

## Dr. Anders Gronstedt Responses to Questions Posed in Chat 2-18-26

**Christopher C:** I'd be curious to hear more on how you create unexpected or unplanned aspects of wargaming or simulations?

Anders: You bake in uncertainty (fog of war, imperfect sensing, stochastic outcomes), you add structured injects that can fire based on player behavior (systems go down, a contact turns out to be a decoy, weather shifts, comms degrade), and you bring subject matter experts in early to seed the scenario with the kinds of curveballs that experienced operators have actually seen. On top of that, a tightly bounded large language model can help generate fresh variations and inject text, but the adjudication still needs to be driven by a real simulation model so the surprises stay plausible and repeatable.

**Iance k:** In China, many of their military installations are embedded into population centers

Anders: Exactly, and that is one reason any serious modern wargame should model civilians and infrastructure explicitly instead of treating the battlespace like an empty ocean: you can track civilian casualties, damage to critical services, and political blowback as part of the score, and you can reward restraint and precision the same way you reward mission success, so players experience the real trade space rather than a sanitized version of it.

**Pamela Z:** Am curious to hear how roles are assigned for 20,000 players and how inserts are introduced./

Anders: At that scale it is usually not one giant 20,000-person staff exercise; it is typically a tournament structure with lots of individual or small-team matches running in parallel, often asynchronous, with qualifiers and brackets, and the "inserts" are delivered as standardized updates to everyone in a round, or as role-specific injects tied to what happened in a player's match. The Marine Corps University's *Top Tactics 25* tournament is a good public example of an online, turn-based, cloud-hosted competition format that scales participation without needing everyone in one live event.

**Trish H:** The potential for stacked intelligences is fascinating. Do you think models can be trained toward human gaps like our failure of imagination in the early 40s?

Anders: Yes, and it helps to see that there is a thin line between creativity and hallucination, which Christopher C asked about below. An LLM can be wonderfully creative at proposing novel enemy adaptations, escalation paths, or "what if" twists, but if it is unconstrained it will also invent details that do not fit the physics, doctrine, or intelligence picture. The way you steer it toward useful imagination is to ground it in real cases and prior wargame moves, force it to stay inside explicit constraints, and then treat its outputs as candidate ideas that humans and the adjudication model must accept or reject, so you get credible surprises without drifting into fantasy.

**Tammy B:** How are the scenarios different from business simulations?

Anders: They are more alike than people think: both are decision games that compress reality, force tradeoffs with imperfect information, and let teams practice choices and consequences safely, then learn through replay and after action review. The big difference is that business simulations usually feel like a marketplace rather than red versus blue, with multiple competitors and stakeholders, decisions that unfold over months or years, and outcomes where more than one player can "win" depending on goals, while military scenarios more often drive toward a single contested objective in a shorter, sharper time window.

**Trish H:** Are they training for these models as part of C2?

Anders: Yes, wargaming is one of the best ways to generate structured decision data and stress-test workflows before you trust them operationally. The "stack" we described is exactly about keeping the AI helpers on a leash: the simulation and adjudication produce the ground truth of what happens step by step, while models can assist with coaching, summarizing, and proposing options without being the thing that decides reality.

**Christopher C:** you mentioned speed to create better scenarios,,, how do you prevent hallucinations... when not using a good LLM?

Ethan: The short answer is that you do not let free-form text generation drive the scenario logic: you constrain generation to a strict schema, you ground it in a curated library of approved forces, weapons, geography, and doctrine, you use retrieval so the model is pulling from vetted references, and you run automated checks that reject anything outside bounds before a human ever sees it. Then you validate by execution: the simulation and adjudication engine is the arbiter, so if an "inject" implies something physically or operationally impossible, it simply will not play out, and you tighten the guardrails until the error rate is acceptably low.

**Tammy B: Running the simulation and all the levers - is this that?**

Anders: Yes, a simulation advances the battle step by step and determines who detects whom, what gets targeted, and what gets hit, while the real-time game engine is the presentation layer that handles the user experience, visuals, rules, and interaction. Put simply, one system decides what happens, and the other shows it to you and lets you act.

**Glenn R: how do we set up a large e-sports wargaming event?**

Anders: A practical model is to run open qualifiers online, often asynchronous, promote the top performers into live streamed finals, and treat every match as data you can learn from. The hard truth is you only get there if you have a really solid online wargame first, with smooth onboarding, matchmaking, spectator and replay tools, anti-cheat, and server reliability, and most of us including us are not there yet, so the right path is usually to prove the game loop and community with smaller events, then build toward true e-sports readiness in steps.

**Sheryl C: how or where are these scenarios being hosted/delivered?**

Ethan: The simplest reliable approach is a packaged scenario library that ships with the application, so it can run fully standalone on a laptop or handheld device, including in air-gapped environments, with version control so everyone is playing the same build. As you add dynamic scenario generation, you can still keep it local by generating combinations from a vetted asset library on the device, and only syncing results outward later when the environment allows.

**DEB D: Can you apply your game model to any application to make turnaround time on requests faster?**

Anders: Yes, but it depends on how big the change is. If it is essentially the same scenario and mechanics, just moved into a different theater or wrapped in a new setting, that is usually very easy and fast. We are reusing the same core game loop, adjudication, AI behaviors, scoring, analytics, and training flow, then swapping the map, units, briefing, and mission parameters.

If the request changes what the simulation has to represent, it can be a lot more work. Examples are introducing new domains or weapon and sensor behaviors, changing the command structure and decision rights, adding multi-echelon coordination, integrating new data sources or external models, or shifting from a scripted training scenario to a more open-ended sandbox with many branching outcomes. At that point we are not just reskinning; we are extending the underlying model, content tools, and test coverage.

So yes, the model is reusable and it does help speed turnaround, but the timeline is driven by whether the change is "same game, new wrapper" or "new behaviors, new rules, new validation."

**John G: Where does human creativity fit into game development? Learning objectives**

Anders: It is everywhere that matters: humans define the learning goals, choose the tradeoffs the game should force, design review, and create the narrative and user experience that makes people want to play again. AI can help with speed, variations, and content generation, but it cannot replace the taste, judgment, and responsibility involved in deciding what should be trained and how you measure improvement.

**Tammy B: How much of this could you automate using AI? There's developing the scenario problem, but then how much of the running of the sim consequences of actions are run by AI? Right now this is a prototype you've worked on for a long time - where might AI come into play in the future?**

Anders: Over the next few years you can automate more of the content pipeline (drafting scenario variants, generating 3D assets, building coaching hints, summarizing performance, even adapting difficulty), but the core adjudication should remain simulation-based so outcomes stay explainable and testable. In other words, AI can accelerate authoring and personalize learning, while the simulation engine remains the referee that keeps the experience honest.

**Trish H: Do players also have baseline awareness of visible assets and/or intel to work with/against as part of the media and immersive context you've shown so far?**

Anders: They have a baseline common operating picture, but the uncertainty is the point: we model fog of war so players must use sensors like radar and sonar to build a track, and even then the information can be late, noisy, or deceptive, which forces the same kinds of judgment calls they face in real operations.

**Chris B: This is my first exposure to using this in a business environment. Can anyone suggest resources where I can learn more about use cases? Or to watch videos of games?**

Anders: Steam, the world's largest gaming platform, is a good place to start:

<https://store.steampowered.com/>

**Chris B:** Second question, does anyone have any ideas or resources for building wargames with small teams and limited resources?

Anders: Keep the team small but balanced: You need at least one strong subject matter expert, one game designer who can turn intent into playable rules and decisions, and one technical coder and a 3D artist who can implement and iterate.

**Christopher M:** Cost to develop something like this and timeframe?

Anders: For the prototype you saw, we kept it lean, roughly a \$140K effort over about a year of part-time development and iteration, but a production-grade digital wargame can range from roughly half a million to multi millions depending on fidelity, content breadth, multiplayer, analytics.

**Ryan C:** I'm interested in the development of business wargames. It seems like something that would vary from one business model to another. Are there any public records of these wargames run by different companies?

Anders: I'm happy to discuss further [anders@gronstedtgroup.com](mailto:anders@gronstedtgroup.com) feel free to check the business sim I showed here <https://www.gronstedtgroup.com/leadershipx>

**Trish H:** What kind of offline play is available for low bandwidth environments?

Anders: The game can run fully standalone on a single computer in an air-gapped setting, which is especially important for deployed or shipboard use. Later, you can sync logs and results to an LMS or analytics service for after-action review and longitudinal tracking. Even for multiplayer, all the graphics are run locally, and you don't need very high bandwidth.

**Trish H:** Are you also thinking of moving the problem sets to an interactive AR offering that layers decisionmaking onto their actual operational environment?

Anders: It is an exciting direction because it can tie decisions to the physical context, but we still don't have any full-fledged AR headsets; that's probably another 3-5 years out.

**Trish H:** And to extend that, do you think that starting by designing problem sets and running them through tabletop work would be a good starting point before pivoting to digital?

Anders: Yes, a tabletop is a great way to test the gameplay before you spend money on development.

**John G:** How secure is this platform?

Ethan: The platform is comparable in security to Unity or Unreal Engine, but the real risks come from the approval and compliance side rather than the technology itself due to it being an open source product that is not as commercially popular.

**Charlie S:** Wargaming outcomes...can they identify soft-skills gaps, such as "trust in leadership"...example?

Anders: It has yet to be proven out, but our assumption is that in an after-action review with an LLM-powered instructor, you can reflect on how well a player communicates, adapts, and maintains cohesion under stress, whether with real or AI-powered teammates.

**Virgil S:** Godot also has its own language. I think it has a C# scripting component, but I'm not sure

Anders: That's right: Godot supports its own scripting language (GDScript) and also supports C#

**Virgil S:** @Ethan S - Do you add in LMS (i.e. SCORM) to your Godot?

Ethan: We are proposing LMS support for both this project and our other effort "Battlefront" but have not done an in-depth integration yet. The goal is to record player data, compile it into an after-action-review for them to study, and to send that data to a service like OptDef to give the player a score and generate a next-level that pushes on their weak points. This could evolve into a more fleshed-out performance tracker, allowing users to see progress over time while highlighting strengths and weaknesses; providing personalized exercises to improve performance.