all those lines overlaid, so they could see who they were snapping with most often.

AUDIENCE: Does the team have any plans for this game after this class? I was wondering if you've ever had discussions yet.

TEJ CHAJED: I think we haven't discussed that yet. But I think definitely we're interested in it.

AUDIENCE: Thank you.

TEJ CHAJED: Thank you.

MATT: Hello. We are *Hello waves*. We're game about forecasting, specifically this idea called forecast-based financing of using forecast to make decisions about possible disasters in the future and how to prepare for them. We would actually like to start with a playthrough of our game. So if anybody would like to be a volunteer to try it out.

AUDIENCE: A guest from not our class, if one could please come down, Andrew.

AUDIENCE: Thank you for volunteering.

ANDREW: Hi, I'm Andrew.

AUDIENCE: Well, do you need a chair?

AUDIENCE: Here's one.

MATT: I'm not quite sure how to make that full screen. But anyways, this is our game. I'd recommend looking at the instructions first and reading through them. Thanks. And we'll keep the

description minimal as you read through it just to sort of show the player's initial reaction to it.

ANDREW: Now, this is real, correct?

NORMAN: You're signed on?

MATT: Oh, is it? Oh.

NORMAN: Yeah, right. OK, sorry. Music.

MATT: Yeah, start up again. So as you can see, our game, as I said before, you control some toys on a day at the beach. And so by dragging and dropping them between the castles, you'll see that their statuses change and say that they want to move on their next turn, or that they want to be collecting-- or that they're going to be collecting candy, or anything like that.

And the game is turned based. So all the actions will resolve on the next turn.

ANDREW: Am I rating this race?

MATT: Yeah.

NORMAN: You can also access the help through the lower right corner as well.

MATT: And so when you go to the next turn, you'll see all the toys move as specified by the status bubble above their heads.

ANDREW: So I go to the next turn, or do I get a next turn?

MATT: But unfortunately, you'll find that when toys have been evacuated, they're unhappy and need candy to survive, so they'll all take damage.

ANDREW: They don't want you to evacuate.

MATT: You can try returning them to their homes.

NORMAN: And so you'll see that on this turn instead, now they have their statuses set to gathering candy, except for the dump truck, who is still evacuated.

ANDREW: Got you, OK. But he needs to be evacuated.

NORMAN: Exactly. And so the idea is that as a player plays through it, they get better at understanding

how to use the forecast to make decisions about the future, both in terms of how much candy that they need to have stocked up in order to weather out the rising tides and also in terms of when they're going to need to move their workers out of-- their toys out of the areas that are in danger.

ANDREW: They're going to be really impacted.

MATT: I think you have those two guys.

ANDREW: Oh, they're in there? Is there a reminder of where they started?

MATT: Yeah, it's on the castle. It's actually blocked out right now, but toward the front of it. But there's a little shadow there, an imprint. Because he ran out of candy and then-- or actually, sorry, because he went to a place that was underwater, he took too much damage and then was swept away by the waves.

And so on the forecast, you can see that high water is coming for quite a while, which is going to be a danger for the toys, both in terms of possibly getting swept away and not having enough candy for all the toys that you're going to have to move out of the way.

ANDREW: [INAUDIBLE] this time.

NORMAN: Oh, he may be out of luck.

MATT: Yeah. And when you've lost two toys, you lose the game.

ANDREW: OK.

MATT: Thank you for playing. So that's *Hello waves*. So in our game, we had a few challenges to overcome. The first was that our game was based on forecast-based financing, which is a very abstract topic.

It's a pretty understandable idea of using information about the future and ideas of risk in order to decide where to allocate resources. But it's still a bit abstract. And building a game around it took a little bit of work.

It's useful to note that it's different than long-term planning. It's not just thinking about what will happen in the future, building a dam to prevent water or things like that. And it's actually about using the information you have to make the best decision for events that may be upcoming in

the semi-near future.

We also wanted to avoid making a game that was overly preachy or simplified, where it was clear exactly how you're going to win. And you could just basically push the forecast-based financing button to win the game. We wanted players to actually think and understand the concepts there, instead of just coming up with the buzzword of forecast-based financing.

And finally, we had the challenge of actually communicating that forecast to players in a way that they would be able to understand and then make use of. You could see in that game that we had water levels and it would show on the map. And we found that players were pretty good at using that in order to make decisions about what was going to happen in the future and how to allocate their resources.

The other big challenge that we had in the beginning was that our initial target audience was policymakers. Like for *Snap*, Pablo had come in and pitched us this game idea. And originally, he had wanted us to build a game for policymakers that would help them understand the benefits of forecast-based financing and, therefore, convince them that they should develop policies that would give resources to plans based on forecast-based financing.

So I would like to take you through a couple of our prototypes just to show you the evolution and comment sort of on our process. Actually, to preface that, we had a lot of prototypes, because of our idea was so abstract and because we weren't sure how to address our audience. So we built a lot of prototypes to start with.

We had ideas that ranged across sort of levels of scope of what you controlled, where you controlled entire cities or where you controlled individual workers and moved them around.

And then we would pull from all these different kinds of ideas what worked, what didn't, what did people understand, what confused them. And from that, we got a really good idea of what concepts helped people understand the idea and brought them into this final game that we ended up with.

So the project started on October 15. And this was our first prototype. It was a terminal-based game, where you had some kind of information about a future rainfall. And then you had to type in your commands of how you controlled different cities.

This, actually, people found it fun. But as you might expect, the feedback wasn't very good. And people didn't quite understand how to move forward with it. It put a lot of cognitive load on

people.

So when we moved forward, we tried to give people more easily understandable actions to use. But the problem with this game was that it was time based and it updated every second. And so there are so numbers flying at people, like even MIT students who play tested it couldn't understand.

So we figured that people like policymakers, who didn't have much experience with games, really wouldn't be able to understand the game at all. So instead, we went to turn based. And that helped. But at the same time, it's tough to see on this projector, but we have a forecast underneath that says how much rainfall is expected.

And again, that wasn't understandable to people, because they couldn't understand what three inches of rainfall meant for their city. And they couldn't understand how that contributed to a possible disaster.

At this point, Pablo actually came by the class and played the game. And then he told us that he wasn't even sure that he could get policymakers to play the game, because they may not have enough time. Instead, he wanted us to switch to grade schoolers, because we could teach them something about forecast-based financing and help them understand as they grew up.

And this was great for us, because making a game that was serious, easy for somebody who didn't have experience with games to play, and also fun and engaging was too difficult for us actually. So moving to grade schoolers was awesome. He also suggested that we try to make the idea of rainfall or water levels more visceral. And that's when we came upon this idea of the rising waters.

Whenever players looked at this, they instantly understood the concept of the game. The feedback might not be there. The beautiful UI might not be there. But the idea of having cities with workers in them and a rising water level coming towards them, everybody understood that. And it made it a lot easier for players to reason about the game.

From there, we added things like nicer art, better feedback, which you can't quite see in the static picture, more improvements to how the forecast worked. And eventually, we ended up with our final version today.

NORMAN:

So to talk a little bit about our actual development process, so our team was structured into three main subteams, production, which was in charge of managerial roles, deliverables, and play testing, and so there was a shared responsibility there; technical team, which was in charge of the bulk of the coding work; and a user experience team, which would be in charge of assets, UI design, and UI design.

We also initially envisioned subteam leaders, where we'd be kind of communicating through them. But we found the concept kind of redundant. And so we've worked pretty much with like a flat structure between the three teams.

So from the beginning, we encouraged good coding practice. And so we used good tools available to us. One of these is Yeoman, which is a JavaScript scaffolding framework. And this helped us out a lot by basically automating a lot of our JavaScript tasks and making our code modular.

We also used Phaser state machine, which is this kind of badly documented new feature in the Phaser game engine, which was the JavaScript game engine that we used. It's a bit badly documented. But it did save us a lot of headaches. And once we figured that out, that proved immensely helpful.

And we also used MVC, which is a computer software engineering term standing for Model View Controller. And again, by encouraging these good coding practices, we reduced dependencies and made sure that our team was productive.

In terms of communication, we also, similar to *Snap*, used Slack, which is kind of like a modernized chatrooms. It's very feature rich. And so you can share files, channels, and things like that.

And we also use the idea of the Daily Scrum. We implemented it in class. And we would say what we had done that class, what we would be doing, and what we wanted to do until next class.

And so now, the major challenges that we faced during the development process though where that our team members came from very different backgrounds and had very different preferences about games. Some of our team members are very hardcore *StarCraft* players and very good at RTS games, while other members of our team preferred like a more laid back mobile game *Fruit Ninja* kind of approach.

And so trying to mediate those two viewpoints and trying to create a game that would engage both types of gamers was a challenge that we had to overcome. And so another big challenge that we had in our development process, as you saw through our progression through our different prototypes, was that our direction was not very clear, until about halfway through the project.

And so partially because we had different ideas on what the game should look like and also partially because we had such an abstract idea of forecast-based financing that even we didn't have that good of a grasp on initially, it took us a while to really get settled on what we wanted to build.

And so this really challenged our development process and made us have to build a lot before we got something that we liked. And so eventually, we did end up having to cut some features of like multiple levels or like a guided tutorial for the player.

We thought that this would introduce too much new content that would need to be play tested, balanced, and tested to ensure consistency with the rest of our game, which we viewed as taking up too much time. And we also cut the idea of adding more individuality to the workers or to the toys that you saw, other than different graphics for each.

MATT:

Some of the worst things that we did on our team is that we spent a lot of time on work that got thrown out entirely. All the prototypes we did, they were actually pretty useful, because of the things we learned about the concept and about how people would play the game.

But we did spend a lot of time on things like art or nitty gritty details that didn't really need to be figured out and that we could have put off until later in the project. We also kept the game direction too vague for too long. As Norman said, we spent a lot of time with that. And it probably ate up too much of our time.

Although it helped us learn, we could have moved faster in the beginning to get to a solid idea. Because once we got to a solid idea of the rising waters, our team started to centralize around it a lot better, because we could actually deal with something concrete. While we were dealing with the abstract ideas, everybody was all over the place and arguing about things that didn't quite line up.

And the worst decision of all that we started with and that sort of made the problem of going too vague and all these other things happen is that we were originally thinking, how do we skin

forecast based financing as a game? How do we take this idea of using forecast to make decisions and then just gamify it, which we eventually realized wasn't fun and didn't help us actually come up with any ideas.

Instead when we flipped it and started talking about what game could we create and use forecast-based financing to improve it and teach players how to play the game and, therefore, allow them to come out of the game, having learned about forecast-based financing, the world sort of opened up. Everything became a lot more interesting. And we found that we started to move faster.

Some of the best decisions that we made, on the other hand, as Norman said, we had good tools, which meant that even when we threw out prototypes, although we wasted things like art resources and things like that, we actually didn't end up wasting very much code, because things like the idea of workers or cities could literally be pulled out of the old games we had, put into our new game, and then reworked to build our new structure.

We also weren't afraid to trust each other and throw out the things that didn't work. Once we started moving fast, we had a lot of ideas that would come out, and we would say, OK, this doesn't work. We're actually going to scrap it. Or we think that this isn't the direction we need to go in. And everybody was willing to go along with it.

It's not a good feeling to see your things thrown out. But everybody understood that was for the best of the game. And I really appreciate their understanding with everything too.

And part of that all comes down to the fact that we were on board with our final idea. We were all excited about the concept that we had. Part of that might have been the relief of coming to a concrete idea after spending so much time being vague. But once we had the concrete idea, we really moved fast and worked well around it. So thank you and any questions.

AUDIENCE: Who did your sound? It's awesome.

MATT: That was from our UI team.

AUDIENCE: Oh, [INAUDIBLE]. OK, I really liked it.

MATT: Thank you.

AUDIENCE: Where did you find it?

NORMAN: So it's on our credits page. Most of it was online. The credits page in our game.

AUDIENCE: And so you don't want to type it in.

MATT: It's a little small, I guess, up here.

AUDIENCE: Oh, OK, great.

MATT: But it looks--

AUDIENCE: [? Switch games. ?]

NORMAN: Yeah, "Hold My Hand," AGP by--

AUDIENCE: Would it be possible to do a [? shortened ?] URL of these?

MATT: For the game? Oh, yeah, sure. We can make a Bitly link, and we'll send it out to everyone.

AUDIENCE: Yeah, just to entire-- yeah.

MATT: That's a good call. Yes.

AUDIENCE: Having watched the early crash and burn playthrough, how often do people-- I don't know if you've actually had a lot of people to play your current version of your game. Do people usually take a playthrough or two before they start getting the concept?

MATT: Yes. That's why something like a tutorial would be really nice. Unfortunately, we didn't have the time to put it in.

AUDIENCE: Yeah, no. I was wondering how that-- OK.

MATT: Yeah, usually what happens is, even if after maybe a couple of turns of playing through they start to get the idea, the problem is that, as the water starts to rise, they haven't prepared enough. And so all their workers or the toys will starve or get carried away by the waves, which is a bit unfortunate and probably makes the players feel bad on the first time.

But then they-- it actually teaches them to think ahead about it. So the next playthrough, they're much more careful and understanding.

AUDIENCE: Yes, I was wondering if you were playing any other games or thinking about other games as the inspiration or thinking about how to deal with sort of getting the right level of strategic

thinking in your game.

NORMAN: Well, so I guess in terms of early on. Because we had a very different idea of what we wanted to do early on in the process, we were thinking about games like *Civilization* and how did they communicate all these complex worker movements and managing multiple cities. But once we actually came up with this game concept, I think we had much, much smaller goals. And so did we have any specific game models, do you think?

MATT: There is none that I specifically think of. There were some things that we were sort of inspired by, standard tricks, like when you hover over one of the characters, they got bigger, some idea of showing off that this is clickable, things like that. But specific games themselves, not really.

AUDIENCE: I was just thinking it ended up being kind of war gamey. And I feel like there's a lot of war games that where I'd thinking a lot about getting that right level of I guess tactics rather than strategy. But, yeah, I guess it worked out kind of.

AUDIENCE: So you mentioned that you were able to completely change how your workers looked and just keep your models. So that's sort of the Holy Grail of object-oriented programming, that kind of object that's reusable and you don't have to throw out the code. Do you think there's a reason why, in particular, you were able to achieve that, because I think that's not-- that's not common necessarily in object-oriented programming?

NORMAN: Partially, a little bit of OCD-ness like very early on, very strictly saying we're going to write this object-oriented code, and we're not just going to hack things together. I think that helped a lot, because we actually spent the time in the very beginning to think about like which objects were responsible for what and what their purpose should be.

Yeah, basically, I think because we moved more slowly in the start and thought more carefully about how that code should be structured, we ended up with having an easier time later on.

MATT: There's also the fact that because we had learned these things from the prototype that we are putting into this game, that also meant that the objects that we created because we wanted similar functionality to things that we had already seen and we knew worked, it meant that we were comfortable pulling out the functionality into that.

I don't want to say it was designed to fit. But there is the fact that we moved it on purpose, really.

AUDIENCE: I would like to say something on it. So we had a very good MVC model. So models were at and has a tree-like structure, I think it was very easy to change the model somewhere else [? and knew it ?] involved-- sorry, used our new evolved models once [? models and no idea ?] of what happens. So basically, it was [INAUDIBLE] change.

AUDIENCE: Thank you. All right, thank you.

MIRIAM: Hello, my name is Miriam. And today, I'm going to be presenting for Team Heat Wave. And we made a game called *Heat Wave*. So the goal of *Heat Wave* was and is to educate Red Cross workers about how to prepare for and handle heat waves.

And if you've never been through a heat wave, it's just defined as elevated temperatures for an extended period of time. And the way you prepare for that is you set up umbrellas for shade. You get water coolers. And during the actual event, you try to go inside and drink an excess amount of water.

So while we had that game goal, we also had design goals of usability, playability, and education-- usability because if the game doesn't open, it doesn't turn on, that's a problem, it doesn't work, playability because we wanted people to be able to play through the whole game, it's just very fundamental, and education because this game is meant to educate Red Cross workers about how to deal with heat waves. So we really wanted that to be one of our primary objectives for the game.

So to meet these goals, we had our design processes. And I'll talk a little bit about each of these in detail in a minute. But we started off just as the other teams did with brainstorming. Then we formed a team based off of our ideas.

And then we broke down responsibility. So one person was in charge of one aspect of the game. Another was in charge of like the graphics, all separate tasks.

And then we continuously updated. And what we used for that was Trello, primarily. And I'll talk more about that in a minute.

So brainstorming, we were actually right there on the board drawing things for a while. And we had two major ideas going as a group of people just considering this topic. The two ideas we had were the game we actually came up with, where you play as a Red Cross worker trying to help people. And then our second game was playing as a heat wave, trying to make people faint.

And while that actually sounds like a lot of fun, we stuck with the first idea. And the reason we did that is we thought we could get more education into the game. Because education was such a primary focus for us, we really wanted it to be the core basis of the game. So we went with the game that would directly correlate to what these Red Cross workers would be doing out in the streets before and during a heat wave.

And based off of that idea, we formed a team. We actually had a pretty large team. This is eight out of nine of our people. Joe wasn't in this photo, because he wasn't there that day.

But we had a really large team. So we had to be really focused as to like who was doing what and what was getting done when. Because when you have such a large team, things can slip through the cracks.

So we used Scrum. And as you can see up there, it says so many emails, because there were so many emails. But the way we dealt with that based off of previous projects was to say, well, we're going to have a lot of emails. There's nothing we can do about that. But we can make the emails relevant, so we can break them into different threads.

So the people who were handling the character objects, they're just going to have their own thread. And maybe like one person who's communicating between teams will also be on that thread. But it's really their thread. So we had a lot of emails. But they were divided up as to who got to see them and who it affected.

And the other thing we did is we had our Trello. And we constantly updated that. And we kept track of what was being done in this sprint and previous sprints. And it was really easy to see what was taking longer than we expected and what was done, what was done but could use improvements. Like you could make a note. We had like a not-done-yet section. So we really managed to stay organized that way.

And the other thing we did to make sure things got done was group meetings. And these were quintessential for our groups. We had a bunch of like three-hour long meetings. And we would get task after task done, bug after bug fixed, and really just worked through for hours on end.

So once we had this game and we were all working on it, we had four focus tests that we ran. And they all had a theme. So our first focus test was about the concept, can we educate people in the basic mechanic of that we think we're going to use. And I'll talk about that in a

minute.

The second one was overarching issues. So I said our three design goals were usability, playability, and education. So that was really our second focus assessed.

The third focus test was visuals. Are we connecting with the user? Are these the visuals we want to have?

And then for the fourth one was game balancing. Does this game feel fun to play? Is it working out?

So the first focus test, we focused on the general mechanics. We had a low fidelity prototype, which is actually a Python-based game. And there was a list of people. And you could say, I want to talk to this person based on some attributes and offer them water, tell them to go inside.

And each turn, you could either continue talking to them, say forget it, you're being annoying, or oh, you accepted water, I'm going to move on to the next person. Or you could do nothing. And what we learned very quickly was that it was working.

People were choosing to try and help people based on their characteristics and based on oh, well, this person fainted really quickly, and they were homeless, so I think homeless people might faint faster. Though actually, I don't think homeless people were in this game.

And we also experimented with different types of dialogues. So we experimented with having like very factual dialogue, drinking water is really important, even if you're not too thirsty versus something like, hide your wife, hide your kids heat wave.

So once we had that general concept in place, we built a basic unit game. And we wanted to see overarching issues. So we had these four-- well five, but one person has already fainted in this picture-- five different characters with different attributes. And you could choose to talk to them.

It was the same exact concept we had in the Python text-based game. This time, it was a little more visual. There were people and there was a dialogue that would show up. But it wasn't fully there yet.

And what we saw was the usability and the playability were definitely having issues here. You

can see that. But the education was definitely coming through it. In fact, people wanted it to be more visual.

And we saw that, and we said, well, we have to test more visuals. So you can see, this was the version we tested later. And we said, how is this working, who are you choosing to click on? And people said, well, first of all, it's confusing.

All these people look the same. I can kind of tell who's a woman and who might be a man, but that's about it. The labels aren't moving with the people. Even if they were moving with the people, they're really too small. We can't tell who's who.

And we said, well, we have this art that we think we're going to redo all the art in the style of. We're thinking about getting rid of the labels. Do you think that's a good idea? And they said, yes, get rid of the labels. They're too small anyway.

And so we showed them this caricature of one of the types of the characters, which is a homeless person. They're a very vulnerable group to heat waves. We said, this is kind of the art style we were looking at. We think it displays more information. Do you think that's good?

And we got enormously positive feedback on that. They say, definitely, I would click on this guy, because he looks like he needs help. And you'll see that later.

Something else we were having an issue with visually was that the way people would go inside is they would just kind of wander out. But the way they fainted, they would actually just disappear, proof. So people didn't notice it actually. And then they'd suddenly be like, well, there was 10 people, and now there's five people, and what happened?

So we said, OK, we know this is an issue. What do you think would be the best way to visually show that people are fainting and that would emotionally resonate with you? And they said, well, we want them to fall over.

And maybe it'd be nice if there were some other feedback, like sound or something. And we said, OK, we'll take that a step farther. And we did. And we'll talk about that later.

And the third thing that we really got back from this visual feedback focus test was that there was no indication when people were getting sick. So in the real world, if you're getting sick, you might like turned red, get really sweaty. If you're about to faint, it'll be very obvious is the point.

And there was no indication here. So we said, well, we're going to do something about it. We're going to put that on the back burner, keep thinking about it. And we are going to fix that somehow. And so we put that all into practice.

And you'll see the screens in a bit. But our final focus test, we focused on game balancing. Is this game balanced? Are people fainting at a reasonable rate? The problem was that we would have like six people all fainting at the same time.

They said no, they have to faint at different increments. And also on top of that, we had an issue where the temperature was somewhat random, going up and down. And people were not adjusting to that.

One day it was 110. The next day was 80. And that's in Fahrenheit, because I assume most of you use Fahrenheit typically. But our game is actually in Celsius now.

But the idea was that people were not adjusting to this change. And it wasn't realistic. So what we do now is we actually ramp up the temperature, and so that the first day, very few if any people faint.

And then the next day, by the end of day, people start fainting, because it's getting hot. And then the third day, they're fainting faster. And you'll see that in the playthrough after this.

So during this whole process, we made some good and bad choices. And I'd like to talk about bad choices first. So we made two major-- these were the two, I think, biggest bad choices we made. They're not completely horrible or anything.

The first was that we depended on external code. And the second is we focused on education, not fun. So when I mentioned the external code, one of our teammates had a dialogue system for unity, which is really cool, it looked really nice, had lot of features. And we said, let's go for it. We can use this as a foundation to build a really complex game.

And something we really didn't stop to think about was how much work is it going to be to develop our game around this system? How much work is it going to be to change the system to fit our game? And also are we going to have time to do all of that and also focus on making the game complete as a whole, instead of just kind of showcasing this one aspect.

And it turns out it was a bit of a pitfall for us, because we didn't expect it to take as much time as it did. And if we were to go back, the teams kind of said, well, maybe we still do this and we

just deal with the issues early on. But also, maybe we should have just designed a simpler game and not use this external code.

Another bad choice we made was we really did focus on education, which was good, because our game does teach people. But it is an educational game in the sense that it's not the most fun in the world. And if we were to redo this, we probably should have focused on fun earlier.

We played with having funny dialogue and making more emotionally resonating effects. But at the end of the day, it's an educational game. And it's really not that exciting to play. So that's something.

But in the process of doing that, we made some good choices too. And one of those good choices we made was group meetings. You get so much done when you sit down with like most of your team for three hours and you just hammer through. And we did that multiple times. And it really worked out for us.

Something else we did well was prioritize. Because we had a team of nine people, we really had-- availability was kind of all over the place. And even when we had group meetings, you could only really expect like six people to show up. You were guaranteed that at least three people weren't going to be there. Even if you did a doodle, which we did.

So what we prioritized was based on group-- we had groups of people with different responsibilities. But if there was a bug, we said, OK, who is available right now? What are you're working on? Is this more of a priority? Can you do this now?

And that process worked for us. We got bug after bug fixed that way when we weren't in a group meeting. And when we got to the group meeting, people generally had a sense of what was going on, because we were doing this process of they kind of had maybe worked on something for a little bit based on it being a priority.

So now, let's get to the actual game. So the game has a start screen within an instruction screen, which is kind of the sublayer. The main game, which has the day scene and then a newspaper scene at the night, and I'll show you all these. And then the end screen.

So originally, this was what our start screen looked at that we play tested. And it was beautiful, as you can see up there. No need for changes, right?

So we actually did change it. And so now, it's the same background as a game for *Cohesion*.

You have your play, but you can also use the instructions, which are pretty useful if you've never played before.

And then that takes you to our main game. Our main game originally, you saw that on the first focus test. I think this is pretty similar to our second focus test.

You see we had a progression with the main game. It took a lot of iteration. And something originally we didn't have, I said, was there was no indication of when somebody was getting sick or their characteristics. It was a lot of issues.

So we made some progress. We added the indicator that say, I'm about to faint. You should probably try to help me now, which is that little explanation part on part of their head.

And then finally, we redid pretty much all of the graphics. And that took a while. So we were in the process of redoing them at this point. And now, the game looks like this.

So it's much more cohesive. All the art fits together. The people are really visually caricatures of the different types of people you need to help.

Elderly people, homeless people, drunk people are all really vulnerable to heat waves. And then there's some characters that actually aren't on this screen right here but that aren't as vulnerable, like normal, like adults aren't as vulnerable. And athletes aren't as vulnerable.

So at the end of each day, when you play this through, and you've helped your people, and you've given them water or set up coolers or whatever you wanted to do for the day, you get to a newspaper scene. And the original newspaper scene looked like this, a lot of text. It doesn't really help you with much.

No one really cared much about it in the first iterations of the game, because the long-term objects hadn't been adding yet. So it's just the umbrellas and the water coolers. And so we changed it to look like this.

It's the same information, just a little better organized. And the other thing that we added in the main game, we added those buttons that says, you can set up a water cooler or an umbrella. And that takes your time up.

So if it's cool outside, maybe you do that. But if it's like 110 degrees in Fahrenheit, then maybe you should focus on helping people not faint. So we had that.

And then say you finish, 20 people faint or whatever-- I think it's 20 people faint-- and you get to the end screen. And our original game, we had no end screen. And the reason for that was, I told you we really focused on education not fun. And we just didn't think, part of the fun of a game is seeing how long you survive. But we just weren't thinking about that.

So we really reevaluated that. And we said, OK, we definitely need an end screen. And we have it looking like this. So it tells you how long you lasted.

And then you can see the credits for what was involved in the game. And you can play it again. So there's more feedback for the user of how they did.

And if we were going to do this project again, there's two things I think we'd do majorly differently. And the first is talking about heat waves. Our group thought heat waves were going to be really cool. We were like, oh, people are fainting everywhere, and it's getting hot, and we have to do something.

And then you start reading about heat waves. And basically, it gets hot. You tell people to go inside or drink water. And that's about it. That's all a heat wave is.

And so what we'd do differently is really the passion aspect. If we're going to do a game about heat waves, we depended kind of on our passion coming from the thinking heat waves were cool. I think we should have focused more on having passion for the game and then incorporating the-- well, also having the game build around heat waves but really having a more exciting game that we could somehow also get that educational aspect in.

What we'd also do differently is accountability. I said we had nine people. Sometimes, people disappear when you have that many people. And keeping tabs on everything, especially when you have separate email threads, can be a little confusing. So maybe look into some of the tools that the other teams mentioned, where you have multiple group chats and you can see them all. That looks really interesting.

Future features, more long-term options. I think that's one of the things in our game that really helps with the planning aspect is OK, do I want to be doing this, like helping somebody now? Or do I want to be installing an object now?

And really, our game kind of depends on that balance of choosing who to help and then also what to do. Am I helping or not? So more long term options.

And then something we originally conceived was to have multiple environments. And we didn't have time to implement this. But in different environments, you do slightly different things. Like if you're in an office, you might want to install an AC unit, which is different than what you'd do outside.

So really look into multiple environments. Different countries have different environments, might have slightly different strategies for what to do. And we really would like to look into that.

So before we play the game, do you guys have any questions? In that case, I'd like a user tester.

AUDIENCE: So someone who hasn't played the game before, again one of our guests. Alex?

AUDIENCE: Sure.

AUDIENCE: Thank you.

MIRIAM: It's not showing up there. Give me a second. Display. You're going to have to give me a minute. I didn't realize it wouldn't show up.

AUDIENCE: Make sure that's not plugged in.

MIRIAM: Yes. I [INAUDIBLE] have it plugged in. How do I mirror the screens? Sorry, it just-- otherwise, he's going to have to stare up there. Oh, thank you. Don't look at my emails. There's so many of them. OK, yeah.

AUDIENCE: Are we good?

MIRIAM: Yeah, go for it. My sound is on silent. There.

AUDIENCE: All right? Let's see, solo, go [? to tech ?] the instructions so thanks for [INAUDIBLE]. All right.

Let me read through the instructions. It's going to offer homeless people water, tell old people to go inside, install umbrellas, or many other options. OK, so pause, click on Pause button. OK. [INAUDIBLE] whoop.

MIRIAM: Sorry, what just happened?

AUDIENCE: I may have clicked that actually.

MIRIAM: Oh, did you start the game? Here, we'll just quit. No worries. Great.

AUDIENCE: OK, thanks. OK, I think we're ready now. Let's play. I'm going for the umbrellas. OK. Yeah. All right.

Wow, that's nice. Does it get darker as it goes through? Cool. Whoop, uh-oh. Wake up. Hey, OK. Should I do it around here?

MIRIAM: Yeah, sorry. Go through again. Try talking to someone.

AUDIENCE: All right. Is it--

MIRIAM: No, it's something long.

AUDIENCE: Oh, OK. No worries.

MIRIAM: All right, that's good. Thank you.

AUDIENCE: Maybe, just start the game over.

MIRIAM: Yeah, should I just-- yeah. It was working. OK, let me refresh the page. There.

AUDIENCE: Oh, there you go.

MIRIAM: I don't know what happened. Sorry about that.

AUDIENCE: No worries. Oh, I see, you can give them options to do. Oh, OK, that changes it just a little, yeah.

MIRIAM: Kind of important, just a little.

AUDIENCE: All right. Oh, OK. Let's see, I'll talk to her. Uh-oh. Oh no.

AUDIENCE: I actually caught the rest of the game.

AUDIENCE: Thank you. Any last questions for the group?

AUDIENCE: I liked the bird sounds and also the fire sounds. Are there other sound effects?

MIRIAM: So we had music. But it was really hard to find good music for a heat wave, because you think it sounds hot and you don't really think of anything. So we had this very eerie music, which didn't really fit either. So we went for the minimalistic bird sounds, because we thought it was the only thing that kind of fit.

Is there any other sounds? No. Yeah, we had other sounds. Just they sounded wrong.

We had fire sounds, like in the background and that made the whole thing sound really dangerous and terrifying, which is not what we want to portray. We want to portray that you can do something and prevent anything bad from happening. And any other questions? Yes.

AUDIENCE: So a lot of things have changed in your game since Monday. A lot has changed since Monday. How did you manage the process?

MIRIAM: So we actually, we were going to have a meeting before Monday. But really, our schedules didn't work out. So we actually had, I told you, group meetings. That's where we got everything done.

We had like a three, four-hour group meeting on Monday night, which we were planning to have. And while there was like a couple core things that we wanted to fix, just like small like bugs that we were working on, and also while some people were doing that, we're just like, well, we have other people who aren't doing anything and can't really help with what's the major thing, so why don't we just like add other features, like it getting light and dark.

Because that's something we really had considered and we hadn't had time for. But in that group meeting, where somebody had time, we were able to add that. And it looks good.

AUDIENCE: How did you test it? Like, did you--

MIRIAM: Oh, so we've been running it like nonstop all of yesterday and today. But then it broke.

AUDIENCE: Yep, definitely, it happens.

MIRIAM: Yeah.

AUDIENCE: You also went from a screen that had pretty small characters on it to a screen where the characters were about 3/4 the size of the screen. How did that decision come about?

MIRIAM: So we had our drawings. And when I told you we focused tested, then we had like a big picture of them on the screen. People loved it. But when we actually put them onto the old screen, they were really small and you couldn't see their characteristics.

And we said it's better to have people overlapping more often but actually being able to see

their characteristics. Because the caricatures are really where the learning comes in. Like, if you didn't notice, the people who fainted were, I think, the drunk people and an old person.

So you start to notice as you play the game, OK, homeless people faint really fast if you don't give them water. And that's just like a fact. People who are homeless typically don't have enough fluids in them. And they're also outside a lot. So it's really hot.

So that's relearning comes in. So we wanted to make those bigger. And to make them bigger, we thought, OK, we'll just put in like a simpler one-lane background. So that's where that came from.

We were planning on doing that before yesterday, I mean before Monday. But we just didn't have the assets in yet. So like a lot of things changed that were already in the works since you last saw it. Cool. Any other questions? Well, thank you so much.

LAUREN: All right, so we're *Cholera Control*. And we're going to talk a little bit about our game and the making of it. So firstly, *Cholera Control* is a game we made for the Red Cross. Our client was Pablo Sanchez, as a few others have mentioned.

And the point that we wanted to get across with our cholera game is that it is an easily preventable disease, if you just like wash your hands. It's really not that hard. But people just don't seem to do it, because there's a lot of factors going on in their life and they sort of just forget. But it's really easy to do, if you just wash your hands.

So we want to just do a quick demo of the game. Where is-- OK. So if someone would like to come down and play our game.

AUDIENCE: OK.

LAUREN: Cool.

AUDIENCE: Cholera is fun.

AUDIENCE: Not supposed to show up on our home video, so that's OK.

AUDIENCE: Are you hot?

AUDIENCE: No, I'm just kidding. I don't know.

AUDIENCE: We'll find out. Don't have any policy for that.

AUDIENCE: OK. Play a game.

LAUREN: Can you guys read that?

AUDIENCE: Yep.

LAUREN: Can anyone in the back not read it? Because, it--

STUDENT: It says cholera is spreading. You got to control it and watch out for outbreaks.

AUDIENCE: OK, I'll just-- Oh, OK. I have \$1,400?

LAUREN: So as you can see, the game is kind of a strategy game. Right now, there's only one locality to deal with. We sort of didn't want to throw too much at the player at the very beginning.

So you can kind of-- also, the game is paused, so it's going to open in that corner there. And so you can have time to sort of just read, make sure you know what's going on, what all those different things do. Then you go back to the map. And time keeps elapsing.

And so over time, more localities appear, up to four. So right now, you just get the second one. They are on different branches of the river. So they're not going to really hurt each other right now, still relatively easy.

STUDENT: We go cheap.

LAUREN: Hopefully. I don't want to offend anyone, if they're like, this is really hard.

AUDIENCE: So the average [INAUDIBLE] that the infection is still growing, more so.

LAUREN: So one of the things you could see, if he goes back to the map, is all the bars above the villages, green is the healthy and red is infected. And there's an arrow, which I think is a-- you can see it pretty well on there-- the arrow, if it's pointing that way, that means that the amount of green is growing. So there's more people getting healed.

If it's pointing the other way, then more people are getting infected, which is bad. So if you see your arrow switch directions, you usually want to get on that really quickly and make sure that nobody dies-- well, no, nobody gets infected. Nobody dies in this game. We didn't want to be too harsh.

AUDIENCE: Are the products accumulated? So my like--

STUDENT: Yes, yeah, so if you buy more than one, they work together.

AUDIENCE: OK. Ew, boiling water.

STUDENT: So boiling water is special, because it helps downstream villages.

AUDIENCE: I clicked that.

STUDENT: That doesn't help.

AUDIENCE: I know. Is my budget growing?

STUDENT: Yes.

AUDIENCE: OK, oh great. Oh, crap, I didn't mean to do that.

STUDENT: You can undo.

AUDIENCE: Oh, I can?

STUDENT: Yeah, well, now you can't.

AUDIENCE: Now, I can't.

STUDENT: That one's going to-- you're [INAUDIBLE].

AUDIENCE: OK, I need to buy them boiling water, and that's it.

STUDENT: You're done.

AUDIENCE: OK.

LAUREN: Well, thank you.

STUDENT: Yep, thanks. Yeah, so if you manage to survive longer-- no offense-- you get to-- up to four villages pop up, or four localities pops up. And then it gets really hard. Cool.

Cool, so first, we're going to talk about what went right, both from the creation of the game and working as a team sort of perspective.

LAUREN: So firstly, I'll go over the iteration was a lot better on this project. In this class, we had a total of

four projects. The first three were each over-- like, the first one was two weeks. Second one was two weeks. Third one was two weeks. And then this one was eight weeks.

So this time, we had a lot more time to sort of decide what we wanted to do, what we should try out, and get to sort of test each and every single one of the ideas that we had, rather than just focusing on one idea of having to do it, because you only had two weeks. The other thing is that after three projects, we finally figured out how to do design meetings really well.

We got them down to that they were only like a half hour each. And when people showed up, they were really productive. And so everyone figured out what they wanted to do. They talked about the game. It got done, and people got out of there. It was very nice.

Another thing we did right was simplifying. It was really hard to adjust scope. It was easier in the other games, because you realize you only have two weeks.

When you have eight weeks, you think you can do a lot, and you really can't. You think you have a lot of time, but classes and applying to jobs, and everything takes all of your time. And so you have to really learn to, OK, like, I only have so much time for this class. I can only do so much on this game, so I'm going to adjust the scope.

And it was really hard even when the client, when Pablo told us, you should change what you're doing, we didn't want to. We thought, no, this is a great idea. This is how it should work. But then when the client tells you do it and all your play testers say your idea is bad, you need to just give in and change.

And we ended up having really great results. With the simpler gameplay, it's a lot more fun. And the options, there are less options, but they're much more meaningful. As you could see, the player was sort of reading through each and every option. And it's because they each do something different.

Before, we had a lot more options. But they were all sort of the same. You just kind of click them, and they do something. But they're all just about the same effect.

With this, they all have different effects. So you sort of have to strategize. Do I want to spend more money for this effect? Or can I just use this cheaper effect over and over again for the same result?

STUDENT:

So I'm going to talk a little bit about what changed from beginning to end. So the UI in the

beginning looks much uglier. I mean, that just came with sort of refining that. I want to talk a lot about sort of the design changes that really changed.

So in our first prototype of the game, you sort of click these buttons here on the right. And that would give you the actions. Whereas, in the final version of the game, you click on the villages, and that gives you the actions.

And this came from something that Pablo told us early on and really made a big difference, which is like people that are used to playing video games get this concept of click action, apply to object. But we realized, both through play testing and again Pablo telling us, that when you don't have that sort of experience, working in a lot of videogames and just having that context in your head, that doesn't really make sense.

It makes much more sense to the people that we're aiming this game for to click on the object and then choose an action. So like that inverting of the order of things actually makes a big difference in helping to understand the game and just making the whole thing smoother.

On top of that, we sort of like really refined the information that we're showing when you look at the map. So right now, like in the original version, you look at the map and you sort of don't see straight away everything you need to see.

You need to go to each of the villages individually to know what's going on. You sort of can only see where each village is, or where each locality is and how much health it has.

Whereas in the final version, we sort of give you every you need to see. And it took a long time to figure out how we wanted to do this UI. Like, how do we show what prevention measures are in place?

How do we show whether the infection is growing or decreasing? And how do we show which way the river is flowing? All of these questions were really important and really led into the final design.

Secondly, about like options, at first we had lots of options, too many. People did not even want to figure out what each of them did. It was too much information thrown at once.

And although we thought that having lots of options would lead to meaningful-- like, oh, well, you can make these advanced strategies. It really didn't, because people didn't even want to bother learning all that.

Whereas, in the final version, having fewer options made it much easier or like much less scary when you opened up the list of options, the actions that you have available to you as a player. And it was much more accessible, while still making the strategies diverse.

So now, we're going to talk a little bit about what went wrong in the process of making the game. So working together is really hard, because people are stepping over their toes. Oh, what do I do now. Oh, I can't touch that, because you're touching that. Like, all these sort of problem arise.

And so at first, we really tried to avoid that. But then we realized working apart is even harder than that. Initially, you think, well, if we're each working on really sort of independent parts and we bring it all together in the end, that'll be happy.

Everyone will be happy. We're not stepping on our toes. We can each work on our own time and that sort of thing.

But that turns out not to work well, because people do things differently. People follow-- like, even simple things, like coding style and how you name your variables and all that sort of stuff makes a difference.

And if you're not working together closely, and if you sort of working apart, bringing that together was a huge hassle at the end. Like we spent a lot of time just making sure everything, all the components that we had made as individuals worked together, just to get the game running.

LAUREN:

So another thing that went wrong was the motivation for this project. At the very beginning, the team wasn't very passionate about the project to begin with, which is really hard. It's really hard to work on something that you don't really love and want to do.

And I thought, you know, maybe this will just get better. People will start to have ideas. They'll feel ownership of the project. And then they'll be more motivated to work on it. But that's not true.

People, if they start off in a bad place, they're not going to get any better. You really have to focus on somehow helping people to either learn to love the project or learn to just be able to work on it. And this can be done through different ways.

The easiest way is just to get people to realize like we have to do this, so you have to work on it. And I found it as-- towards the end of the semester, it took me to figure this one out. But if you give people vague instructions, they will wait until the very last second. But if you make it crystal clear, exactly what you want them to do and give them is as detailed instructions you can, it's much more likely that it'll get done in a timely manner.

And also, we felt really restricted by the topic and the requirements, which was another just problem in motivation. And so crunch is something I was sort of alluding to, is that when you have the instructions and you don't want to do something, you will put it off. I do it.

I am sure everyone else does it that I don't want to do this. I don't understand what they want me to do, so I'm just going to wait until the last possible second when I have to do it or I'm going to fail. And that's when it gets done, which is what I think was a leading thing to our crunch. And so we had constant crunching. Every time we knew we had to play test, the night before, everyone was up working.

And there was a huge crunch after Thanksgiving. We took until Thanksgiving, and then everyone realized we don't have that much for longer in this semester. We only have like a week and three days before this final presentation. So there was a lot of crunch there.

And the work was low priority because of other classes, which is a really unique problem to doing a school project, as opposed to doing a project in the real world, where people still might have other things to do, but this is their job. Whereas, when this is a class, you can't really fire someone from your class.

Let's see, you want to talk about what we learned?

STUDENT:

Yeah. So what we learned in terms of game design, at least for me and probably for everyone on the team, what was most surprising is that giving more choices and making a game more complicated does not make it more fun. Initially I thought, more choices means more strategic diversity or whatever, and that'll make everything more exciting. And every time you play the game, you can try something different.

But that's not really true. Having less choices can be even more fun than having more choices, because then you don't have-- like when the player doesn't have to think about as much at once, they can really go deeper into what the current options are. So that was really surprising.

Also in terms of game design, requirements can be restrictive and often you can't do what you want to do. You as a designer can't do what you want to do, can't fulfill your own fantasy for what the game is opposed to look like, because of the requirements, because of what clients want, and because of what players want.

And you have to learn to sort of deal with that. The designer is not always right. Just because I have an idea for what the game is supposed to look like, if I test on 10 people and all 10 people say I don't like it, I'd prefer it to be this way, then I'm wrong. I have to change that. And that's something that's hard to deal with, but it's something you have to learn to do.

Yeah, we learned a lot about cholera. Basically, as we said in one of first slides, washing hands is overpowered in real life. If you just wash-- very simple things can prevent cholera. And that's something we wanted to get across in this game. In the game, washing hands is overpowered, just so that they get that idea across.

And about coding, there's a lot of simple things you can do to really polish up the game, like tweening, which is a animation API available, and Phaser that we used. But really any animation API that does little things, like having the window pop up instead of just appear, having things go like "paching!" and move in and out, that makes a huge difference for very little effort. And it makes the game look much more polished.

In terms of future changes for the game itself, we initially planned to have seasons and lots of random events that would sort of make the game more exciting, especially near the end, once all four villages are up and you sort of get into this pattern. But we basically didn't have time to do that.

And then the second thing that at least I personally wanted to add to the game was more advanced infection models. It turns out that modeling infection is-- well, it's not surprising really. But modeling infection is really hard.

Balancing your model of infection is really, really hard. And it's sort of almost impossible to make it have this sort of smooth curve, where it starts out easy and then gets sort of hard at the end but not too hard. Infection is exponential in nature. And that's really hard to deal with.

If I had more time to work on it, I'd probably spend a lot more time thinking about different models to use for infection and make it easier to balance. Like right now, you change like one little number by hardly anything, and the game goes from really, really easy to completely

impossible, just because of the nature of infection.

LAUREN: And lastly, future changes for the class. One of things that we all felt would be really useful is if the professors had enforced the sprint task list, because you could submit your sprint task list at the beginning of your week. But you never had to tell them if you actually did everything.

You just had another sprint task list, which maybe if that it was looked at carefully, there might be the exact same things two weeks in a row. But a lot of times, you never heard anything about your progress. You just kind of had your progress.

And it wasn't until the end that we really realized like, oh, we're kind of far behind. We need to fix that. So it would have been nice earlier in the project to have that sort of feedback, like, hey, you should be-- are these things done? You should be doing more.

And more freedom in choosing the topic, like we mentioned before, we felt really restricted. There were things we wanted to do that we couldn't, because of the confines of this project. And so a little more freedom would've been nice. But you know, it's a class. So finally, does anyone have any questions or comments, feedback?

AUDIENCE: OK. What would you do if-- what would your team have done if they were able to choose a topic or have a little bit more freedom? Was there anything in particular you talked about as a team? Or when you're talking about things you wanted to do but the client said no.

AUDIENCE: Humor.

STUDENT: Yeah, in terms of like if we were doing about cholera, things we wanted to do about cholera but we couldn't, like death is not in the game. We can't make it like overly morbid or anything, not that we-- But there are certain things, like we wanted to have more options.

But since the game is sort of oriented towards-- or rather, the options have to be oriented towards what an individual can do. So the game is supposed to teach kid to young adult as an individual how can I prevent cholera, which is like you wash your hands. You boil your water, that sort of thing.

But things like put basically bathrooms, the sorts of things that a government or the group of people as a whole could do, but an individual can't really do, were options that we wanted in the game, and could have sort of interesting mechanical effects within the game, we couldn't do, because it doesn't make sense. I can't teach you to make bathrooms. You can't, not really,

as an individual.

LAUREN: Yeah, and sort of as Jen shouted out, tongue-in-cheek humor. There was a lot of time during the game where we're like, oh, we should have this happen, but it was offensive. It was funny.

It would be funny to an American audience. But because it was for an audience in Ghana, we changed the-- we didn't know whether to use village or town or city. And so we asked Pablo. And he said use the word locality, which to everyone playing it, play testers, we're like this is a really weird word. Why does it say locality?

And even we forget to use the word locality. And we call it a village accidentally. And so we'd want a little bit more freedom there and sort of make it maybe a more humorous game.

But we didn't really-- we were really afraid of being offensive. That was like one of the first things we talked about when we started making the game is how do we make sure we don't offend anybody accidentally.

AUDIENCE: Any other questions? We're done? All right.

LAUREN: Thank you.

AUDIENCE: Thank you.

AUDIENCE: Thanks.

LIZ RITA: We're the *Saving Gora Gora* team. I'm Liz, Rachel, Kevin, Justin. There are others in the audience. Go to the beginning. Yeah.

So to start us off, I'm going to talk about, let's see, the goals that we had from the beginning. Our product is about cholera and cholera prevention. And our audience was Ghanaian children. So we had to find a way to connect Ghanaian children to this information about cholera prevention, positive behaviors, proactive things they can do in the community, while also having a fun game that they can relate to.

What this ultimately meant is that we had to do a lot of research, because we didn't have a direct connection with our audience. And we couldn't directly play test with children from Ghana, ages 8 to 13. We had to make a lot of assumptions, which ended up being wrong after we talked to Pablo, specifically about them.

So we did some more research. We looked at Ghana, cartoons targeted towards children in Ghana. And from those cartoons, we looked at the color palettes, some behaviors within the cartoon specifically, and names and sort of character designs from the cartoons.

And we got Ghana research from documentaries and stuff, also informed decisions that we used to change minigames, which we'll talk about later in the presentation.

A quick overview of the narrative-- you start the game. There is this monster hiding in a bush. You go to check him out. It turns out that he's being persecuted, because everybody in Gora Gora is sick. And they think he's the cause.

And Sal is actually a cool guy. You're a bunny, and your name is Kojo, and you're a cool guy, too. So you're asked to prove that he's innocent. And you take that up, because you're an awesome person.

The gameplay ends up being that you explore the town, looking for ways to prove that Sal is innocent. And you do that by unlocking these minigames and getting clues from the minigames after you defeat the minigames.

And after each minigame, you can go talk to the mayor, or you have the choice to, because he has an office. And when you have all the clues, you finally convince the mayor that Sal is harmless, which is great.

KEVIN WANG: So our first minigame dealt with water filtration. And the concept behind it is that friends and neighbors are bringing containers of water to Kojo's mom. And they want to see whether the water is safe to drink, or whether they have to purify it beforehand by boiling.

So that, and when we initially designed our game, it turned out to be water filtration kind of factoring, in which bottles of water were going down a conveyor belt. And you were asked to remove the bottles that were unpurified and that needed to be cleaned.

So as you see on the left, that was our very first iteration. And I guess the issue with that, as Pablo brought up to us, was that it wasn't relatable to the kids that we were trying to target, because these kids likely won't ever see a facility that has these conveyor belts, that does this kind of purification.

So we had to go through several iterations to figure out a way to make the game a lot more relatable. And our final product is on the left-- or on the right, sorry. And as you can see, it's

outside one of the homes in the village.

And your mom is just standing behind a pot of boiling water. And now, you just have plain containers, bowls, buckets, tanks. And your goal is to kind of see where the water is coming from, how warm it is, and try to figure out whether you want to boil it or not.

So some challenges we faced were, in our first iteration, it was more of a real time game play. You were kind of rushing against time to pull off all the unfiltered water, as all the bottles continued to scroll to the right, regardless of how fast you were going.

But then we realized that took away kind of from the educational aspect, because it forced the player to just kind of hurriedly just pull everything that they can. So instead, we changed it to a styled gameplay that was kind of at the player's pace. They can do everything as fast or as slow as they want. And that, I think, gives the player a better chance to kind of learn everything there is to learn from the minigame.

RACHEL WANG: Cool, so the second minigame is a water collection game, where you're asked to help your friend [? Korku ?] find where to collect water from. And so with this development process, it was kind of similar to what Kevin was saying about making it, like have as high of a learning value as possible.

And then another aspect of this game was making it fail-safe. So obviously, we thought there was one right answer that the player should take away from it. But we didn't want them to fail and then not win the game at the end, if they were to not choose the right location.

So one way we dealt with that was, every time the player chose a wrong location, there'd be immediate feedback, saying, oh, this isn't such a good place to collect water from because. And so as you can see, this example is, if you were to collect water from the river, you don't know if there's excreta from upstream or if someone's peeing in the river right now. So then it'll ask you to try another place.

We also did a feature cut pretty early on, which was also finding someone to dispose of your excreta. And so we focused on the water, which would hopefully focus the player's attention in figuring out where to place water and really think about the water aspect and not so much as another aspect of the game.

So the other challenge was, as Liz and Kevin already pointed out, was there's a huge cultural difference. So again, I have like a before and after shot of what our landscape looked like. And

so originally, there was snow-capped mountains, a lot of grass. But from our research, we saw that this didn't really look like Ghana landscape.

We had a water treatment facility, which originally was the correct place to collect water from. But we eventually changed that to be a well. And so that's more realistic in terms of where people collect water in Ghana. Additionally, at the end there is a task bar, so it provides you more feedback. If you clicked on any of the buildings it'll provide you feedback about what that place is, and whether or not you should or shouldn't be placing water there.

JUSTIN

MARTINEZ:

Our final minigame is focused around teaching the symptoms of cholera and the importance of early prevention and recognizing the early signs of cholera to prevent the disease from spreading. So the premise of this game is that the doctor in the town is very busy. He can't go out into the town to actually check on patients to see if they're seeing any of the early signs of cholera. So he asks you to help him by going out and talking to some of your friends in town, and then asking them if they're having any of the symptoms, and then ask them to come in to the doctor.

We initially started off by having this game be a game where you help the doctor in the doctor's office. And he's diagnosing patients, and you're sort of diagnosing the patients with him. And we realized that this would quickly turn into sort of a quiz-type of game, and it would really detract from the learning aspect. It would essentially be the same as just reading the systems out of a pamphlet. So we decided that by adding more interactivity by forcing the player to go out into the village and actually find people and talk to them and then ask them to come to the clinic would be a lot more engaging, while still maintaining the same educational aspect about trying to search for the right symptoms.

Additionally, we made a change where we had the player go talk to some of the villagers that were adults. And then we realized that-- after speaking with Pablo, that the social hierarchy in Ghana is not necessarily that of the United States. And this sort of interaction, where a child asks something or sort of demands something of an adult, isn't really something that would happen, just because of the stricter social hierarchy. So we decided that it'd be more appropriate if we had the child go talk to some of his friends that were of the same age. And then tell them, since they're on the same sort of level, to go to the doctor's office, rather than talking to elders.

Some of the challenges that we saw were that this game is heavily dialogue-based. It's almost

entirely interacting through our dialogue UI. And so we needed to do a lot of iteration on that in order to make sure that it was clear who was speaking and give the appropriate choices so that the player can make meaningful decisions while having his interactions with the villagers.

And finally, since it is so dialogue-heavy, there's a problem that comes when making a game for children when you have a lot of text, because they might not be the strongest reader. So we focused a lot on making sure that the vocabulary was simple, and we tried to keep the sentence structure as simple as possible, and straightforward, to get the message across to children.

So some of the overall challenges that we faced while doing the game development was mastering the states with Phaser. This is sort of an intricacy of the Phaser engine itself, where when you spawn off the different stages of the map, making sure that we can maintain the data across all these different states, so that there can be a sort of progress through the game. And finally, we want to make sure that the game wasn't too difficult. For example, with the facility game at the beginning when we had the very fast-paced clicking, and then you would fail almost all the time. We wanted to make sure that it wasn't too difficult, and that it was accessible.

The cultural disconnect we touched a bit on in some of our previous games, making sure that the target audience will be able to play this game and relate to it in order to pull the message from it, and making the play meaningful. We wanted to make sure that the game play was still getting the message across, and finding a good balance between education and play.

RACHEL WANG: So now I'll talk about how our teamwork worked out, and what our process like as a team. So we used four main tools. The first one being Trello, to keep track of our tasks and progresses. So, we'd put all our sprint task lists, our product backlog, on to the Trello. People would claim the different tasks, and then be able to say if they were doing them, if they were done, or if they haven't started. And then that means we could track how people's progress was.

The second one was weekly meetings. So, every week we had a meeting to either do some research about Ghana, talk about how we could improve the game play and make it more meaningful, if we should cut some features, and whatnot. So, these meetings were really great for quick and efficient decisions, and also really important decisions.

Our third tool was Slack, which some other teams talked about. But it was really great because, first, they had Google Drive integration, so you could see when people were pushing

code if you should be pulling code. And then, in addition, we could have different channels. So we had a separate coding channel, where all the coders could talk to each other, and that might not be relevant to other people. And then having a whole team channel was also efficient, because it would be faster response than email.

And then our final one was Google Drive, which is where we have all our assets, had all our dialogues, our characters. And this worked out to be a really great tool for collaborative storage.

Some downfalls of this was-- I'll talk about the second part, was basically with all these tools you need to make sure that everyone's actually using the tools. It's not actually effective to be using Trello if not everyone's using Trello. And we found this to be the case, not everyone would be updating their progress on Trello, or not everyone would be on Slack. But some mitigations we had to that is, in like Slack, you can mention people's names, and then it'll send them an email if they're not actually on Slack at that time. We did have to remind people to be updating the Trello board or updating their progress in Slack, or just ping people individually in email. But I think we worked it out.

Something we could have improved on was enforcing deadlines, and having deadlines. As Kevin will talk about, we had different roles between coding and art designing. So, obviously the art assets had to be done before you could really code things and put it into the game. So in that aspect, we should have probably had clearer deadlines so that we could stick to a stricter schedule. And then, in addition, we should have probably had a bit more frequent communication with the entire team. So we did have a lot of communication, but sometimes not all the team members were present, and we should have probably improved a bit about sending out meeting notes and catching up people who weren't at the meetings when we made certain decisions. And to that regard, something that we could just do for next time is, send out meeting notes, and keep everyone up to date, and possibly even have smaller meetings where not everyone is present. So I guess in terms of what went right, Kevin will talk about that.

KEVIN WANG: So, as Rachel was saying, we separated a lot of the tasks of the project between the team members. And since we had seven people, we designated four to coding, two to design and art, and then a few others to the other aspects such as narrative, project management, and sound. So, that actually helped us a lot, in that having only four coders rather than seven really cut down on the number of merge conflicts and issues that we had in previous projects.

Because, obviously, the fewer number of people working on each file, the less likely it's all going to break when people merge their stuff together in GitHub.

And with the art designers, it actually helped to have just two. Because, although they had to do more work, it was easier for them to collaborate and make sure that all of their art followed the same theme, rather than coming from three or four different sources that look slightly different. Eduardo and Liz were very capable in making sure that all the art had the same kind of color palate, and the same kind of cultural theme to it, which helped, I think, unify our game in the end.

And, as Rachel also said, we had semi-frequent meetings, mostly at the end of the weeks, just to catch up and see how everybody was doing on their parts, and what we were planning on doing for the next week. And it also served as a good time for us to really sit down and get some work done, whether it's drawing some new assets, coding a new minigame. And having most of the people there together at the same time allowed us to really bounce ideas off each other when we were implementing this, which made things a lot more efficient.

And lastly, we realize that at the beginning, we had probably more than we could take on over this eight-week period. So we did a good job of cutting features early. And the main one we did was cutting the fourth minigame out of our game entirely, so that we could really focus on making the three minigames that we have currently really robust and really relevant to the ideas we were trying to convey.

RACHEL WANG: So moving forward from this where hopefully we would be able to find more of our target audience, our focus tests helped a lot whether it was speaking to Pablo, other MIT students, people in the class. They gave us a lot of good feedback about the game. But obviously we didn't get a chance to speak with Ghanaian kids, which is our target audience. And that could definitely help if we had a longer period and were able to find even kids ages 8 to 13.

In terms of iteration, I think we could still focus a bit more on balancing the difficulty. And then we mitigated that by doing things like what Justin said about changing the text dialogue, and also providing a lot of help tutorials or instructions throughout the game. But that's something we could always think about, so that even kids younger than our target audience could access the game. And then, obviously, just making sure the game is culturally relevant, thinking about word choices, thinking about assets, something we could always improve on.

And then, finally, to actually keep moving forward we could implement another minigame, find out other things about cholera we really want to get across, and add other adventures to the game. But the game right now is pretty contained, and gives, we think, three valuable lessons about cholera, and three very concrete preventative actions that you can take, and bring back to your village and your family. So I guess that's what we would think about if we were going to go for it.

And now, we'll move into our demo. So, I guess, if there's any questions first we can take those, or else we'll be looking for a demo-er.

AUDIENCE: I have a question regarding how you used like the structure of like having the many minigames inside of a big game. I wonder like does that help splitting the tasks up among different members in the team and make the process better? Or did that add more difficulty to finish all the contents and correctly tie them together?

RACHEL WANG: Yes, really, I think the modularity really helps, so the three of us took the lead on one minigame each. But then obviously there were components that cross all the different minigames. So, for example, our other coder worked a lot on the conversation and dialogue UI that shows up in all the minigames. And additionally, at the beginning we did run into an issue where we thought, oh, really the game is just three minigames. So we worked to come up with an idea to tie everything together. And that ended up being an inventory clue kind of thing. So you could go around town looking for clues that you needed to unlock the minigames. So, for example, to play the water collection game you need a bowl to collect the water. So you would go around town and find someone-- maybe someone was like wearing a bucket on their head. And you'd be able to collect that bucket from them and then play the game. So, in that sense, we were able to come up with this idea that could tie together all the different minigames. Does that answer your question?

AUDIENCE: Is its mother a cat?

LIZ RITA: Yes, [? Bunnykins' ?] mother is a cat.

AUDIENCE: All right. [INAUDIBLE]

[LAUGHTER]

AUDIENCE: So, you cut the minigame number four?

JUSTIN Yes.

MARTINEZ:

AUDIENCE: About when in the project did you do that? And about how much work had you actually put in to that minigame before you decided not to complete it?

JUSTIN That game did not make it past the paper prototype. After we did the paper prototype, we
MARTINEZ: realized that, that game and the symptoms game were actually very similar. And so we [? introduced ?] some of the storytelling aspects into the doctor minigame that we pulled from the fourth minigame.

PROFESSOR 1: Thank you.

[APPLAUSE]

[INAUDIBLE] will volunteer. Thank you.

[LAUGHTER]

AUDIENCE: I sure helped!

[LAUGHTER]

From our previous discussion, I wonder if I can take his bucket.

LIZ RITA: [INAUDIBLE] like, go closer.

AUDIENCE: Oh, gosh.

I don't know if [? Korku's ?] is going to get that bucket. Maybe I'll keep it. Wow, I already have two quests.

[LAUGHTER]

[LAUGHTER]

Never play with fire by yourself.

[LAUGHTER]

Whoa, look it's turned into a cat.

[LAUGHTER]

Ah, cool. I mean, if I say no, I should come back later, don't I?

[LAUGHTER]

Thank the bowl, and click me when you're done. All right. Well, I know where to go, but we'll do this anyway. There's no water there!

I guess that makes sense. If there was water in the house, I wouldn't need the bucket. I do all the wrong things first. It looks like there are a lot of bacteria in the lake. I have really good eyes.

[LAUGHTER]

You-- I don't want rusty water. Oh, the bathroom, that's a great place to get water. I won't go there. Yay! I have water. Can I go swimming?

Yes, yes, I can. I forget what you wanted. What did you want again? Paper. I know-- no, I don't actually. Oh, I guess I do know something. Ah ha!

[LAUGHTER]

I guess I need to go in the houses now. That's my guess. Oh, cool. You-- I better boil that.

RACHEL WANG: You might want to start the game first.

AUDIENCE: Oh, start the game. I'll do that first. I like this water. This is great. I just know that the well is good, because I was just there. Pump and clear, well, rust was the problem before, so where am I getting this hot water from the pump? Discolored. Discolored. Well, I am going to be safe, because I don't have to gather firewood.

I'm running out of firewood. Now I'm going to be in trouble here. We'll go for it. It's from the pump, how bad could it be?

[LAUGHTER]

AUDIENCE: Is this the rusty pump?

[LAUGHTER]

AUDIENCE: Oh, right. That's the bad stuff. The well's the good stuff. Oh, that's [INAUDIBLE] pump. Ah, whatever.

AUDIENCE: It is the cholera that you've got to test in this game, right?

AUDIENCE: Hm, I only have three firewood. Let's just do this. One of the next one will be clear, I'm sure. Well and clear. There we go. We'll boil the rest to be safe. All right. Throwing up and have diarrhea and very thirsty. Oh. So in other words, you've seen the monster. You should go see the doctor, because you have all those symptoms. All right. Where are my check marks? Oh, look I got three clues. Is that good enough? Let's go out to mayor. I don't have any more spaces for clues, it must be good enough. Nope.

LIZ RITA: [INAUDIBLE]

[LAUGHTER]

AUDIENCE: Yay!

LIZ RITA: I just jumped over that office.

AUDIENCE: I guess the repetition's not a bad thing, is it, for your demographic? That's good.

[LAUGHTER]

PROFESSOR 1: That's it. Thank you.

RACHEL WANG: I think that's it, that's all we have. Are there any last questions?

PROFESSOR 2: I actually do have one question about the music. Because you have it on the credits. Is this like a piece that the person wrote for you? Or is this something that you downloaded from the website?

RACHEL WANG: We downloaded it.

PROFESSOR 2: You might just want to check their website to see exactly how they want to be credited. [INAUDIBLE] title of the song they want used, or something like that. But if you've already done that, that's good.

PROFESSOR 1: Great. Thank you.

RACHEL WANG: Thank you.

[APPLAUSE]

PROFESSOR 1: All right. We're done! Thank you so much.

[APPLAUSE]

We really enjoy teaching this class every year. We really hope you got a good experience for it. I liked seeing a lot of the presentations, and I liked the changes that we saw from rehearsals to today. So there was a lot of really good information in there, both in the games and in the presentations. Again, if you have any questions about future classes-- we don't actually offer a game design degree at MIT, but we do have a concentration of comparative media studies. If you take any four CMS classes, you get a concentration. It can be from this list here. There's information on your desks about other things you can do with games at MIT. Course six students, if you're interested in doing UAPs, if you haven't thought about that, start thinking about that now. We could help with mentoring and advising on UAP projects and other kinds of research. And I'd like to give a thanks to the instructors, Phillip, Sara, Andrew, for all your help with running the class this year. So, thanks again.

AUDIENCE: Thanks, Rick.

[APPLAUSE]