PROFESSOR: So let's go ahead and start Lecture 6, which is all about strategy. And we're going to try to be numerical about it because numbers make us feel smart.

So strategy is a nebulous thing. It can't be easily defined, and very many people spend all day talking about it without giving any statistics or numerical results. Which I think is quite a shame because, especially in this Battlecode, we can actually do computations and say, OK, if I build this many of x, I can build this many of y. And we can get to some pretty interesting results.

I'm going to start out by just having a seat here, and I'm going to talk about some sort of general stuff. And then we'll move straight into numerical strategy, which I've got here on the right of the view screen. And you can see, like, I've implemented Battlecode in Mathematica to a limited extent for the express purpose of showing graphs like these. So we'll get to those.

Let's talk about Battlecode a little bit. The way I see it is that strategy can be expressed as having a triangle aspect to it because you're basically picking between a technology route, an econ route, and a military route. And then among military, I like to try to draw another line.

And you can sort of imagine this triangle as being on a plane. This is on a plane that's like perpendicular to this line, indicating that you can sort of be in any of them. I could have drawn it as a triangular prism, but I chose not to.

So then this triangle, you can have a defensive military strategy, or you can have an offensive one. Offensive is spelled that way. And I also like to populate this map with even more things to make it less clear what's going on. So I'm going to put over here large map, and I'm going to put over down here small map.

Because if you're on a small map, you want to have more military, at least early on. I mean, this is all time-changing. So you could describe a strategy in time as being like, OK, first, I'm going to do a lot of military, and then I'm going to get some econ, and then I'm going to get tech. So like that's a strategy, though it looks pretty weird. That looks pretty weird, and it's crossing out my small map.

Anyway, so small map, it is really another way of saying it's far to the enemy base or it's close. So on a small map, you can get right to the enemy base. You got to be prepared if they've got a rush strategy.

Maybe what you ought to do instead of saying, OK, I'm going to use the map size, is do sort of a heuristic. Because you could imagine this map-- I'm going to do example maps here-- and this map currently identical.

Now let's make them different. This one's got like tons of mines, and this one has a little bit of mines. Obviously, they're different. I mean, if the headquarters are here and here in these two maps, well, this one is effectively a small map, and this one is effectively a large map.

So here's a suggestion. It's not very expensive to sense mines at location. It costs 10 in the MethodCosts.txt file. So you could consider saying, all right, I'm this headquarters. For the first couple of rounds, I'm going to compute the sum of the number of mines between me and the enemy in sort of a volume looking like that.

And so I'm just going to compute this number, just going to total it all up. If the map is, say, 50 by 50, then how many tiles do you need to count? Well, say you make it 4 wide, just for halibut. And then down in this direction, I guess it would end up being sort of like 50.

So you'd have about 200 tiles to look at times 10. That's going to be 10 bytecodes per sense tile, and so you've only got 2,000. You can do that easily in one turn because the bytecode limit is 10,000. So then you could very quickly get an estimate of how easy or difficult it is to get to your enemy right off there. And so then you could find your spot, say, OK, I'm here or here on my Tech Tree. The system doesn't have to be incredibly complicated.

So let's delete some of this mess that I've made here so that I can make some notes about the differences between defensive and offensive strategy. Later on, we're going to see the number of units that you can get if you're pushing for an early advantage, and that's going to sort of tie into this.

So there's a time aspect, where you're saying, if you're defending, you have a number of advantages. One is that you can have additional units up at the time when the enemy arrives. Another is that the pathing is much easier because you don't have to go anywhere. You can establish a better concave in theory because the enemy might have to defuse mines.

It's worth noting here that, of course, you're not guaranteed this advantage because the enemy knows where you are. You're sort of around your headquarters. I mean, it's generally true that if you can recognize the enemy is around his headquarters, so he's all sort of clumped up here.

Well, it's pretty easy for you to say, all right, I'm going to stay at a distance r from that enemy, and I'm just going to sort of group up there. And as soon as everybody is within a distance r and not farther than a distance this other letter, then I know that I'm in a good position to attack all at once.

And so if the enemy's just sort of hiding there in its hole, then that's also a vulnerability as well as an advantage. So you could imagine that case. And you could also imagine the case where there's a ton of mines, and so the minefield is like this giant, thick thing, which is filled.

And so you're defending over here. And yeah, you're defending here, and the enemy's coming at you. And they kind of have to go in a line because it's such a big mine field, so you can just sort surround them. Yeah, I meant to do that outlining on purpose. Oh, my goodness. Who can use Microsoft Paint? I certainly can. Let's see here.

So yeah, there's this concave issue. And then there's another issue that you can have the base take hits. So your base has 500 hit points. So you can use that as a sort of buffer to try to protect your robot. It's a very good idea, but it would spread out the damage a little bit. So in the event of an emergency, it could be superior to being farther from your base.

Also consider that it's easier to heal your robots because you're going to have this area sort of established. You're going to know where things are. You're going to have a safe path to the nearest medbay that you might build. And later on we're going to look at the effects of building medbay versus other things. It's going to be really good.

So these are just sort of a first look at things. There's some other stuff in here, like about tactical withdrawal. When do you realize that everything's gone to shit, and you've got to back out? What's the value of preserving your units versus the penalty of having them not in combat? They're in transit between wherever they are and the medical bays that you've built.

Because you could easily imagine filling the whole map with medical bays just to reduce that transit time. And would that be worth it compared to just not having them go to medbay or only engaging in battles within a certain range of the medbay? So you might actually be retreating to a range rather than directly to the thing itself. These are things that you'll probably end up thinking about if you haven't already.

Let's jump straight into some simulations. So what I've done in this model-- let me zoom in just a little bit-- is I've got a system where you can build bots. Bots will die if you have less than zero resources.

You can build generators, as you can see here. I mean, I'm not expecting you to read this. But just showing you the parts that are in the model so that when you look at the model results, you can see for yourself whether you think they're realistic. So you can build generators.

And as in the game, the generators will provide you an additional 10 resources per turn. And the cost of building a generator is 20 times the number of encampments that are currently built. And here's plus 1 for the additional one that you're building.

Same thing applies for suppliers. It's got the same build cost. But instead, it's advantage is that the spawn time then becomes this number. Note that this is different from what it was in Lecture 1.

In Lecture 1, the spawn time was 10-- it was this. It was 10 times 0.95 to the power of supplier number, where supplier number is the number of suppliers you have. That's what it used to be. But it's been switched to this number in order to make it slightly worse, because it was too good.

This and the other changes have been updated on the Battlecode data sheet that I made that you might remember from the website. So if you're using that, make sure that you've updated recently. And when you download it, hit F5 or Refresh, because it has the same file name as before.

So your computer might not download it again unless you specifically tell it to. That's just something to worry about. Because it's terrible to be on a bad version of the specs because then you think I'm wrong, and that's just very, very hard to handle for me.

So here's your spawn time here. So that's what your suppliers are doing for you. And so I've made this model that tries to build these things, tries to build robots, and it kills the excess ones. So we're trying to reach a large number of robots. to Zoom out just a little bit here.

So you can see here that we've got power over time, and this is for 500 rounds. I've got it so that you can change the length of the simulation. And we're going to see quite a number of simulations here today.

So you could see here that, at first, you start out with no power. I don't know if this is 100% realistic because I think in the game you might actually start out with full power to begin with. But that's a transient effect. I wouldn't really worry about it.

We're physicists here. We talk about orders of magnitude. Yes.

So you could see here that initially your power decays upward to some relatively stable value. But it begins to decrease as you build robots, because as the number of robots increases over time, the upkeep is increasing.

And so you're decaying downward to stable values as you go. And then it becomes a stable number of robots as they are produced, and then they also run out of energy. So I've also plotted down here the maximum number of robots that are achieved within a simulation. So this shows that you hit 40 robots at round 391.

We're going to end up looking at this quantity and trying to see how we can make the most robots at the shortest round number. Or how we can have a trade-off between maybe getting the most robots at an early number or getting more robots at a later number, which would embody the difference between a rush strategy and an econ strategy.

I've also plotted the spawn time so that you can see how the effect of suppliers is changing how long it takes for robots to spawn. In this example, there aren't any suppliers, so the spawn time is flat.

So let's change the example a little bit. I'm going to zoom out just a tiny bit, making everything illegible, so that you can see also the inputs to this function that I've made. And so I'm going to make it so that the bytecode usage is 5,000. And what effect should that have?

It should decrease the number of robots that we can sustain. And sure enough, we see that when we run it, the maximum number we hit is now 27 robots at round 261. So we've hit there sooner, but it's a smaller number. It's just worse in every way. You want to save bytecodes. So let's go back to that example.

Now, generator built rounds and supplier built rounds is the number of rounds between every time it will attempt to build a generator or a supplier. This is the basic version that I started with. So let's start by building generators every 10 rounds. This is assuming that there are enough generators around for us to build on.

So you can see here that the first generator is built around here. There's a tiny dip, and here's another and another. But the generator doesn't actually kick in and start to take effect until 50 turns later because there's a capture delay, and it doesn't do anything while it's being captured.

Then once these are captured, the amount of power you have skyrockets. So this was the previous maximum of 200. And now we're up to 2,000 power in just 500 rounds. This shows that you can really amplify your economy rapidly.

You'll notice here that the sawtooth behavior is becoming increasingly severe. This is because each one of these motions downward indicates the purchase of yet another generator, which costs more than the last.

Because as I mentioned before, the cost of capturing an encampment is 20 times the number of encampments you have already. So that's going to linearly increase. You could see that it's recovering faster and faster though because, as has been mentioned, the generators do produce 10 energy per turn.

So here we can see that it's not hitting any limit to the number of robots that can be built. And if we extend the simulation out to 2,000 rounds, we can see that it just continues on as it was before, and the number of robots is unlimited.

So if you do find a map that has a whole lot of encampments to capture-- and we haven't specified that there's a maximum number of encampments as far as I know-- then you could end up just continuing to build these things. And then it doesn't matter how terrible your micro is or how terrible your tech is because you'll just overwhelm the enemy. Hey, look, at round 1991, what a good year. That's pretty special.

So let's go back down to 500 where we can see what's going on, and let's change generator built rounds to 1. So now we're going to try to build a generator every round. But look at here. These red dots indicating the completion of generators are non-uniformly spaced.

That seems to indicate that it's not able to build generators every turn. That's because there's not enough power. You can see here that the power is as close to 0 as it possibly can without destroying robots from bringing it down to negative values.

So you can see here that you can now get to have more robots than you otherwise would have. No, that's not true. You can't have any more robots because we're still limited by the rate at which they're being produced. They're only being produced one every 10 rounds, and that's really cramping our style, as it were.

So let's increase this back up to a million so that it's not building any more generators. Let's try building just suppliers. So let's build a supplier every 10 rounds. As you can see here, here's the money being expended on suppliers.

And pretty soon, even before round 100, we start hitting a point where the number of robots we have, which is accelerating over time as the spawn time is decreasing, the number of robots going up is preventing us from getting enough power to continue building suppliers. So we're actually capped.

This behavior, where the power goes down and crashes, is going to really hurt our ability to play the game. It becomes exacerbated the faster you build suppliers. Because you see here, I built these suppliers, and then, bingo. I've only built a total of-- let's see here, supplier number, I've built a total of eight suppliers within the first, I don't know, this looks like 40 turns.

And now I'm not going to be able to build anything else for the whole game, because the next encampment that I buy would cost 8 times 20 is 160, plus 20 for the next one, is going to be 180. I simply can't afford that if my power is never exceeding 170 over here. So that's pretty much going to be a crippling blow.

So maybe, as my player, I need to do one of two things. One, maybe I should build suppliers and generators. Two, I could notice that maybe the enemy has killed a bunch of my generators, but I still have suppliers left over.

Have you thought about that? Maybe some of your suppliers will have to destroy themselves to make room for more power generation. It could be something that happens to you less than not ever, or which is to say more often than not ever.

So let's make it 10 and 10. So we're trying to do some of each. Well, it's not really working. It's like building some, and then the others because one of them is taking precedence over the first. Maybe we'll try building these every 20 and these every 10. Well, now it's building some of each. This is not so bad.

So now we're getting to 76 robots at 496. Our power is going up, but we're not so much wasting it as we were before. And we can see that the spawn time has gone all the way down to half. Now, this does assume that none of your robots are being destroyed.

If you're actively losing robots, it makes perfect sense to build only suppliers because, at that point, you can clump up your units and lose them, which will give you the money to continue building units. It's only when you've favored suppliers and then not attacked that you're wasting your effort and time.

So this is something that I saw in some of the scrimmage matches that I've been looking at, because we devs have access to all kinds of secret information about you, including the performance of your players in scrimmage matches. And so I've been downloading these maps and looking at them.

And in these matches, I've tended to see that, a lot of times, we're favoring suppliers but not attacking. Other times, we're favoring suppliers, and we're just sending our units one by one to be killed by the enemy blob. I think that's also sub-optimal.

But at the same time, we have seen quite a few good strategies. I've seen some great pathing, and I've seen some great solving for the map. I think that there's a lot to be done there. And I have confidence that we're going to see a lot of great matches in the sprint tournament.

So let's go back to the model here and look again at maybe getting a build order. So here, the build order I'm defining as, what order do you get suppliers and generators? This way of doing it, as you see I've implemented it here, is kind of inefficient, because whichever one goes first is just whichever one hit the round counter or something.

So here I re-implement the whole system with an explicit build order. So I can say, all right, please give me a bunch of 2, and then give me 2 and then 1 in alternating turns. So you can see here-- it's not working very well-- 2 is my number for the supplier, and 1 is my indicator for the generator.

So here we do five 2's, and then we alternate 2 and 1 forever. So there are the five 2's, and then we've got a 2, 1, and a 2, and a 1. Well, that's not very good because we still crash, and we only hit a maximum of 57 robots.

So let's consider reducing this number of 2's. Well, that's much better. So now we've gotten up to 87 robots around 495. Now, if you've been keeping track in your mind of which number of robots we've hit at which round, congratulations. You're much better at that than me. Later on, I'm going to be showing plots so that we don't have to memorize how many robots we have at what round.

But for the sake of illustration, it's pretty useful to be able to say, all right, if I add one more supplier here, how does that help me? Well, if you look very carefully, it's going to shift your early number of robots up, and it's probably going to shift your late number of robots down. That's the way it's changing the shape. So you can see here that that's a very subtle change.

Let's change it so that we're only building suppliers, and then we're only building generators. So there we've built a few suppliers, and then a lot of generators. That's 62 robots.

We can keep changing it and try. Here's a few more suppliers. Well, look at that. We never got the chance to build any generators because we ran out of energy when we were building suppliers. This is the same issue that we were seeing before.

So let's explore the space automatically. This is kind of weird. What I'm going to show you is a plot of the peak number of robots that you've gotten during the simulation versus the number of rounds. And what I'm going to do is I'm going to sort of move along the space in some line, which is going to be an automatic search for these various things that we're looking for.

So yeah, so where do you want to be on this graph? You want to be up here at the upper left because that means that you're getting a lot of robots at a small number of rounds. This is the phenomenon that we were seeing before, where going from 1, 2, 3, 5, 7, 8-- you could see that it's skipping some numbers because they would have written over one another.

Yeah, I'll show it with all the numbers. Well, that's odd. It changed shape. Ba, ba da, ba, ba, ba, ba, ba, ba. Well, let's go with it. So this example is-- did I define the explicit system? This is the explicit system here. Did I set that? And then did I define it? Well, all right, that's how it is.

So this system is showing you from 0 up to 15 for every time it's trying a longer or a shorter system of building the suppliers. So if it says 0, it means it just goes straight into generators. And if it says a higher number, then it has built that many suppliers before it goes into generators.

So you could see, yeah, so that's there. And the numbers sort of start to overlap one another, so I made this little function that gets rid of the numbers if they're too close to another number. And that changed the way it looked quite a bit.

[INAUDIBLE] 2,000 at the bottom if you take that [INAUDIBLE]?

I see, I see. It was going out to 2,000. Thank you, thank you. So let's make this, yeah, let's make this just to 1,000. Strategy exploration is there. Simplified exploration is here. Yeah. So there are the numbers.

And this phenomenon is the thing that we've been seeing before, that as you have more and more suppliers to begin with, it's OK up to a point, at which you can't build any more robots. And you hit a limit because you've run out of power. OK. That's what we've seen, seen it 100 ways. That's good.

Let's switch to a mix of things. So instead of doing all one, we do a mix so that we're alternating one and the other. So in this example, we're starting with i numbers of suppliers, and then we're going to follow it up with alternating generators and suppliers.

And you can see here that the best result seems to be something. I mean, you really can't decide just based on this graph because these are maximum points. And if this is 8, is that a better way to operate for a rush strategy? Well, here's a way of addressing that question.

Here I've plotted the number of robots versus the round for this 0. And you can see here that it's always above the other peak robot counts. So that indicates that it's definitively better to start building generators before you start building suppliers, at least if you're aiming for the peak robot condition. You're trying to get the greatest number of robots.

So there you go. It's better always to use zero. And here if we compare different strategies, we can see them on the same space. And one is obviously better. This one above here that's new is much better than the one we had before. It gets us more robots at the same round number. And again, we're trying to get to the upper left of this graph.

This strategy consists of building a generator, and then building a supplier i times. So we'll do that set i times, and then we'll go into building 1, 2, 2. So then we'll be building a generator for every two suppliers. And we could see that this is much better at any point. So that's pretty interesting. That is pretty interesting.

Because then we can say, all right, I've played around with this. Maybe it's better to have three. Is that even better? We'll see, we'll see. That changes the shape a little bit. So that gives me a bit of an improvement in this region. So if we really want to have the most robots at this round, possibly the best way to do it is to start here.

And I know that these are all pretty obscure-looking, so I'm going to just skip straight on down to over here, where we can look at the effect of a rush. And we can see it very concretely that, in this example, we've got the first guy in blue.

He's building only generators for 40 rounds, and then he's going to start building suppliers. So there you can see blue is building only generators, and then he starts building suppliers. And his robot production rate increases.

Here in red, the red guy is starting by building suppliers. And so you can see he caps out at 41 suppliers, or 41 robots for a moment, and then he's back to oscillating. So he's got a military advantage in this window starting from round 100 going up to round 300, where, for a time, he's going to outnumber the enemy. And his goal is to win during this time.

So if you outnumber the enemy starting at round 100 and going to 300, here's what you have to do. You have to get to the enemy and start shooting at him. So if the map is really, really big, like if the map is 100, size 100, you don't have the pickax upgrade. So you're going to be defusing mines at 12 rounds per mine.

If you've got 50 mines in the way and 12 each, there's no way that you're going to get there. Because it's going to take you 500 rounds just to get to the enemy, and you'll never actually get to start shooting.

But if you can find a straight path to your enemy, even if the map is 50 units long, you can get there and start capitalizing on your unit advantage before his econ play kicks in. So there you go, a very concrete example of the effects of rushing or playing an econ.

At the same time, you could definitely imagine intermediate strategies. So here I'm just going to go ahead and plot a third strategy, which maybe I'll make in green. And maybe this strategy is a mix of the two. So instead of starting with just generators or just suppliers, we'll do a combination of generators and suppliers.

Now look at that. That's pretty interesting. Let's extend the simulation length to 800 rounds, and we'll be able to see the effects. So now we see that the green player does almost as well as the red player. It's almost indistinguishable at this distance.

And given the defensive advantage we talked about earlier, it may be enough for the green player to hold off. He'll still be able to smash the red player. And he's still got an advantage over the blue player without being that much at a disadvantage. That is to say that the green advantage lasts until about round 600.

Let's zoom in a little bit so that we can see round 600 and try to assess how much better the green player is than the blue player. So here, once again, green is building alternating generators and suppliers.

And he has an advantage over blue starting at about round 100 and going to about round 450. That's a pretty big round and that's a pretty big advantage for a pretty long time. It's only at that time that blue begins to take off and gets a serious advantage.

So you can see here that there's a pretty significant difference between different strategies and that you can achieve a significant difference of units. But how many different differences of units are important to start really making a difference? I mean, if you have 10 units versus their 11, how good is that? Is that a lot? Is it a little? It doesn't seem like very much.

Well, to answer that question, I've made yet another simulation. Because why not do a simulation? Simulations are fun. So here's a look at it. And what I've made here is sort of a very basic simulator that works in the following way. I'm going to zoom in a little bit.

So this is some combat, and I'm going to simulate the combat between 10 and 11 units. I'm going to do it a couple of times. I'll start out doing it slowly, so I'm decreasing the compute time here. So there's 11 units on the bottom, and these are their hit points. And there are 10 units on the top.

I'm going to do it once again. I'm going to start with 10 and 10. This is team A on top and team B on bottom. As you can see here, that when there are the same number of units, they all lose hit points at the same rate, and they all die at the same time, at least in my simulation.

When there's an advantage, that extra robot that you have can start dealing additional damage to his robots. And so that advantage becomes more significant over time. So if we have 100 units versus 101 units, then we'll be here all day. But the point is that you'll end up seeing a more and more significant difference between the two.

That didn't seem to work quite right, did it? But perhaps it did. It may just have the exact same result. I'd be quite curious. Well, in any case, we're going to go back to the example we were showing before.

What I'm going to do is I'm going to say, OK, when you have 10 versus 11, the 11 team ends up with four units left. And two of them are very low hit points, but two of them have a lot of the points remaining. So we can start to define two quantities, the robot kill ratio and the HP kill ratio.

So the robot kill ratio is to say I put in 11 units, and I got out four of mine remaining. So 11 is the bottom. I got four remaining, so that cost me seven. But in the process of costing me seven, I was able to take out 10 enemy units. So that's 10 divided by 7, and if we make it numerical with a decimal point there, we see that number. Question.

Yeah, how [INAUDIBLE]?

So I'm actually modeling the combat by distributing the attack evenly from one side and the other. And then any remaining units will have the opportunity to deal more damage to the enemy team. So I don't know exactly how accurate it is. And if you can manage to maintain a tight choke point, then you can trade evenly with your enemy. But that's the best you can do if you're outnumbered.

If you have the advantage and you surround the enemy, then you could do even better than this. This is something like a middle ground between those two cases. Good question. If you want to see the actual way that it works, learn Mathematica and read the code that's right here.

So we see here that there's the robot kill ratio. That's pretty good. We came to the fight with 10% additional units, and we end up getting 40% more value out of our units that we lost. On a hit point basis, we didn't get that much of an advantage. Bringing a 10% advantage got us 6% hit point advantage on how many hit points we were able to lose versus the enemy.

So let's look at a lot of results. So here on the left, I made a table. So I've just done the simulation for 20 units versus i units, where i goes from 21 to 30. So you can see the force ratio that I'm bringing to the table. Here I'm imagining that I have more than the enemy. But of course, you can just invert these numbers and get the reverse case.

The force ratio I'm bringing is 5% extra, 10% extra, and so on, and that achieves these unit kill ratios. You can see here I've capped it out at 20,000 so that it doesn't display the infinity sign. Because here, once you have more than 1/3 more than the enemy, you don't lose any units at all during combat. That's according to the simulation that I've set up. A pretty amazing result if you ask me.

So you can see here that the unit kill ratio magnifies quite intensely. The hit point kill ratio doesn't magnify that intensely. So here, 5% force ratio gets you about 20% unit kill. 10% gets you 40%. 15% gets you 80% extra. And then once you get up higher, it just ramps up more from there.

So it shows that even it's a fight of five versus four, it's probably a very bad idea to engage the enemy. Because you're not going to be able to kill any of their units before they kill all of yours, which means that all of theirs can sort of head back to a medbay, get healed up, and then be back to the fight.

It's worth mentioning here that the medbay can end up being as productive as your headquarters. So let's go back up to where I've talked about the medbay. Let's talk about it here. So let's talk about the production abilities of your various capabilities.

You've got your headquarters, which is making units at 40 hit points each. And let's say that it produces them every 10 rounds, or maybe every 5 rounds at best if you've got a lot of suppliers.

So this is your headquarters building units. So it's building hit points every turn. So it's building from four to eight hit points per turn. Well, that's not too bad. That's building four to eight hit points per turn.

But what about everything else? A medbay can heal two hit points for every of the adjacent units. So that there, right there-- if you can multiply correctly-- it's 16 hit points per turn. That's so many. That's the value of two headquarters right there, even considering headquarters at full supplier capacity.

That just shows that even if you're using medbays incorrectly or doing it wrong in the various ways that you can, you'll still be doing a much better job than a headquarters alone. I mean, that's ignoring the fact that there are other things at play here. Like you have to have injured units in order to produce hit points in a medbay. But it's certainly worth mentioning that even this basic analysis shows what it is. Very clear.

Now, what's the killing ability of artillery? It's been said that they're overpowered. Or that they're quite strong, and they're definitely worth using. But is the killing power of an artillery worth it compared to the healing power of a medbay?

Purely from a numerical standpoint, you can see here that the best the artillery can do is to deal 40 damage to its target and 20 damage to each of eight targets adjacent for a total of 200, if I did that math correctly. And it does that every 20 turns, which is its attack delay. So that gives us 10 hit points per turn. Look at that. Oftentimes, 10 less than 16. True. It's even going to tell me that.

[LAUGHTER]

Well, what a big surprise. So you can make the decision for yourself. And of course, each one has its own uses, because the artillery is not always going to be able to deal 10 hit points. Medbay's not always going to be able to heal units. Maybe it's too far from the headquarters to be a real defensive asset. Maybe not. There are a lot of considerations to include.

But doing a calculation like this can certainly help you out in making your decisions. For example, I mean, I think it's worth doing the following check. Check if there's an encampment near your base. If there is, go ahead and make it a medbay.

If it's not that near, maybe it's better to make it artillery. Like if it's adjacent, you probably want a medbay. But if it's not close, the medbay becomes less and less valuable. Whereas if it's in still artillery range, maybe it's still worth having.

It's the kind of question-- this is the kind of trade-off you could make, and you could just hard-code it in. If it's more than a distance 16, distance squared 16, then go ahead, do it this way. Otherwise, do it that way. Very simple way of addressing the problem.

So let's go back down and let's talk more about encounters of units. So let's continue with this unit kill ratio results table. So we're here. And you know, it's kind of hard to look at numbers. Oftentimes when people have a lot of numbers, especially in a table, they plot it. So here I've got a plot. Doesn't mean that much. Let's do a list plot so we've got points.

And so we're asking the question-- here is the input. Is that the input? No, the input, it goes up to 1.3. Yeah, the input is here. So this is having 5% advantage, 10% advantage, and so on. And the question is, how many hit points is that worth for you? What hit point advantage does that get you?

And so we could do a best fit for these results. So in Mathematica that would look like-- find a fit. You put the data in. You put the expression ax plus b, where a and b are the parameters, and x is the variable in question. And this is your fit rule, and then you can just apply that to the expression.

And in one line-- we'll have a plot. So we'll plot this thing, mess that it is. Well, OK, why don't we just call it here. The line is this, so we're finding the best fit line here. And x goes from-- I don't know-- 0 to 1.5, and that ought to do it for us.

We'll plot that line, and we'll plot these points. So we'll call it points equals that, and we'll show the line together with the points. And we'll also be able show other things. PlotStyle Red. We'll make the line red because that's real style. That's real style.

That almost worked. There you go. So there's a line. What line is that? It's the line where-- the line. Please tell me the line. It's 0.69. So there you go. The slope here is 0.69. I mean, of course, this is probably not valid down at these low numbers.

But the point is that for every additional 1% or one unit that you add, that's going to be an extra 0.7 units that you're going to damage of your enemy. And it's sort of fairly linear between 1 and 5.

Let's see if we really dominate the enemy how the results will sort of stack up. So here it's simulating a whole bunch of combat, and it's going really slowly. Why is it going slowly? Well, because I put in like that delay thing so that you could see what was going on. So I'm going to change the delay to 0 and scroll right on back down, giving everybody in the room nausea.

There's also a bunch of print statements. So we're seeing like 200 print statements, which was to print the dynamic results of the match so that you could see them fighting, sort of seeing numbers decrease on a board, which is a lot like combat in the eyes of a nerd.

So then we can see here a bunch of force ratios. And here in this example, we see that we're maxing out at a certain number. So according to the way that I've set up this simulation, beyond a force ratio of 2, you're not getting any additional advantage out of your additional troops because they just can't find a place to stand.

In reality, when you have four versus one or eight versus one, there's a place for all of your eight guys to stand around all of their one poor little guy. But if you have a large number of units, it may be less realistic to assume that there's that much room available for them to double team, or quadruple team, or eight-tuple team the enemy.

So you could see here that these are some results. And now if we plot them again, we see that we sort of hit the ceiling right over here, and so the line sort of is totally destroyed and comes out wrong. Let's go up to 3 so we see how bad-looking it is.

And this is just yet another example of how doing a best fit line to something is not the right answer 100% of the time. It's the right answer 99.99% of the time. Or basically use the r squared number to see if it's-- no, I'm just kidding. That's not genuine statistics. And most of you are probably poo-pooing the idea that I even considered for a moment or a nanosecond including the two words in the same sentence.

So that's a sort of summary of these issues. Like what force ratio gets you what unit or hit point kill ratio? You can see here a big takeaway lesson that just 25% more units or just 30% more units than the enemy can make a giant difference. And you should never engage when you're outnumbered, unless you're right next to the headquarters and there's no choice.

I strongly recommend that you consider the little micro-tricks, trying to get your units into the right place, withdrawing when it's a good idea. Remember that if you're retreating from the enemy and the enemy is adjacent to you, that means you can take damage every turn while you're retreating.

So in this game, retreating is definitely not encouraged for that reason. What you need is like allied units to sort of distract the enemy while your injured ones are going to heal. Or you need to have your medbays close enough that you can survive the retreat.

Because the way the game works is, you move, and then you deal attack damage. So if you retreat from your enemy, you're not adjacent to him anymore after you've moved. And then he moves toward you and deals damage. And then you retreat, and you don't deal damage. And he moves toward you and deals damage. And it's just a very bad situation. I've been in it myself. It was very painful to watch.

So with that, I mean, I think that's going to be all that I've got for you today, even though it's slightly early on the days I've gone over. And so I recommend that you just sort of sit here, enjoy yourselves, ask me questions. And when the Bubble tea arrives, the deputies will spring into action.

So thank you very much for coming to Lecture 6. I guess I should explain what's going on in the next lectures. Next lecture is going to be an exciting guest lecture, where we've got one of the developers of Battlecode who's been remote and who was the winner of Battlecode last year, fun gamers, on that team.

He's coming here to talk about various important things. I forgot what he's-- oh, my goodness. What a terrible plug for his talk. It was going to be about-- no, that wasn't a helpful comment. That was just talking. Yeah, he's going to talk about, I think, emergent complexity and like really awesome tricks that you learn if you do Battlecode N years, where N is a large number.