Session Speaker	Robin-Yann Storm
Session Title	Tools Design Roundtable Day 1: Design & UX
Track	Design
Duration/Format	60-Minute Roundtable
Presentation Outline	In these roundtables we will discuss and share experiences on Tool Design & Production. This is a fairly new topic that, as we learned in previous roundtables, every studio has to deal with, but is talked about relatively little. To the point of the nomenclature still being discussed as well. These roundtables will give attendees the time and place to discuss how they handle feature requests, UX & UI improvements, and tool production in general for each of our own internal tool needs. For example, at the GDC 2023 roundtables, attendees brought up such topics as: What UI/UX implementation tools do you use, what are the pros and cons of one off tools vs reusable tools, how do you retain simplicity in complex tool environments, how much time is spent teaching tools vs building them, what kind of design is best for procedural tools, how and when do you upgrade tools, how to implement user research in a good way, and how to avoid the 'One more UI feature' that creeps into huge tools. This roundtable is separate from the tools tech roundtable, which is more programming focused. Attendees have noted that this split of roundtable topics works well. I will open up the roundtable by doing an introduction, and starting off with two small topics: What is the best thing that happened for you tool design wise in the last year, and what is the worst? After that warmup, I take questions from the attendees, which will be written on the paper board in the room. These questions will then be discussed one by one. I will also prepare a list of topics to be discussed in case we do not get enough questions to fill the full 60 minutes, but considering the amount of great discussion we had last year the chances of that needing to be necessary are slim. What is new this year is the addition of a third day. Each day will have a particular subtopic we will try to align to: Tool Design Roundtable Day 1: Design & UX Tool Design Roundtable Day 2: Production & Strategy Tool Design Roundtable Day 3: User Research Examples of topics that could be discussed for the Day 1 roundtable are: 1. What user experience techniques have you seen best fit for your toolsets?

	2. Is classical UX applicable in tools, or is it very different? 3. What differences, if any, have tool service companies seen with their tool designs? Examples of topics that can be discussed for day 2 are: 1. What communication channels do you use to inform users about new tool features? E-mail? Meetings? Videos? Gifs? 2. What is a ticketing request pipeline that works really well for you? 3. In what size team did your tool request/ticket pipeline work best? Examples of topics that can be discussed for day 3 are: 1. What user research methods do you use? 2. Do you have a dedicated user researcher on your team? 3. What surprises has your research uncovered in the last year?
Description	This Tool Design roundtable is about the UX of in-house and external toolsets. Come and meet, discuss, and find out what everyone has been doing with Tool Design in the last year. Examples of topics discussed in previous years are: 1. What user experience techniques have you seen best fit for your toolsets? 2. Is classical UX applicable in tools, or is it very different? 3. What differences, if any, have tool service companies seen with their tool designs? This years' questions may be different, or similar! Feel free to join, and ask.
Takeaway	Attendees will share their experiences, challenges, and successes with tool design. They can expect to leave with a list of actionable methodologies, and real world experiences from other teams, that your team can adopt and learn from to improve in-house tool design.
Intended Audience	This roundtable is intended for tool designers, producers, tool programmers, and technical artists who wish to discuss and improve the design and efficiency of in-house tool design.

Session Speaker	Måns Isaksson
Session Title	Engineering Mayhem: Technical Deep-Dive into Environmental Destruction in 'THE FINALS'
Track	Programming
Duration/Format	30-Minute Lecture
Presentation Outline	The goal of this presentation is to give a detailed overview of the technical implementation behind the large feature which is "destruction" in THE FINALS, as well as the technical lessons we learned from working with such a complex system in a relatively small team. I will first introduce the audience to THE FINALS, explain the core concept of the game, and how it's destructive elements set it apart from other shooters in the same genre. Destruction is usually binary and subtractive and we will point out that in THE FINALS debris from destroyed buildings are fully simulated and stick around to create new gameplay spaces which creates many new design and technical challenges. We will keep this part brief and refer anyone who's interested in a more design-oriented approach to a talk by Ludvig Kingfors, where he covers the design challenges of making a fully dynamic world. Once THE FINALS and it's destruction goals have been established, we will take a deeper look at some of the studio-specific context that influenced the final technical design. Here we will cover the goals we had with destruction at Embark (persistent debris, everything should be destructible, spectacle, no "canned" animations) and how that ruled out some pre-made solutions such as Chaos in Unreal Engine. To expand on why Chaos did not work, the main drawbacks were (but not limited to) poor performance, inadequate collision detection, difficult to author the experience, and the general experimental nature of Chaos in Unreal. We will also touch on how we are a relatively small team at Embark trying to make a live-service game so managing in-house developer experience is critical for delivering a complex feature such as destruction where fast iteration is key to success. We will now move on to the core systems and concepts which collectively make up destruction. We will then walk through the arrived upon solution, any benefits and drawbacks of said solution, and finally any broader learnings that can be gained. We will also mention how having fully dynamic geometry poses

	challenges for other gameplay system that typically rely on the world being more or less static (such as audio, lighting, pathfinding, etc.). First we will will cover our workflow for pre-fracturing, importing, and processing fractured assets. The main beat of this section will be how we empower artists by moving the fracturing step into the hands of our talented Houdini artists. After this we will move on to the connectivity graph. This graph keeps track of the connections between destructible pieces for the whole level and is auto-built without the intervention of level designers. We will mainly cover serialization, replication, optimizations, and technical limitations. We will also cover what we call "Simple destructibles", a less complex destructible. Next we will focus a bit on our gameplay layer which Implements all relevant Unreal APIs such as sockets, attachments, rendering, physics, damage, and more. We will cover implementation of Unreal's gameplay API, rendering (using custom "composite mesh"), and management of physics shapes. Lastly we will have a look at strain. In this section we will cover our systemic approach to having building collapse. We will have a brief overview of what we call "the strain system" which implements an in-house constraint solver to solve for the load applied on each connection in the destruction graph which allows us to break connections whenever structural integrity is lost. Now that the audience have a good understanding of the technical implementation we will loop back to the stated design goals laid out at the start. We will examine what works well and what does not in context of these goals. An example of something that works well is automatic generation of the destruction graph. It allows level designers to work with modular, re-usable, kits, and to not really care too much about destruction details when setting up their block-out of the level ("it just works"). Something that we might have done differently is to focus more on debugging tools, especially for strain. It is an opaque system that is difficult for designers and artists to work with which leads to them often having to take help from an engineer when tweaking the system.

Description	THE FINALS is a fast-paced, competitive FPS where emergent gameplay and a focus on dynamic systems enable players to come up with creative playstyles and change the play space to fit their needs. Front and center of these dynamic environments is the destruction system, which enables most parts of the environment the be destroyed, buildings to collapse, and debris from buildings to re-shape the play-space, all while being physically simulated. In this session Måns Isaksson (Game Programmer) will walk through the technical challenges the team at Embark had to overcome to bring this level of destruction to THE FINALS. He will delve into the benefits and challenges(!) of a completely systemic approach to destruction, walk you through our workflow – from constructing and fracturing structures to integrating them into the playable environment, and explore technical aspects such as replication, simulation performance, rendering, connection graph construction, and how we manage the physics state of our simulated debris.
Takeaway	Attendees will gain comprehensive understanding of how we went about solving the technical challenges presented while crafting a large and fully dynamic destruction system in THE FINALS.
Intended Audience	This presentation is primarily intended for mid- to senior level programmers. It covers several technical aspects of a larger system and as such many implementation details will be omitted, prior experience will be helpful to infer these details. However, effort will be made to to make it comprehensible and valuable to more novice programmers as well.