GUEST SPEAKER: You ready? The project that I chose to work on is a 2D puzzle game, and it's based on a puzzle version of Tetris that I've created over many years. I programmed in different formats. But I haven't figured out a good way to deploy it, so I thought maybe Unity would be a good option because it claims to be cross platform.

So, anyway, so I'm a one-person project, and that's OK. It was a fascinating learning experience, and I made decent progress. The game that you're going to see is not very-- oh, something's wrong with it. The Canvas needs to get away from the playing board. Is there a way?

Anyway, so the game is very low-end graphics. It's very bare bones, but conceptually, it's an implementation of Tetris, with the variation that only certain pieces are used. And it has a puzzle objective, so you could actually win the game or solve it. The objective is-- the simple objective is a wipeout-- where you want to clear the board exactly, so that the last piece clears all the row. Is everyone familiar with Tetris and the row-clearing mechanism [INAUDIBLE], I hope?

So you'll see it and figure it out if you're not. If I can-- I knew I wanted to do a 2D game, grid-type puzzle because that's what I've been working on a lot is developing various ideas for logic puzzles that use 2D grids. And I thought Tetris might be too ambitious, so I was looking at the tutorials I found on Unity sites.

There were some-- there was one for Minesweeper, so I was originally going to do that. But then when I found the Tetris tutorial, I said, oh, maybe I can go with this. My first step was to implement the Tetris tutorial-- implement the code from the Tetris tutorial, which that code was for a full Tetris game with a 10-wide board and all the pieces that are standard Tetris. But what I wanted is just a couple of pieces that work well for the wipe out games that I do-- the puzzles. So I just needed a T-piece and an L-piece, so that's all I implemented.

So I adapted the Tetris code a lot. So let me just jump a bit and show you a quick demo if I could move my mouse. Oh, it's two screens. So before-- oh, yeah, the game is already going. So you can select which width you want.

The board can be-- I'll show you how that moves-- the board can be seven-wide, or we'll start with five-wide because that's easier to finish the wipeout. 10-wide is really slow. It's possible to play this wipeout game with wider boards, but it just takes so much longer, and it's tedious. So I find that five, six, and seven are the most fun and challenging levels, and the game can be very challenging. You need to learn patterns, as you'll see.

So then you have the choice of three piece options, as either a T-piece, which I'll start with, or you can do an L-piece, or the T and L together, which under the current limitation, they're randomly chosen, which makes the game extremely hard because it's much easier to solve if you can plan ahead a little. Because you get to the point where-- well, I'm jumping ahead.

So let's do the T-piece and show the game in action. So this is what it looks like. We start with a piece down at the bottom. It manually drops-- are my cursor keys going to work here? Yeah, so there's basically four-- the four arrow keys are used to control this.

The pieces fall like in Tetris, so you can rotate. There is only one rotate, so right up arrow is rotate clockwise. And you could drop a piece, and then the rows clear. That's the Tetris mechanism. So now this takes a little longer to do a wipeout, but I'll show you the basic idea.

The trick is that you learn patterns, so you'll start to see a pattern here. So I can clear that, and now I can clear a row by putting this. It takes four pieces to clear a row. So eight more pieces and I'll be down-- all but the white dots. So let's just plow through that, so you can see what happens.

[LAUGHTER]

One more. So it's-- you see the pattern, so part of what I think is fun about this game is learning these patterns. It's challenging to figure out how to learn pattern. There sort of like macros. Oh.

[APPLAUSE]

Look at that. Thank you. So let's just try one-- I don't know how much time I have, so I'll just do one more, so we can discuss what went into it. There were a number of steps forward in the learning experience. First, was implementing the Tetris game, modified to use only the two pieces, so I started with just the two pieces.

And then, the next step, I wanted to set up controls to let me select. So the first thing I did was the width buttons-- to modify the width. There's just one thing that's control, which is the right border, as you saw it moves around. So learning to move that was my first learning experience. I was happy when I got that to work.

And I had to figure out along the way how to make buttons work-- how to connect to them and run the scripts for buttons and things like that. It was very challenging, and I had never programmed in C# before, so that was a challenge in itself. But I made progress with that.

Once I figured out buttons, I said, I'm running with buttons. I'm going to use buttons for all they're worth. So I try to do the Piece Selection buttons and then the Start buttons. And the Stop button doesn't quite work. It sort of works, but it seems to create something wrong in the data structure, so that when I run the next game, it crashes. So that's not good, so I avoid the Stop button now.

So let me show you the six-wide board with the T-piece also. Maybe I should do the L-piece just to show you a different piece because I may not be able to get beyond this. So how do I want to do this? So, yeah, this will work.

So one of the patterns you learn is that with two L-pieces together, you can make a two-by-four rectangle. So if I put this one down here, then the other completes the two-by-four rectangle. And that rectangle, you can see, fills in the remaining pieces, so that that wipes that one out.

[APPLAUSE]

And I won't have time to finish it, but I'll just show you what happens when have the end of pieces. Let's do it with, say, the five-wide board. So you see-- so we randomly get an L- or a T-piece now. And notice there's only one kind of L, which is the L L, not the reflected L. So how do I want to do this?

So part of the problem here is that this L will not turn around to go in the other direction. Now if I'm fast, I can slip things in before it falls.

[LAUGHTER]

So I implemented the drop, so that it drops-- let me just get my piece ready-- I want it to go there. So it will drop, but not freeze the piece. So the code actually waits until there's a fall. You see that every second the piece tries to fall another step, and the piece freezes when it can't successfully fall in response to a fall action. So this isn't great.

PROFESSOR: Anybody have any questions?

GUEST SPEAKER: Yeah, you can start asking questions while I muck with this. Nope.

AUDIENCE: Have you actually completed it with the RAM [INAUDIBLE].

GUEST SPEAKER: Oh, yeah, absolutely. It's just a little more tedious, like I said. It takes-- no, it's definitely doable. It just-- sometimes it takes multiple tries because-- here's a trick. Because I'm preparing for Ls. So I didn't get an L, but I could put a T in here. If I get an L next, good, then I could put this here. Now if I an L, I'll wipe out all those three rows. So I want to do things like that.

[LAUGHTER]

Let's see. There-- and this could work. I get another L. Nope-- didn't. So how do I want to do this? So you don't always get the piece you want, of course, because it's random. Let's see how this works. This will get it down to almost one row.

[ARGH]

Boo. I didn't want more L-pieces, but I have to decide what to do with it. I guess I'll just bite the bullet and do that. So now I'll try again. Yes, now if I-- whichever-- I'd prefer a T-piece now. So I can get it down to one cell. I'm close to a wipeout, but not what I wanted. If I can get a T-piece, so this is good. I like getting-- making the shape of a T-piece at the bottom because there is a way to win if I get the right pieces. So, L-- I need a T and an L. let's hope for an L next.

AUDIENCE: Yay.

GUEST SPEAKER: Yay.

[APPLAUSE]

So I think this is a fun game. What was this-- there was a second thing. The scripting was a big challenge, and learning to get the buttons to work properly. Oh, the text-- that's what I did last night. I figured out how to enable text.

Someone gave me a hint that I wanted to set enable true or something. But it was very hard in one script that detected the wipeout. So I wrote the code last night to detect the wipeout by checking for all the rows cleared. There's a grid that's hidden behind this little matrix that holds blocks. And the script operates on that, and so I had the script that checks for the wipeout. And if it finds there's a wipeout, then it needs to call another script. Well, it basically needs to go tell the text to enable, so that you can see it.

So I created the text and disabled it at the start, but then I needed to be able to programmatically, or in the script, turn it off and on, which these things sound simple, but they are not trivial when you're first encountering them. So I struggled-- doing googling-- looking for things like how to find an object. And so there's find object by type, and then there is the game object defined, which let's you find by name, which was very useful.

So once I figured that out I could look up my-- I think it was a text object. I had a text object that I created, but that wasn't good enough. I needed to get the text component itself. So I had to call get component and figure out what kind of specifications to put after that to tell it what kind of component to find. The syntax was tricky.

One website said you needed a dot before the angle bracket text and another one. But what worked was not having the dot there, so there's misinformation out there, or else Unity changed. I don't know, but, finally, I got it.

And I thought there was a method called enable that I could say enable or disable, but turns out, it's a variable. It's a, I guess, a component variable, or something, that you can set to true or false. Finally, I figured it out, and I was able to get the congratulatory text to appear

[APPLAUSE]