Evaluating Player Decisions in Ethical Simulations

Trevor Hubbard

College of Information and Computer Sciences

tmhubbard@umass.edu

January 21st, 2020

Abstract

As society's reliance on computing technology rapidly expands, it's become increasingly apparent that Computer Science students ought to receive some sort of ethical training to supplement their technical training. Our team has explored an alternative to a traditional lecture-based course: an interactive simulation that tasks players with navigating a CS-centric workplace ethical dilemma. We hypothesized that this interactive approach to ethics education would increase student engagement, and that the simulation's nonlinear, dynamic narrative would prompt further discussion amongst students about their experiences. In order to provide feedback to players about their experience, it was necessary to develop mechanisms under which player decisions could be evaluated. This paper focuses on the design and implementation of those mechanisms; which involve the creation of a stakeholder theory-inspired model, and a subsequent quantification of the player's exposure to said model. This model is relatively simple to design, parameterize, and evaluate, and ought to serve as a solid foundation for future simulation development.

Introduction

Since their inception in the early 20th century, computers have become irreversibly embedded in practically every walk of life. Their proliferation has only continued to accelerate within the last 20 years, as the advent of the smartphone has ushered in an age of perpetual connectedness. Undoubtedly, computing technology has enabled huge leaps and bounds for the progress of humanity as a whole; like any other technology before it, though, it's also ushered in a number of issues. Constant connectivity has allowed for immeasurable improvements in human communication and collaboration, but it's *also* created absolutely unprecedented levels of data collection (which, in turn, has resulted in a slow-but-steady erosion of privacy). Artificial intelligence can be used for a myriad of positive things, from the optimization of energy grids to the discovery of new, lifesaving drugs. On the other side of the proverbial coin, the same tech is used to make potentially biased, unfair decisions about someone's eligibility for a loan, or about their likelihood to recommit a crime.

The potential for unethical use of technology is stronger than it's ever been; in lieu of that, it's essential that the developers of these technologies have received robust training on ethical thinking. By emphasizing considerations of safety and fairness throughout *any* development process, professors could hope to impart a sense of social responsibility on their students before sending them out into the industry. At UMass Amherst, this emphasis has already begun to rear its head: courses like COMPSCI 305 and COMPSCI 320H spend some time exposing students to the repercussions of a lack of ethical thinking. These courses are definitely a step in the right direction, but there's still work to be done: UMass has no mandatory course dedicated to exploring some of the ethical quandaries that almost *every* developer will have to face at some point in their career.

Throughout the course of the last year, I've been a part of a research team dedicated to addressing UMass's lack of an ethics course. Our team (consisting of Professor Lee Osterweil, Professor Peter Haas, Research Associate Heather Conboy, and undergraduate students Kyle Tan and Jonathan Trott) has developed an alternative to a traditional college ethics course: an interactive, ethics-focused simulation. In the simulation, players assume the role of a developer at a popular tech company, and are dropped into the middle of some ethically questionable scenario. To progress through the narrative, the player can choose to converse with a number of different customers, colleagues, and business partners; since there are a limited amount of conversations allowed to happen, though, the player is encouraged to think about the best way to gather as much information about the ethical quandary. Then, at the end of the simulation, the player can decide whether or not they want to bring their concerns to a supervisor. Depending on what information has been gathered, the supervisor could react in different ways.

In constraining the number of conversations, we're hoping to accentuate the importance of an immensely popular ethical framework: stakeholder theory. To make ethical decisions in the workplace, a developer must constantly be thinking of each and every group that their work will eventually impact - that way, they'll be more apt to avoid decisions that'll negatively impact these groups. When designing the simulation, we wanted to use the narrative as a tool to demonstrate the impact of the player's choices; ultimately, though, in order to do that, we needed to figure out *which* possible sets of conversations would lead to certain outcomes. We also wanted to avoid making it seem like there was one "correct" way to run through the simulation - doing so would strip the simulation of the complexity of these sorts of situations (which, in reality, are far from black and white). The question of how to fairly, realistically evaluate a player's decisions was one that our team struggled with during a large portion of the development cycle.

Generalizability was also a huge concern of the team's, both in relation to the decision evaluation mechanisms, and to the simulation's framework as a whole. When the idea of a simulation was initially proposed, we knew that a short simulation couldn't totally replace an entire course - so, instead, we aimed to create *multiple* simulations, each suited to address ethical quandaries specific to certain classes. Heather Conboy (the team member that wrote the code for the simulation) developed the simulation framework with generality in mind, and each of the three undergraduate students working on the project (myself, Kyle, and Jonathan) all developed our own scenarios for the simulation. (This scenario development took up a large portion of the Spring 2019 semester!) While I developed the evaluation mechanics, I tried to embrace this spirit of generalizability: instead of writing a multitude of different responses to each possible set of conversations a player could have, I instead tried to design a system that could be easily ported to other simulations with ease.

The resulting system that I've created manages to address both concerns of fairness and generalizability! What's more: the framework required to implement it is relatively simple, *and* can double as a framework for the scenario development process as a whole. Because of that, we hope that more professors will be encouraged to develop scenarios suited to their own courses, and further reinforce the importance of ethics education at UMass!

Previous Research

Throughout the process of designing the evaluation methodology, I carried out research in a variety of different fields. Generally, the publications I sought out fell into one of three categories: ethical analyses of computer science as a whole, studies on games as a method of communicating values, and technical documentation on systems that could be leveraged for the simulation framework. In developing my own evaluation tools, I attempted to synthesize some of the ideas communicated in the publications I'd found; before discussing *my* work, then, it's only natural to examine a selection of the sources that served as inspiration:

Computer Science Ethics

The following sources examine ethics in the context of the field of computer science. They were instrumental in helping to choose which perspectives to highlight within the simulation's narrative, and serve as further justification of the need for more ethical training.

ACM Code of Ethics and Professional Conduct

As we developed our simulation, we wanted to be careful about the way we portrayed some of the ethical issues within; this Code of Ethics served as a basis for understanding what an "ethical" way to approach a dilemma might look like! The Code's preamble mentions that computing professionals ought to, "reflect upon the wider impacts of their work, consistently supporting the public good" - we tried to embrace this ethos throughout scenario development by putting positive emphasis on perspectives that placed public good over personal gain.

The Code contains a list of best practices for ethical decision-making, all centered around serving the public good. A number of these practices served to inspire the ways that we examine our simulation's quandaries. My scenario, for instance, highlights tension between two conflicting forces: the competitive edge gained from a timely product launch, and the importance of strong cybersecurity testing for a medical device. Sections 2.5 and 2.9 of the Code encourage "comprehensive and thorough… analysis of possible risks [of computing systems]" and the design of systems "that are robustly and usably secure" - naturally, then, I chose to reinforce the relevance of cybersecurity within the simulation's narrative.

Weapons of Math Destruction

This book is a masterclass in what happens when developers *aren't* thinking of the public good. It consists of a number of case studies that each focus on the unintended negative repercussions of using computing systems to automate historically human processes. In almost all of these cases, the original intentions motivating the creation of these systems were fairly noble: predictive policing software, for example, could work to improve the efficiency of law enforcement and better enable officers to "serve and protect". In practice, though, these technologies implicitly introduced biases that'd been hidden away in the data fueling them. (Predictive policing *actually* served to create a biased feedback loop of predictions that resulted in the over-policing of minority communities.)

These cases studies were an excellent basis for understanding the complexity of certain ethical quandaries, and how easily they can emerge from a myopic view of a project's impact. Inspired by these stories, we aimed for our simulation to increase the player's sensitivity to the well-being of all stakeholders, with the hope that they'll avoid causing these same problems later in life.

Teaching Ethics Through Games

The following sources discuss the utility of using games in the context of values-based education, and which strategies can be employed to increase their effectiveness. These were incredibly helpful in understanding the best ways to effectively communicate ideas with players.

Persuasive Games: The Expressive Power of Videogames

This book is the seminal work on “procedural rhetoric”, a term that author Ian Bogost coins in order to capture the way that games are able to persuade their players of certain concepts. In contrast with traditional rhetoric (which is the process by which language is used to convey a particular point), procedural rhetoric uses the rules of a given procedure (such as the mechanics of a videogame) in order to convince players of something. If the procedure’s rules are well-defined, a player will need to learn them in order to complete the procedure; by offering no alternative to the rules, the player can be persuaded that these rules are absolute truths.

In an attempt to embrace the power of procedural rhetoric, we chose conversation as the main gameplay mechanic of our simulation. By forcing simulation players to talk with various stakeholders when faced with an ethical dilemma, we’re hoping to encourage the same sort of behavior in real-world situations. I kept this concept in mind when designing the evaluation system: in order to reinforce the value of communication, I aimed to commend players that tried to gather information from a diverse range of stakeholders.

The Ethics of Computer Games

This book is a great examination of the existence of ethical frameworks in games! Throughout the book, Miguel Sicart analyzes a wide range of games with a critical lens, breaking down the effectiveness ethical statements the designers are making (either intentionally or unintentionally). According to Sicart, a game can make strong ethical arguments by allowing the player opportunities for reflection. A game can't wear its ethics on its sleeve: by explicitly telling players whether their decisions were "good" or "bad", it can strip the player of the responsibility of figuring that out for themselves. In accordance with that reasoning, I avoided integrating real-time feedback with my evaluation of the player's decisions, instead focusing on using my evaluation as a basis for encouraging future reflection!

Technical Documentation

The following sources helped to further develop some of the more technical aspects underlying the simulation framework. They helped in understanding how to better implement some of the ethical ideas discussed in other sources.

Designing Games for Ethics: Models, Techniques, and Frameworks

This book contains a number of articles investigating different facets of how games approach discussing ethics; in this way, it's similar to Sicart's book. One chapter in particular, though (Chapter 17: "The Doctor Will Be You Now"), spends a decent amount of time talking about the creation of a medical ethics simulation, and the underlying mathematical model that works to evaluate the players' decisions.

The approach taken by the researchers developing their ethical simulation is somewhat reminiscent of my own: each player decision is weighed by how much it affects a certain group of people (i.e., the government, the consumers, colleagues, and the player's character), and then these weights are used to then determine which consequences ought to be shown to the player throughout the game. Since I had plans of employing a similar weighting system, it was encouraging to see a successful implementation of it elsewhere.

Little-JIL 1.5 Language Report

Little-JIL is a visual, agent-coordination language that was used as the basis of the simulation's framework! Heather, one of the members of our research team, built the framework over the course of Summer 2019; this manual has served as a great resource for understanding how its parts connect with eachother as I've edited it in order to include my evaluation mechanics!

Current Methodology and Goals

Scenario Conception

When this project first began in the Spring 2019 semester, there was no clear consensus on what piece of the simulation ought to be developed first. (We weren't even sure about which framework we'd use for development until the end of the semester!) Eventually, we came to the conclusion that the actual simulation *scenarios* could be built independently of a development framework - so, after various storyboarding lessons from Professor Michelle Trim, Kyle, Jon, and I each began to write our own scenario.

While brainstorming ideas for my scenario's ethical quandary, I became fascinated with the medical devices industry. Medical devices, like countless other everyday tools, are being exposed to the "Internet of Things" paradigm. In 2015, for instance, Medtronic released a smartphone app that was able to connect to a patient's pacemaker, enabling a remote transmission of medical data to doctors - this was the first application of its kind, and it was widely lauded for saving patients from countless hours at the clinic. Since 2015, many more "smarter" medical devices have begun development, all of which promise to solve a myriad of issues for their users. Unfortunately, though, making these devices smarter presented its own new set of issues: what if someone's pacemaker was hacked?

Although there haven't yet been any recorded instances of a medical device being hacked, various other Internet of Things devices have been criticized for their shoddy cybersecurity measures. 2016 was host to the largest DDoS attacks the Internet had ever seen, and it was largely fueled by a malware called "Mirai". Mirai enlisted countless insecure IoT devices in a malicious botnet, and then used this botnet to coordinate a massive DDoS attack against a popular DNS provider, Dyn. Once the attack had been made known to the cybersecurity community et large, one point became increasingly clear to everyone involved: these smart devices ought to have much stronger cybersecurity measures in place.

Inspired by these events, I decided to focus my scenario on a fictional medical devices company, EvoHealth. In the scenario, players take on the role of an EvoHealth software developer working on one of their latest medical devices: an artificial pancreas. As the scenario develops, it becomes clear to the player that the artificial pancreas has no cybersecurity measures; despite that, the management of EvoHealth is pushing hard for the completion of the device, as they're angling to time its (hopefully successful) release with the IPO of the company.

Under this premise, the player is driven to investigate this lack of security. By the end of the scenario, they're forced to make a decision: do they continue their development on the device in spite of what they've learned, or do they dig in their heels and try to convince the management of the benefits of cybersecurity?

From Scenario to Simulation Framework

Once Kyle, Jon, and I had each created a basic outline for each of our scenarios, we moved back to the drawing board to discuss how each scenario would progress. In accordance with our goal of creating a generalized simulation framework, we wanted to make sure to structure our scenarios in a similar way. Eventually, we came up with the following outline:

A high-level outline of the structure of the scenario

Upon beginning the simulation, a player is delivered some relevant background information regarding their avatar's role - in my case, it's a synopsis of EvoHealth's current project, the artificial pancreas. This background sets the context for the presentation of the quandary, during which we begin to drop hints to the player about the ethical dilemma at play. I was careful when writing this quandary scene, as to avoid a blatant description of the scenario's tensions; instead, I opted for a subtle approach, only lightly referencing the device's lack of security. After the player has been exposed to the quandary, we allow them an opportunity to reflect on what they've learned so far. This time for reflection is delivered as a writing prompt, which was written to encourage students to think critically about their avatar's role.

What new responsibilities do you have after being assigned to this project?

What aren't you sure about, or what questions are raised for you about those responsibilities?

The two initial reflection questions provided to the player

Following their initial reflection, the players begin the "Information Gathering" stage, which is where the majority of the scenario's content lives. During this stage, players have the ability to talk with various friends, colleagues, and business partners. Each person has a different perspective on the quandary, and it's up to the player to try and piece together a larger understanding of the quandary through conversation. *However*: during this stage, the player may only engage in at most five of the nine available conversations. (They could choose to engage in less than five conversations, too!) Before players select their choices, the simulation warns them to "be mindful of who is best positioned to fill in your understanding of the project, and who will help you think critically about the task assignment at hand".

By limiting the amount of conversations the player is allowed to have, we hope to evoke some careful consideration from the players. As was mentioned earlier, we want to implicitly teach students about stakeholder theory. When dealing with an *actual* ethical dilemma, it's incredibly important to understand who'll be impacted by a decision. That way, you won't run the risk of making decisions that only support a small portion of the stakeholders. In real life, though, these sorts of dilemmas can be time sensitive - so, although you might want to try and talk to every possible stakeholder, it's unrealistic to expect that you'd be able to. In this way, we thought that the conversation limit was an adequate way of modelling some of the real-life complexities of dealing with these dilemmas!

Once the player has finished conversing, they're prompted to take a side: do they continue working on the project, or do they request a delay? Immediately following their

decision, they're given another opportunity to reflect on their actions. Finally, after reflection, the player proceeds to the Consequences stage, where the player learns about the various implications of their decision, and is offered some feedback about their conversation choices.

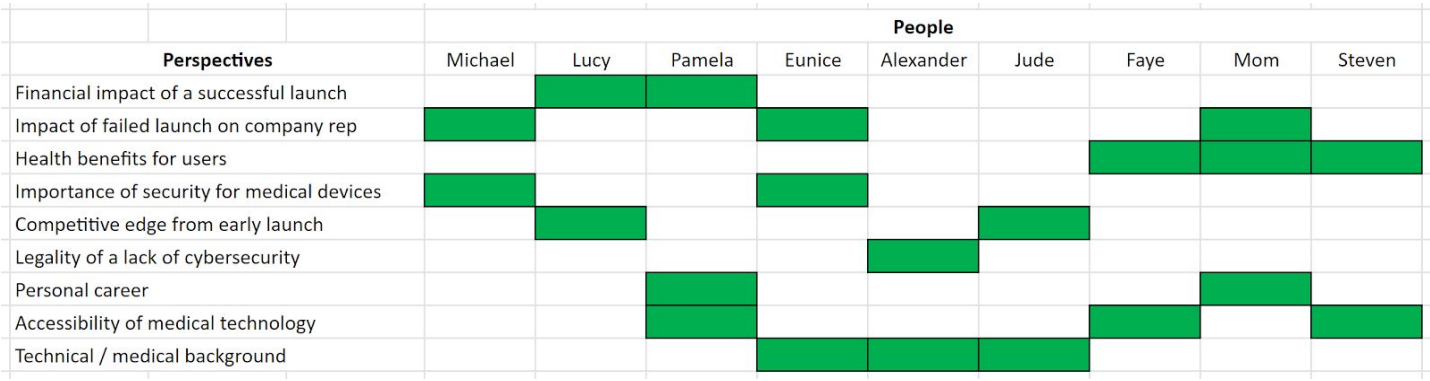
When we created this high-level overview of the structure, we weren't at all sure about how the Consequences stage would be generated. In any choice-based simulation, it's natural for a player to want their choices to have some palpable impact - so, at the very least, we knew that we'd want to dynamically create consequences using the choices the player made. Additionally, we aimed to create some sort of consequences system that could be generalized across different scenarios - that way, professors could create their own scenarios with ease. Finally, we wanted to avoid choosing one particular choice set as the "correct" choice set, thereby encouraging players that there are multiple ways to effectively approach dilemmas like these. With those three goals in mind, I set out to create the simulation's evaluation mechanic!

Research Results

Creating a Perspective Coverage Matrix

During the scenario development stage, Professor Osterweil recommended that we created something that was eventually dubbed the "perspective coverage matrix". The rows of this matrix would list out perspectives covered within the scenario as a whole, whereas the columns would list each of the people that the player could have a conversation with. If a character made some reference to a perspective, then that particular cell would be positive; otherwise, it'd be blank!

This document was designed to help us keep track of which topics each of the scenario conversations covered, and Professor Osterweil hoped that it would help us figure out if there were any topics that were represented more / less than others. As luck would have it, this document served a larger purpose: it was the basis for the evaluation system that I created!



	People								
Perspectives	Michael	Lucy	Pamela	Eunice	Alexander	Jude	Faye	Mom	Steven
Financial impact of a successful launch		■	■						
Impact of failed launch on company rep	■			■				■	
Health benefits for users							■	■	■
Importance of security for medical devices	■			■					
Competitive edge from early launch		■				■			
Legality of a lack of cybersecurity					■				
Personal career			■					■	
Accessibility of medical technology			■				■		■
Technical / medical background				■	■	■			

A screenshot of the perspective coverage matrix I made

The perspective coverage matrix definitely served its original purpose, as I noticed that certain conversations I'd written barely touched upon any of the listed perspectives - so, I edited their conversations a little in order to include references to those perspectives. As I iterated through the editing process, something started to dawn on me: certain conversations I'd written felt much more *important* than others. For instance: the conversation with Eunice, a cybersecurity consultant, felt much more relevant to the dilemma than, say, the conversation with your mom. Initially, I thought that this was an issue - I figured there ought to be equal value in having each of the conversations. After thinking about it more, though, I realized that this dynamic mirrored reality more than its alternative. Talking to a parent about a work issue is absolutely natural, but ultimately, that parent's advice will most likely be fairly surface level - you'd be much better off discussing it with someone more knowledgeable, or someone who was likely to be directly impacted by the technology.

Inspired by this notion of “importance”, I returned to the perspective coverage matrix and started to define which conversations were the most important to the scenario. At that moment, the ambiguity surrounding the evaluation mechanic started to clear up: what if players were evaluated by how many “important” conversations they chose?

Redefining Importance

After I’d spent a little time identifying the “important” people in my perspective matrix, I began to realize a couple things. First of all: certain perspectives (like “importance of cybersecurity for medical devices” and “impact of a failed launch on company reputation”) were commonalities between some of the “higher importance” people I’d picked; other perspectives (such as “personal career” and “competitive edge from early launch”) appeared more often with some of the conversations that I’d ranked as less important. Implicitly, in ranking the importance of each conversation, I was identifying certain perspectives as more important, too!

Subsequently, I realized that basing the evaluation mechanic on whether or not the player engaged in certain conversations was a flawed system. One of our initial goals was to avoid creating a “correct” way to play the simulation, as to allow different players to have different (but equally fulfilling) experiences. Under an importance-ranking-based evaluation mechanic, any set of conversation choices would need to be compared to this set of “correct” conversations, which would be counterintuitive to this original goal.

These two realizations led me to a third: instead of ranking the people by their importance, I could rank the different *perspectives* by their importance. That way, there’d be some flexibility introduced into evaluating the player’s choices!

Creating Evaluation Metrics

This realization about perspectives' importance quickly led to the development of a new metric: "stakeholder value." The metric uses the notion of perspective importance to define how "valuable" the conversants' (which we'll furthermore refer to as "stakeholders") conversations are. Initially, I defined a perspective's importance in a binary way: either a perspective is important (an importance of 1) or not (an importance of 0). With that in mind, the equation for stakeholder value is as follows:

$$Stakeholder\ value = \frac{\sum^{Stakeholder's\ perspectives} importance}{Number\ of\ covered\ perspectives}$$

In other words, the stakeholder value is the average importance of all of the perspectives that they cover! Next, I created the "player score" metric - this is the metric that could be used in order to evaluate the player's choice of stakeholders. As follows is the equation for player score:

$$Player\ score = \sum^{Stakeholders\ conversed} Stakeholder\ value$$

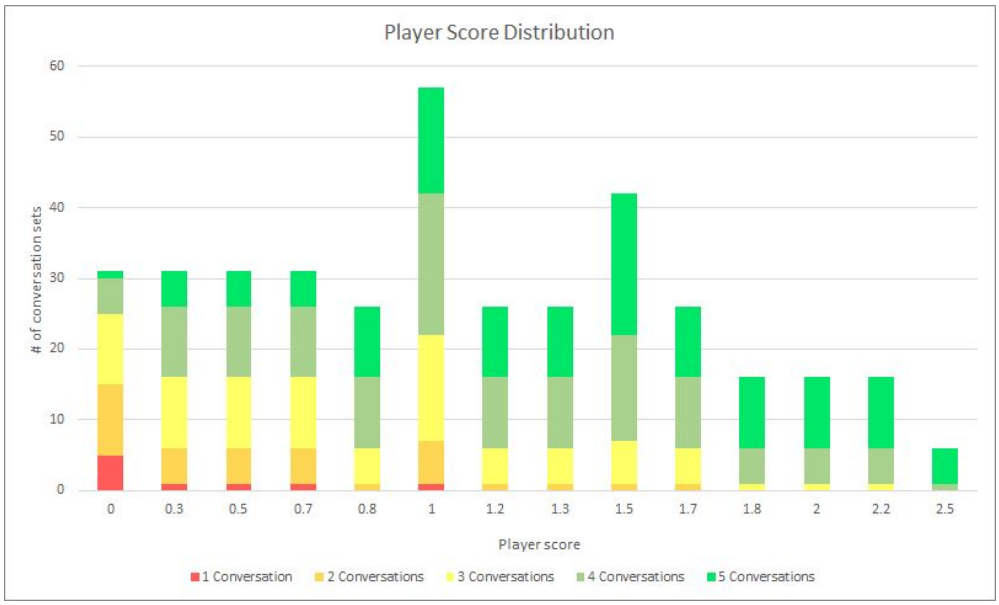
As an initial test for my rudimentary scoring system, I assigned importance to perspectives that could negatively impact the well-being of stakeholders if ignored - these were perspectives like "importance of cybersecurity for medical devices" and "legality of a lack of cybersecurity". I *didn't* assign importance to perspectives that were purely beneficial to singular parties - for instance, "personal career" and "competitive edge from an early launch" weren't

marked as important, as ignoring them wouldn't have any direct negative impact on the stakeholders representing them. As follows is a list of how I calibrated the perspective values:

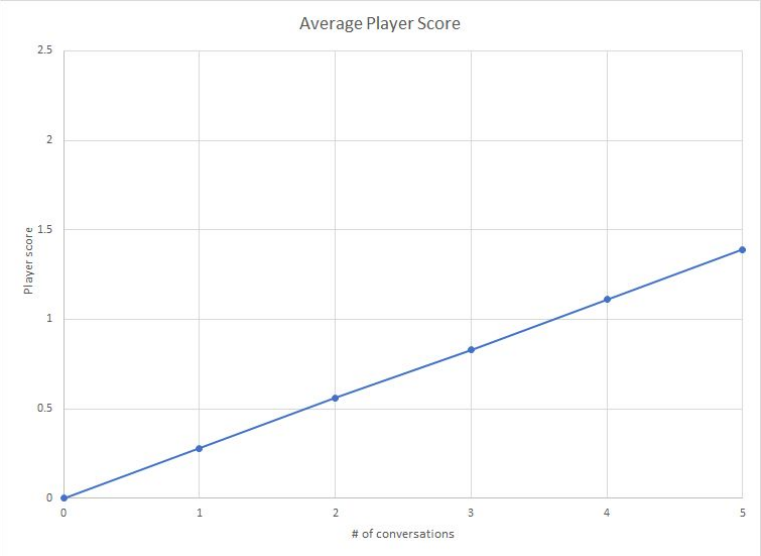
Perspectives	Importance
Financial impact of a successful launch	0
Impact of failed launch on company rep	1
Health benefits for users	0
Importance of security for medical devices	1
Competitive edge from early launch	0
Legality of a lack of cybersecurity	1
Personal career	0
Accessibility of medical technology	0
Technical / medical background	0

Only three perspectives were marked important under this schema

Once the importance parameters had been set, I wrote some Python code to simulate the outcomes of different possible runs of the simulation. I generated all of the possible sets of choices for stakeholder conversations, keeping in mind the possibility that the player could engage in less than 5 conversations. As follows is the score distribution for those parameters:



After seeing the distribution, I was pleased that my player score metric seemed to create a wide range of possible scores. The actual distribution of those scores, though, wasn't as pleasing - I had been hoping for something closer to a Normal distribution. (That way, it would mimic the distribution of, say, students' grades on a test.) Additionally, the correlation between the number of conversations had and the player score, while behaving in the positive manner I'd expect it to, isn't as pronounced as I'd want it to be:



Ideally, the average player score of 5-conversation choice sets ought to be closer to the maximum player score (which is 2.5, in this case)

So, in order to add a little more variability to the score distribution, I started to tweak each perspective's importance value! Instead of treating importance as a binary variable, I decided to add a little variability. That way, I could introduce the ability for a perspective to still be important, but not *as* important as other perspectives. For the sake of simplicity, I kept the importance values between 0 and 1. Then, I assigned importance according to the impact that each perspective could have on larger groups of people.

Perspectives	Importance
Financial impact of a successful launch	0.5
Impact of failed launch on company rep	0.75
Health benefits for users	0.5
Importance of security for medical devices	1
Competitive edge from early launch	0.25
Legality of a lack of cybersecurity	0.75
Personal career	0.25
Accessibility of medical technology	0.5
Technical / medical background	0.25

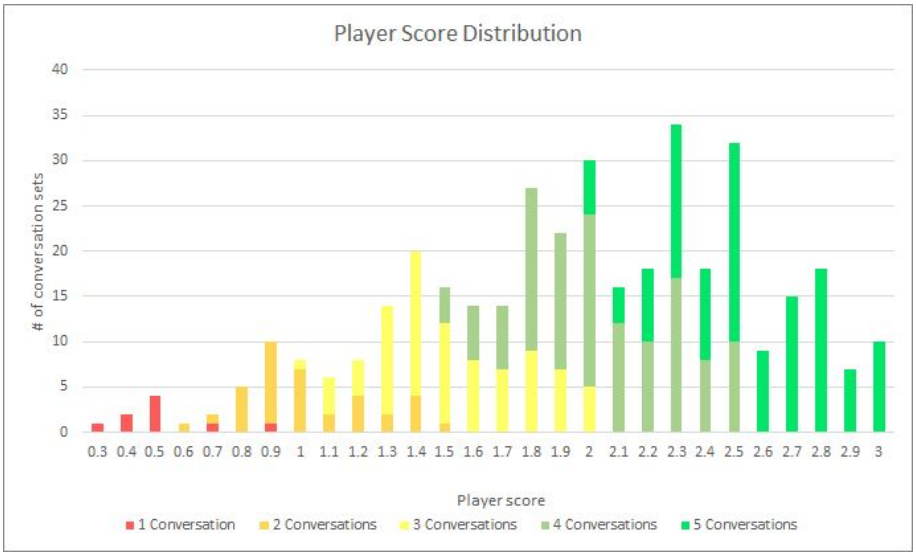
The weights of the perspectives' importance have been changed

Cybersecurity was the only perspective to be assigned a 1, as someone's *health* could be negatively impacted if EvoHealth's device *were* to be hacked. Below that, at 0.75 importance, are "impact of failed launch on rep" and "legality of a lack of cybersecurity" - both of these could have a negative impact on the employees of EvoHealth, as they could have layoffs or salary cuts as a result of fallout from a hack. Next, at 0.5 importance, we have "financial impact of a successful launch", "health benefits for users", and "accessibility of medical technology" - these perspectives all concern positive impacts on larger groups of people (like EvoHealth employees, or the patients using the artificial pancreas), but are less important than some of the other perspectives that concern *negative* impacts on stakeholders.

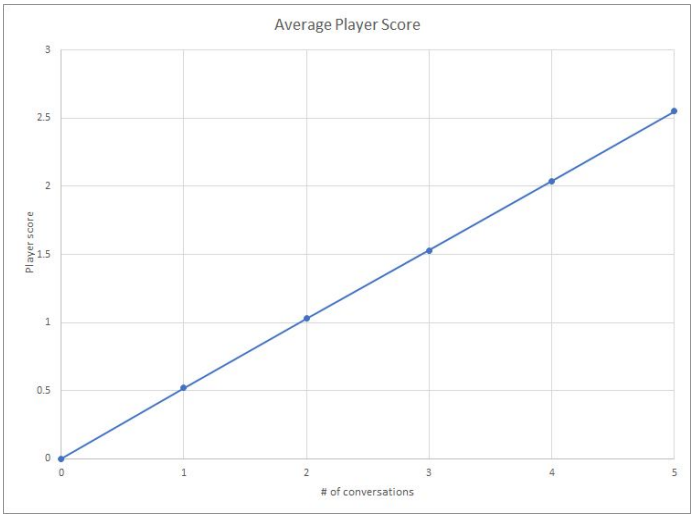
Lastly, at an importance of 0.25, we have "competitive edge from an early launch", "technical / medical background", and "personal career". While all of these are important in their own right, they're not quite as relevant to the player's eventual decision about asking for a delay. An early launch doesn't *guarantee* EvoHealth a competitive edge over other medical devices companies, so this shouldn't hold too much weight. The "technical / medical background" perspective covers some contextual information that explains things like Bluetooth

connection and diabetes to the player; while it's important that the player understand the intricacies of what their avatar is working on, this information shouldn't have a huge impact on the player's decisions. Finally, the player's personal career: although it's an incredibly important factor in someone's life, people shouldn't put their personal gain over the well-being of others.

With these new weights, the score distribution changed to look like this:



It was immediately apparent that the resulting score distribution looked *much* more Normal than the original one! Furthermore, the positive correlation between number of conversations had and player score was much more pronounced:



Despite the clean-looking distribution, I still wasn't fully satisfied with the player score metric for one main reason: if there were two stakeholders that covered the same perspectives in slightly different ways, they would still be treated as equal. In my scenario, for example, Michael and Eunice covered the same two perspectives: "impact of a failed launch on company reputation", and "importance of security for medical devices." Under the current scoring system, they'd have an equal stakeholder value, and therefore would contribute to the player's score in the same way. In reality, though, Eunice ought to be the higher-valued stakeholder, as she focuses on the (more important) cybersecurity perspective in more detail than Michael does.

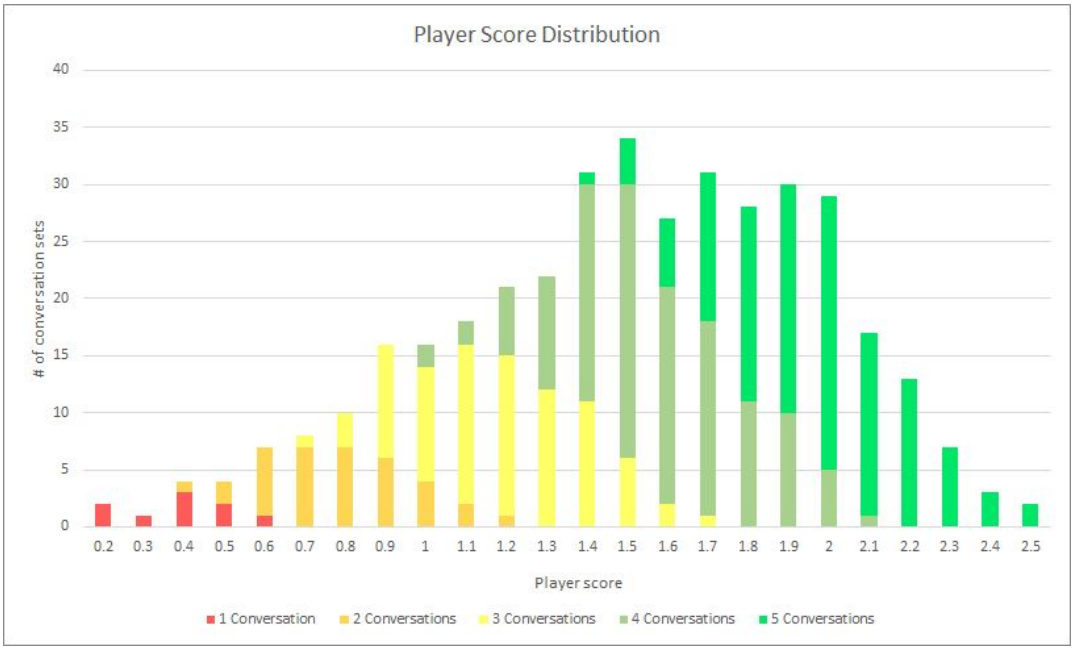
In order to address this problem, I introduced *another* metric: "coverage". Coverage is a 0 to 1 decimal value that's meant to indicate how deeply a perspective is covered. A value of 0 means "this perspective was not covered at all by this stakeholder"; similarly, a 0.5 would mean "this perspective was *somewhat* covered by this stakeholder", and a 1 would mean "this perspective was heavily covered by this stakeholder". Coverage is used to weigh the stakeholder value metric like so:

$$Stakeholder\ value = \frac{\sum^{Stakeholder's\ perspectives} importance * coverage}{Number\ of\ covered\ perspectives}$$

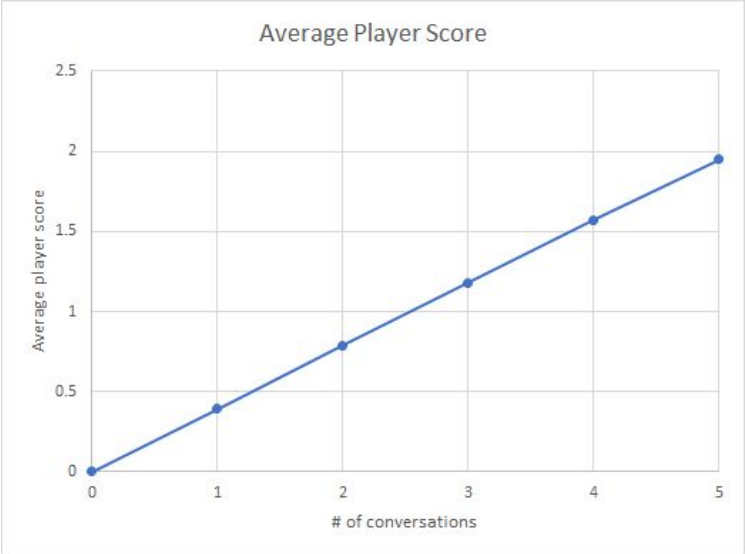
Having defined a new metric, I revisited my coverage matrix and assigned coverage values for each of the perspectives. I kept it simple, using only coverages of 0, 0.5, and 1:

	People								
Perspectives	Michael	Lucy	Pamela	Eunice	Alexander	Jude	Faye	Mom	Steven
Financial impact of a successful launch		0.5	1						
Impact of failed launch on company rep	1			0.5				0.5	
Health benefits for users							1	0.5	0.5
Importance of security for medical devices	0.5			1					
Competitive edge from early launch		0.5				0.5			
Legality of a lack of cybersecurity					1				
Personal career			0.5					1	
Accessibility of medical technology			1				1		1
Technical / medical background				1	0.5	1			

After updating the matrix, I decided to collect more statistics on the distribution of the possible player scores - here's the result:



While the possible range of player scores decreased, the Normal-ness of the distribution seemed to have increased a large amount. Furthermore, the same positive correlation between the number of conversations had and the player score is preserved:



After seeing that distribution, I was fairly happy with the player score metric! It was easily portable from my scenario to other ones, and relatively simple to parameterize. Furthermore, it allows for a multitude of different conversation choice sets to be considered “the best”, which helps to allow for a little more variation in different players’ experiences! Next, I needed to think about how to integrate the player score metric into the Consequences stage.

Implementing Player Score into the Consequences

When we’d initially designed the scenario framework, we intended that the Consequence stage have a two-fold purpose: concluding the narrative presented in the scenario, and evaluating the player’s choices of stakeholders. Fulfilling the first of these purposes (the narrative conclusion) was much simpler than the second: I wrote two conclusions to the scenario, and configured the simulation to choose one depending on whether the player chose to ask for a delay or not. In one conclusion, the player’s request to delay is granted by the project manager; despite missing their opportunity to undercut their opposition with an earlier launch, EvoHealth eventually gets a lot of positive recognition in the press for their commitment to the safety of their customers! In the other conclusion, the player refuses to ask for a delay; this leads to a future where the device is eventually hacked, which causes the public’s perception of EvoHealth to plummet.

Fulfilling the second purpose of the Consequences stage would turn out to be a larger challenge. Despite having just created a “player score” metric, we had no intentions of showing the score to the players - various research (like Michael Sicart’s “Ethics of Computer Games”) had suggested that explicitly showing players scores and statistics about the ethics of their choices worked to counteract a necessary layer of personal reflection. So, in light of that, we

wanted to design the Consequences stage so as to prompt *more* reflection from the player, while still maintaining some subtly in regards to the delivery of the player evaluation. In response to this problem, the team came up with a two-part solution, both parts of which rely on the player score mechanic and the perspectives to some degree. We'll call the first half of our solution the "Perspective Evaluation" section, and we'll call the second half the "Coverage Visualization" section.

The Perspective Evaluation section begins with a clear explanation of the ethical dilemma that the player has been working through during the simulation:

> In this scenario, your company, EvoHealth, is nearing an important deadline for their artificial pancreas device. Timely completion of the device would be hugely beneficial for EvoHealth and its employees, as it'd likely guarantee a successful product launch and a lucrative initial public offering for the company. Additionally, diabetics would have access to the most inexpensive device of its kind! Despite the benefits, there are murmurings of concerns about the security of the device - EvoHealth hasn't devoted resources to any sort of cyber-security testing. For a medical device like the APDS, a hack could be fatal for a user. So, throughout the whole scenario, there's a tension between the numerous benefits of a faster launch, and the potential dangers of ignoring cyber-security concerns.

This is the most direct reference to the quandary we've included in the simulation! Previously throughout the scenario, we'd made a point to avoid directly relaying the quandary to the player; in order to provide a fair evaluation, though, it was necessary to acknowledge what the player was being evaluated *on.* Once the player has been primed with the description, we provide them a short, 1-2 sentence evaluation of their performance. The evaluation sentence is chosen in accordance with the player's score. This sentence is the most direct form of evaluation we've included in the Consequences stage - it primarily serves as a transition into the main content of the Perspective Evaluation section, but will also identify if a player has done a particularly poor / exemplary job picking stakeholders.

As follows is a list of the different evaluation sentences that can be chosen:

Excellent

You've done a great job at understanding the quandary and choosing to talk to stakeholders whose perspectives mattered the most. Here are a couple of perspectives that were covered by conversations you didn't have:

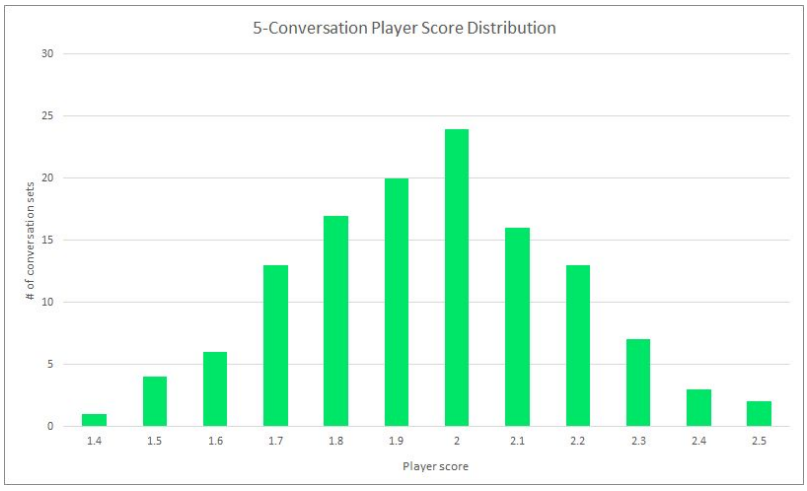
Sufficient

You seemed to grasp the quandary fairly well! Still, though, there were a couple of perspectives you could have tried to focus on more. Here are a couple of them:

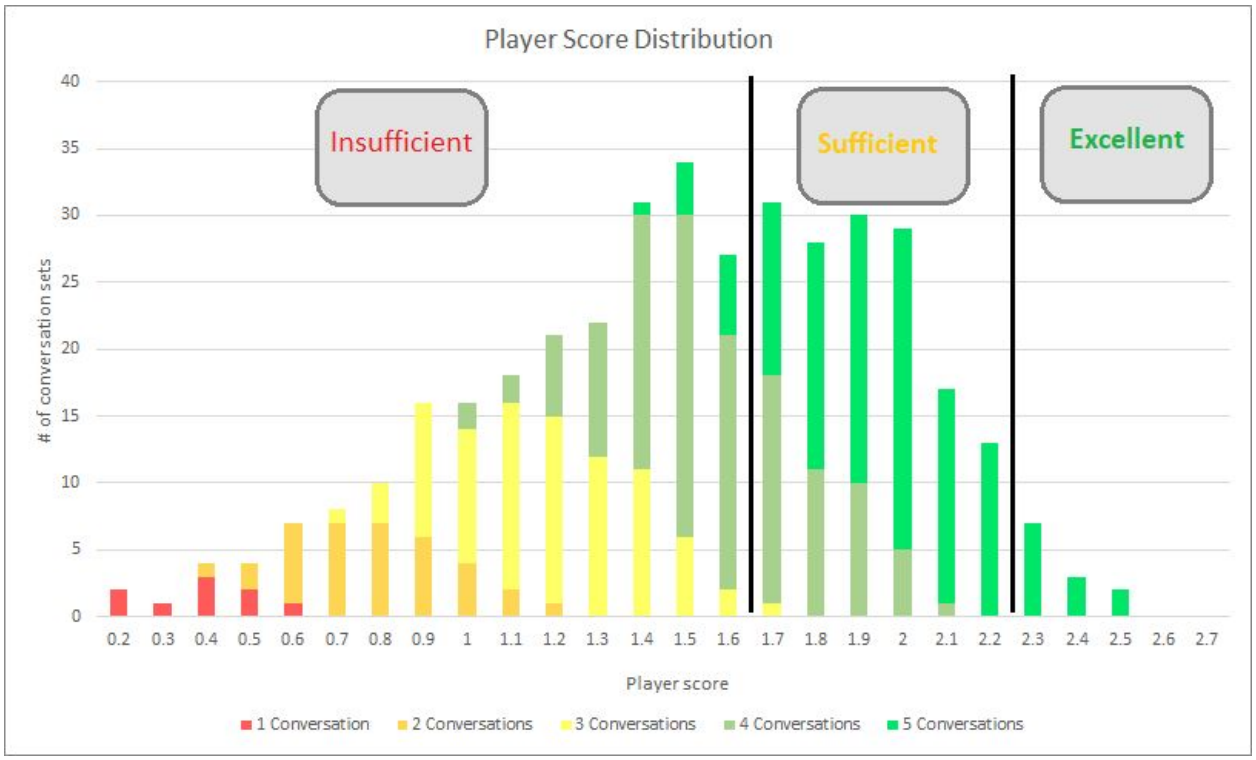
Insufficient

The perspectives covered by the set of people you chose to talk to were insufficient for understanding the quandary as a whole. Here are some of the perspectives that you could have covered more:

In order to determine whether a player's score is excellent, sufficient, or insufficient, it was necessary to define player score thresholds for each score. So, in order to do that, I turned to the distribution of possible player scores. Since one of the main lessons we're trying to convey with this simulation is "talk to important stakeholders before making important decisions," I wanted to give "insufficient" to players who didn't take full advantage of the opportunity to converse with 5 stakeholders; with that being said, I only included the possible scores of players who spoke with 5 stakeholders. Here's that distribution:

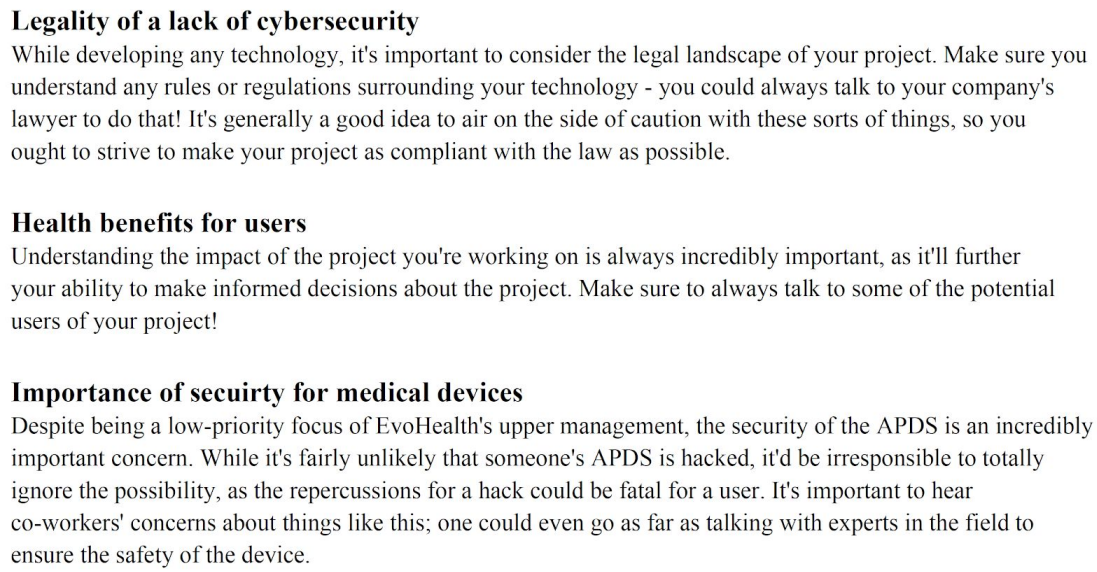


The Normal nature of this score distribution made it fairly easy to choose thresholds - I picked 0-1.6 as the range for “insufficient”, 1.7-2.2 as the range for “sufficient”, and 2.3-2.5 as the “excellent” range. This is what the thresholds look like on the entire score distribution:



An important point here: like all of the other parameters I’d created for the evaluation mechanics, these thresholds aren’t static - the designer of the scenario could assign them to be looser or tighter depending on their needs. Additionally, have a relatively small impact on the Consequences stage - other than the aforementioned evaluation sentence, they only change one other result (which will be mentioned shortly).

More important to the Perspective Evaluation section is the next portion: examining the player's conversation choices! Throughout the simulation, the player is exposed to a number of different perspectives; however, since they're only able to talk to a maximum of 5 stakeholders, there are at least 4 stakeholders whose perspectives won't be addressed fully. So, within the Perspective Evaluation section, the player is notified about the perspectives that they didn't see:



Legality of a lack of cybersecurity

While developing any technology, it's important to consider the legal landscape of your project. Make sure you understand any rules or regulations surrounding your technology - you could always talk to your company's lawyer to do that! It's generally a good idea to air on the side of caution with these sorts of things, so you ought to strive to make your project as compliant with the law as possible.

Health benefits for users

Understanding the impact of the project you're working on is always incredibly important, as it'll further your ability to make informed decisions about the project. Make sure to always talk to some of the potential users of your project!

Importance of secuirty for medical devices

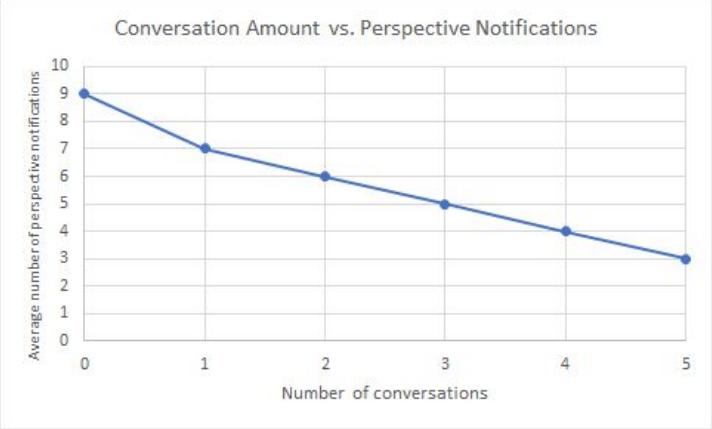
Despite being a low-priority focus of EvoHealth's upper management, the security of the APDS is an incredibly important concern. While it's fairly unlikely that someone's APDS is hacked, it'd be irresponsible to totally ignore the possibility, as the repercussions for a hack could be fatal for a user. It's important to hear co-workers' concerns about things like this; one could even go as far as talking with experts in the field to ensure the safety of the device.

An example of the different notifications that can appear in the Perspectives Evaluation section

It's not entirely trivial, though, to determine when a player has seen "enough" of a certain perspective. In order to do that, I introduced yet another metric: "perspective threshold". This is a 0-1 value representing "how much" of a perspective that the player ought to have covered in order to avoid receiving one of the above messages (0 meaning 0%, and 1 meaning 100%). You can calculate how much of a perspective a player covered using the following formula:

$$\frac{\sum_{Player's\ stakeholders} coverage}{\sum_{All\ stakeholders} coverage}$$

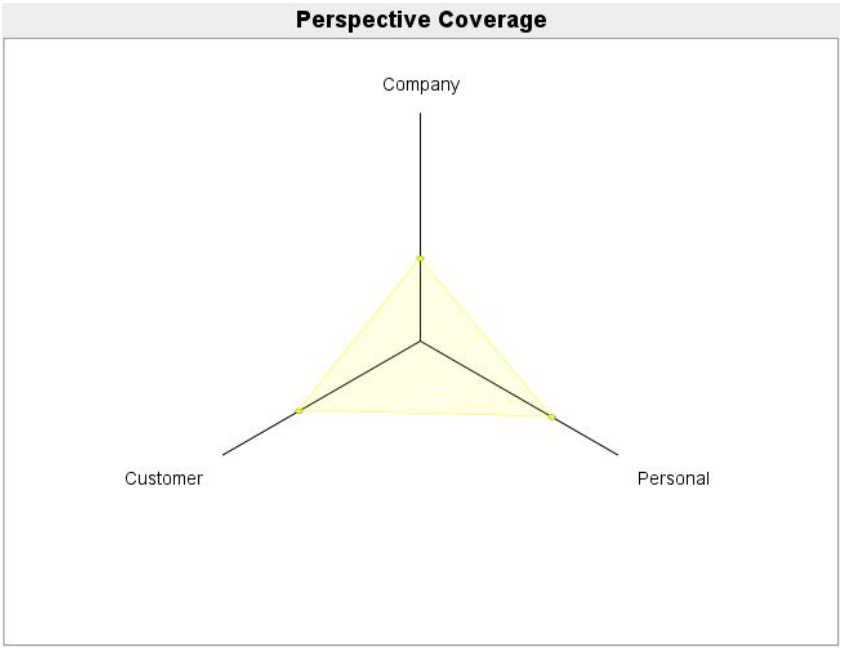
When running my initial tests of the Perspective Evaluation section, I'd set every perspective's threshold to a happy medium: 0.5. If I had set the threshold too low, I'd run the risk of players not receiving notifications for *any* of the perspectives; if I'd set it too high, I'd be equally likely to bombard players who'd already *understood* the perspectives with redundant information. At 0.5, perspectives that hadn't been at least half-covered would be represented in the evaluation. After running some more simulations, I was pleasantly surprised that things were turning out as expected: on average, the more conversations a player had, the less perspectives they'd be reminded about trying to cover.



Additionally, there were no sets of conversations that *wouldn't* lead to the player being notified about any perspectives. (Some high player score conversation sets resulted in a minimum of 2 perspective notifications.) So, even if a player has a good grasp on the scenario, they'll still have a little more to learn about some of the underrepresented perspectives.

After the Perspective Evaluation section comes the "Coverage Visualization" section. Once the player has learned a little bit about the different perspectives they missed out on during their simulation run, they're then given a number of visualizations showing off the perspectives they *did* cover.

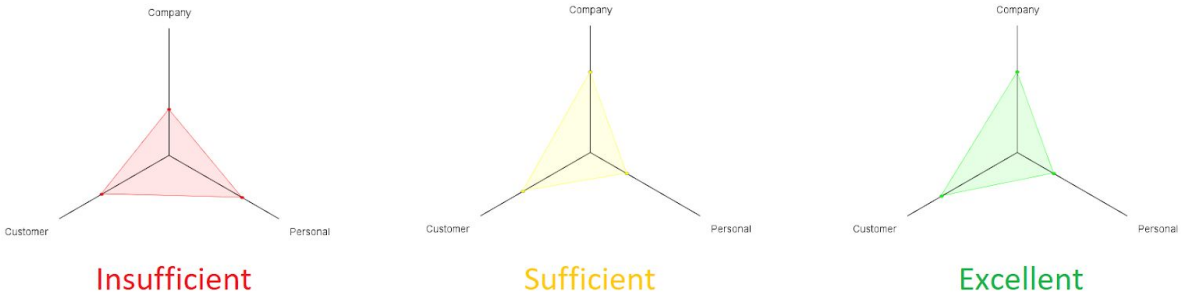
The visualizations are presented in the form of spiderweb charts. Initially, I'd included each of the 9 perspectives on the chart - each would get their own axis, and the point would be placed at the point along the axis which denoted the player's coverage of that perspective. (So, a coverage of 0 would place the point at the origin of the axis; total coverage - i.e., 1 - would place the point at the axis's maximum.) This design looked a little too cluttered, though, and so I introduced a new concept: "perspective superclasses."



This chart contains 3 perspective superclasses: "Customer", "Company", and "Personal"

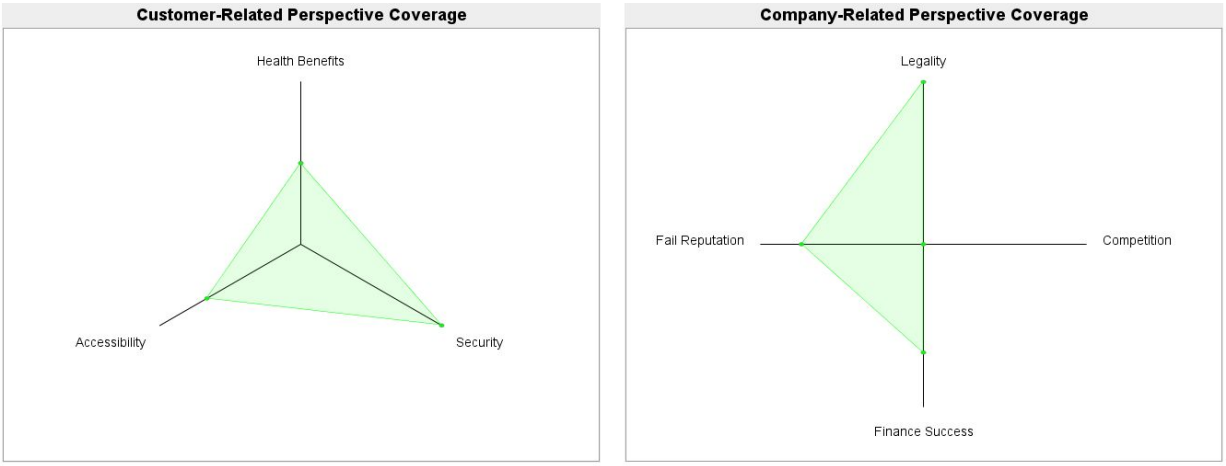
Each perspective was assigned to a particular "superclass" category - these are the overarching labels that you see in the above spiderweb chart. By grouping together perspectives of similar focus, I was able to declutter the visual of the spiderweb chart. The simplicity of only having three axes *also* allows for easy comparison between players - that way, students will be able to show eachother their charts, and professors could better understand the perspective alignment of their students at a glance.

Additionally, the spiderweb charts are colored according to whether the player received an insufficient, sufficient, or excellent player score. This choice was more geared towards professors, who could be seeing a large number of player charts. (At no point is this color coordination explained to players, as not to distract them from the contents of the chart.)



These are some of the resulting radar plots for different conversation sets

After the superclass spiderweb chart, the player is shown two more spiderweb charts. Both of these focus on the perspectives contained within particular superclasses (namely, the "Customer" and "Company" superclasses). These focused charts provide more context to the superclass charts, and provide further visualization of the player's choices.



This player seemed to focus especially on security and the legality of a lack of it!

Thus concludes the Coverage Visualization section! Currently, the simulation's codebase is able to dynamically generate both the Perspective Evaluation and Coverage Visualization sections based on the player's conversation choices. The Perspective Evaluation, Coverage Visualization, and Scenario Conclusion sections are combined into a single .PDF report, which is then shown to the player to conclude their experience with the game!

Conclusions

Research Reflection

This was my first foray into any sort of academic research, and it was certainly an eye-opening one! When Professor Osterweil had initially proposed the project, I'd had goals of using Unity to create an interactive 3D office environment that the player could explore in first-person. In this pie-in-the-sky vision of the simulation, conversations with stakeholders would change dynamically according to player input and the player's prior choices, and multiple rounds of information gathering would provide a much more detailed impression of the player's ethical thinking skills. With all of these lofty ambitions, it was only natural that the first thing I learned about the research process was "think realistically."

Professor Osterweil warned us that our goals would change over the course of the project, and he was absolutely right. For a while during the 499Y semester, I bounced around a couple of different research hypotheses. I was never quite sure what I wanted to work on - initially, after I'd gotten over the idea of using Unity, I thought about implementing a "rewind" mechanic, which would allow players to return to an earlier stage of the simulation. After some

thought, though, I wasn't necessarily sure if the players of the game would have enough content to *want* to rewind further.

So, I pivoted - next, I was thinking about some of the different ways you could introduce variation into the simulation. It was around this time when I thought about assigning certain "time constraints" on the player: under this schema, players would carry out their info-gathering stage over the course of a couple in-game days, each of which would only allot a certain amount of time for the player to converse with stakeholders. In our current simulation, this mechanic materialized as the conversation limit! Still, though, the thought remained: I wanted to introduce a decent amount of variability to the player's experience in order to keep the simulation fresh.

Eventually, though, I realized that any variation introduced would have to eventually *lead* to something. It was at this point where I started thinking about the conclusion of the simulation - how could we design the simulation so that any variation eventually lead to varied *outcomes*? At that point, I started working towards designing the mechanics that would eventually become the player score evaluation system!

Professor Osterweil really helped to set me on the right track during the entire hypothesis development process. He'd assured Kyle, Jon, and I that we were a part of a team throughout this process, and as such, we simply needed to contribute some feature or mechanism to the simulation. The entire team is hopeful that work on this project will continue in perpetuity, and therefore, a slow-but-steady stream of feature development from faculty and undergraduate researchers is exactly what's needed. In lieu of that hope, I've included a list of future features that I would have attempted to develop if I was continuing to work on the project!

Ideas for Future Research

Since my research was primarily focused on the development of a scoring system, it's only natural that some of my first ideas for future development revolve around the scoring system. Currently, it's not actually used for much - it varies a couple of sentences in the Consequences phase, as well as the color of a couple spiderweb charts. In the future, though, it could introduce a fair bit of interesting variation to the simulation! What if the player needed to *convince* their project manager to delay the artificial pancreas, and higher player scores could change the likelihood of them winning her over? You could apply this sort of variation to a multitude of different areas throughout the simulation - what if conversations with certain stakeholders changed according to your current coverage of certain perspectives? Changes like these would necessitate a significant scenario writing effort (as the variability would need to be carefully written), but it might be largely beneficial for players, who would feel as if they had much more impact on the simulated world.

On the topic of variation: I think it's incredibly important to continue to expand the simulation by increasing its variability. This ought to always be a constant goal of development - the more variation that exists, the easier it'll be to try and evaluate each different player's journey! Thankfully, there's an endless pool of techniques one could employ in order to differentiate players' experiences. Earlier in the paper, I'd mentioned the idea concept of splitting info-gathering into multiple different days. I think this would be hugely beneficial, as it would offer *much* more time for a player to gather information and understand the complexities of these quandaries. Despite spending a generous amount of time storyboarding and writing my current scenario, it's still relatively simple. If more time was allotted to the player, then conversations with stakeholders could become *much* more complex, and less like monologues.

Of course, in order to introduce multiple rounds of info gathering, one would need to create some content to fill up that space. Luckily, I still have *plenty* of ideas leftover from the Spring semester's scenario planning process. In my initial scenario layout, the player had the opportunity to make about 3-4 "actions" per day, with the entire scenario taking place over the course of a typical 5-day workweek. Under that schema, a player could perform a maximum of 20 actions throughout the simulation. Some of these actions could be conversations with the stakeholders; additional simulation time would allow certain conversations to reference eachother, opening the potential for repeat conversations with stakeholders. For a small while, we considered the prospect of reinforcing the aforementioned "convincing your project manager to delay" mechanic by creating a "petition your co-workers to help convince your manager" mechanic - I think this could *easily* add more depth to the potential conversation ideas, and could mirror a multitude of real scenarios where development teams have done exactly that.

Another possible "player action" could be the ability to repeatedly "test your code." In the scenario, this is the final week of work before sending the project off for evaluation - so, in order to increase the simulation's realism, the player's character would naturally want to balance their stakeholder conversations with actual work. "Testing your code" could be a repeatable action, after which some hidden "testedness" variable would increase - or, as is sometimes the case in development - decrease. Then, this testedness stat could be used to influence the scenario in some way - maybe the player is promised a higher bonus if their testedness meets certain benchmarks? (Therefore, the balance between working and conversing with stakeholders mirrors the balance between your personal prosperity and the potential wellbeing of others.)

Finally: some of my pie-in-the-sky ambitions have *still* remained, even after realizing that I probably wouldn't be able to implement them immediately. Each of us undergraduates were very excited about the prospect of a radically different presentation for the simulation: instead of

using text (or even video) to deliver information, we wanted to depict things in an interactive environment. We had discussions about the merits of both 2D and 3D spaces - personally, I'm still a believer in 3D. I've had some experience with 3D Unity development thanks to UMass's CS590G class, and I've been astounded with how accessible some of the development tools are. After taking that class, a student could definitely create an interactive office environment. The nature of Little JIL's agent-coordination systems lends incredibly well to process flow control - maybe someone could design a way to link it to Unity as a back-end for a 3D office front-end? If this level of simulation could be realized, then an entire new level of study could be opened: do players react differently to the quandary if they feel more engrossed in the world? What if they experienced the world through the use of virtual reality headsets - would that have any realizable impact? That might sound like an incredibly lofty ambition, but it's much closer than you might think - Unity's virtual reality tools are as easy to use as the 3D development ones.

Ultimately, the future is bright for this project. There are so many different directions that development could head in - it's pretty exciting to think about it! Plus, there are a team of wonderful people leading its development, so I have no doubt that it'll continue to be successful.

Acknowledgments

From the moment I began working on this project, I was surrounded by plenty of great team members - so, it's essential that I thank them all here! First off: both Professor Osterweil and Professor Haas have been absolutely amazing on countless fronts. They were constantly pushing us to help influence the simulation's development - according to them, since they'd never done anything like this before, we were all on an even playing field in regards to knowing

what to build next. They did an exceptional job trying to pass on their understanding of the research process, and they've left me excited to pursue more later on in my career!

Throughout the Spring semester, we spent countless hours learning the in's and out's of scenario development and storyboarding. Professor Michelle Trim offered us invaluable lessons in this field, and I'm very grateful for her involvement in the process! I'd done little in the way of creative writing in the past, so this portion of development was really eye-opening for me! This was creative writing intended to have some educational merit, too, so it was quite the new challenge. Fortunately, Professor Trim was there to help us through each and every step!

After the Spring semester was over, us undergraduate students took our summer break. Then, upon our return in the Fall, there'd been a seemingly miraculous appearance of an *entire simulation framework* for us to port our scenarios to! Of course, this was all thanks to Heather Conboy, the Research Associate who spent countless hours writing the code for the simulation! Heather's been absolutely wonderful to work around - she was always looking for ways to help us implement our ideas into the simulation, and was incredibly patient and kind when attempting to help us install each component piece of the code. Without Heather, this project would never have reached the state it's currently in.

Finally, I can't express enough how great it was working alongside Kyle and Jon. Both of them were so dedicated to the project, and helped me an invaluable amount throughout the entire process of development. They're both two incredibly intelligent individuals, and I'm really glad that I've finished these research semesters with two new friends.

Appendix

As a complement to the thesis itself, I've compiled a list of some important documents that someone might want to look through when reading this! The documents are ordered by their creation, with the oldest documents listed first. The code for the project can be found in the project's GitHub - email Heather Conboy in order to access it!

Report - Serious Games

I'd created this document at the very beginning of starting work with the Ethics Simulation team. It contains an examination of a couple popular games, and how they use certain mechanics to teach lessons. Hopefully someone reading this might be inspired to create some new mechanics similar to those mentioned in the report!

Scenario Development

This was one of the first documents that we wrote when developing my artificial pancreas scenario. Professor Trim had provided us with some various questions to prompt us thinking about things, and then we responded to those questions in the context of the scenario! This document might help future scenario developers understand our thought processes when we created one.

Testedness Storyboards

This Powerpoint contains some of the earlier storyboards I'd made for the scenario! Using this method helped to organize some of the thoughts we'd developed in the "Scenario Development" document.

Testedness - Scenario Layout

This document contains the entirety of the scenario layout for my scenario! It contains a wealth of information about why each of the conversations was written like it was - scenario developers might want to read through this in order to understand my motivations for any particular scenario design decisions.

499T Proposal

This proposal was the result of my 499Y class - it explains the project as a whole, and acts as a proposal for starting thesis-creation in 499T.

Little-JIL Code Breakdown

This is a quick list of notes that I'd taken on Heather's Little-JIL simulation framework! Heather ought to be the go-to person for any questions about the framework, but this document has some notes that ought to help someone get understand how the framework interacts.

Perspective-Coverage Matrix

This spreadsheet contains the perspective coverage matrix that I defined for my scenario!

Scoring System Code Explanation

This is a list of all of the different code I'd added to Heather's Little-JIL framework in order to implement the scoring system! This is fairly essential to understanding how the system works.

Thesis Presentation

This is a copy of the Powerpoint I made in order to present my thesis to my committee!

References

As follows is a list of the references I used throughout the writing of this thesis:

ACM Code 2018 Task Force. 2018. ACM Code of Ethics. (June 2018). Retrieved May 19, 2019 from https://www.acm.org/code-of-ethics

Alexander Wise. 2006. Little-JIL 1.5 - Language Report. Department of Computer Science. University of Massachusetts, Amherst, MA.

Bogost, I. (2007). Persuasive games: The expressive power of videogames. MIT Press

Cathy ONeil. 2017. Weapons of Math Destruction, New York, NY: Broadway Books.

Gibson, David, and Karen Schrier. Designing Games for Ethics: Models, Techniques and Frameworks. Information Science Reference, 2011.

Sicart, Miguel. The Ethics of Computer Games. MIT Press, 2011.