Converting Traditional Training to Serious Gameplay

Matthew Becchio, Madison Quinn, Samantha Beigel, Alexis Damron, & Shane Taber
Engineering & Computer Simulations, Inc.
Orlando, FL
mbecchio@ecsorl.com, mquinn@ecsorl.com, sbeigel@ecsorl.com, adamron@ecsorl.com,
shanetaber@ecsorl.com

ABSTRACT

In recent years, traditional training methods have undergone a significant transformation with the incorporation of emerging gaming technology and methodologies. This modernization of training has gained immense popularity because game-based training has been proven to be highly effective for user retention, engagement, and real-time critical thinking. Game-based training offers an interactive and immersive learning experience where learners can engage with the content and learn by doing rather than just passively consuming information. Many organizations are struggling with how to convert their traditional training programs into game-based training. Engineering and Computer Simulations' (ECS) development team has broken down the process of converting a traditional training scenario into gameplay-driven training. They identified the obstacles they found during the process and how they overcame them. They share the lessons they learned in the development of conversion. The dissection of their process includes methods to identify elements of gameplay, establish mechanics and features, and apply game design concepts while advocating for the targeted audience to ensure a solid user experience and effective training. The team also illustrates their process for organizing these elements into a digestible, living document, which is key to the design and realization of your serious game. By following the ECS development team's approach to converting traditional training to game-based training, organizations can unlock the full potential of their training programs. This can lead to increased engagement, better retention, and more effective training overall.

ABOUT THE AUTHORS

Matthew Becchio is a software engineer at ECS. His responsibilities include designing and developing serious games and gameplay features based on client requirements and concepts, maintaining and assuring solid User Experience (UX) throughout the development cycle of the application, and tracking and running gameplay metrics/analytics. Mr. Becchio has received his Bachelor of Arts degree in Digital Media – Game Design at the University of Central Florida. He has been making entertainment and serious games for 6+ years as both a software engineer and game designer. He holds multiple awards for both professional and personal projects. Matt has experience in building applications across a range of platforms including desktop, mobile, WebGL, and virtual reality (VR). He has also been a sitting committee member for the Serious Games Showcase & Challenge (SGS&C) hosted at I/ITSEC for 6+ years, and SGS&C's 2024 industry chair, where he has evaluated and judged a variety of serious games that have used a range of hardware devices including desktop, mobile, WebGL, VR, augmented reality (AR), mixed reality (MR), and custom-built hardware controllers and devices.

Madison Quinn is the Research Analyst at Engineering & Computer Simulations, Inc. (ECS) supporting all applied research projects in support of military training research. Ms. Quinn is responsible for conducting scientific literature reviews, designing usability study survey instruments, administering surveys, conducting interviews, and supporting data collections through quantitative and qualitative analyses. Most recently, Ms. Quinn supports research on combat medicine training prototypes and assessments and has experience with immersive and innovative technologies used for training such as VR, AR, MR, haptic gloves, and haptic-based weapons. Ms. Quinn holds a Bachelor of Arts in Psychology from Southern Illinois University and a Master of Science in Industrial / Organizational Psychology from Bellevue University. Ms. Quinn is currently working on obtaining her Ph.D. in Industrial / Organizational Psychology from Grand Canyon University.

Samantha Beigel is a digital artist at ECS. Her responsibilities include utilizing computer software to conceptualize, design, and produce assets, ranging from 3D models to User Interface (UI) elements. These assets are crafted to simulate the envisioned 3D environments for various projects and applications, ensuring that the digital elements are seamlessly integrated into the full product. Ms. Beigel has received her Bachelor of Science in Digital Media -

specializing in Modeling and Simulation at Indian River State College, and her Master of Science in Interactive Entertainment - specializing in 3D Art and Game Development at the University of Central Florida. She has been contributing to the development of applications by designing and creating assets for over 4 years professionally in the military simulation industry.

Alexis Damron is a digital artist at ECS. Her responsibilities include developing assets through comprehensive work pipelines, including conceptualization, design, asset creation, integration, and optimization per clients' orders. The work she produces ranges from 3D assets, Animations, Cinematic video work, and UI designs. Ms. Damron has received her Bachelor of Fine Arts in Studio Art - specializing in Illustration from the University of South Florida, and her Master of Science in Interactive Entertainment - specializing in 3D Art and Game Development from the University of Central Florida. Alexis has experience developing assets for a variety of platforms including WebGL, VR, Desktop, and mobile applications. She has been developing assets and application for the military simulation industry for 3 years.

Shane Taber is the Chief Technology Officer of ECS and focuses on the development of innovative emerging technologies for use in military training and education. In his time at ECS, Mr. Taber has been instrumentally involved with developing and designing several flagship simulations and technologies, including the Tactical Combat Casualty Care Sim (TC3Sim) and VA Virtual Medical Center (VA-VMC). Throughout his professional career, Mr. Taber has continually advocated and explored the application and utilization of emerging media and technology for learning, education, and training, including Extended Reality (XR), AR, VR, Synthetic Training Environments (STE), and haptics. Mr. Taber has filled various duties and roles including Creative Director, Project Manager, and Development Director, and currently serves as part of the Senior Leadership Team. Prior to working at ECS, Shane worked in academic research at the Institute for Simulation and Training (IST) developing cutting-edge research projects, simulations, and technologies in the emerging technologies of MR for application within military, science museums, and cognitive behavioral therapy. Mr. Taber holds a Bachelor of Arts degree in Digital Media and Interactive Entertainment from the University of Central Florida, providing a foundation in fine art, video production, game design, digital media, and 3D visualization. He is also a Certified Scrum Product Owner and holds certificates in project management and CMMI.

Converting Traditional Training to Serious Gameplay

Matthew Becchio, Madison Quinn, Samantha Beigel, Alexis Damron, & Shane Taber
Engineering & Computer Simulations, Inc.
Orlando, FL
mbecchio@ecsorl.com, mquinn@ecsorl.com, sbeigel@ecsorl.com, adamron@ecsorl.com,
shanetaber@ecsorl.com

INTRODUCTION

In the dynamic realm where entertainment meets purpose, developers can dive into the art of creating a game design document tailored for serious gameplay. It's no secret that the training, simulation, and modeling industry has been leveraging modern game development technology to improve the effectiveness of training for years. However, there is often a discernible difference when comparing the products between the simulation and modeling industry and the games industry, and not just visually or technologically, but more so philosophically. While the development tools are adopted into the pipeline of developing training software, too often, the approaches and design philosophy used in the games industry are overlooked. Many developers tend to rely on traditional approaches to developing training structures and wedge those frameworks into a technology that it was not necessarily built in parallel with. So, how does one also adopt the commercial game industry's philosophy while maintaining traditional training structures? The answer is part of what fundamentally defines a serious game, and that is a cohesive design that keeps the player engaged with the learning objectives while completing the goals of the game. One of the most powerful tools used in the video game industry that can help to achieve this goal is a Game Design Document (GDD): a living, breathing document used to design, maintain, and inform everything and anything about the design of the game. This paper aims to explain the different components of a GDD and how it can be used to enhance the development and implementation of conventional training. Furthermore, it will highlight the benefits of using a GDD and provide examples of how ECS has leveraged a GDD in a case study of its own product. It is important to keep in mind that this is not the end-all format solution for a game design document but is instead a breakdown of the process the ECS development team used for the case study and should be used as a guide that can be expanded or condensed to fit the needs of the game.

GAME DESIGN DOCUMENTATION

For the purposes of this paper, a serious game recently developed by ECS will be used as a case study to provide specific examples and context for the importance and application of a GDD for converting traditional training to game-based training. This game was developed for a government agency as part of a broader curriculum and was intended to provide a capstone experience that was based on existing traditional tabletop practical exercises that were already being used. The ECS approach resulted in a design for a turned-based resource management game. The below sections highlight the key components of the GDD, why they are important for serious games, and how ECS implemented them in the design of the DESC Capstone Exercise.

Game Overview

The first core section of the GDD is the Game Overview, which is the first impression of the game and a glimpse into its composition. It is designed to be the surface-level lens of the game and is where anyone invested in its development, whether that is a client, developer, SME (subject matter expert), or product owner, should be able to grasp the vision and direction of the game. The core elements that make up the Game Overview are:

- Game Concept
- Target Audience
- Learning Objectives
- Look and Feel
- Core Features
- Game Flow Summary

The game concept can be thought of as an elevator pitch of the product; it is a concise verbal depiction of the game that helps to conceptualize the overall design and intent and sets the stage for what is to be expected as the reader traverses the document. Whether it is a serious game or a simulation with game-like features being developed, some key points to touch on are the purpose of the game, designs, and key selling features to intrigue the reader and get them in the mindset of the implementation of those features. Focusing on these critical points was the mentality that the ECS development team took when crafting the game concept, which can be seen in the example below:

> *The DESC Capstone Exercise is a strategy resource management game based on the JP 4-03 scenario in Appendix C that puts the player in the role of a JPO personnel to simulate, train, and improve planning, decision-making, and contingencies execution for petroleum management. The game utilizes an Action Point System, simulating real-world effort, to challenge the player to make calculated moves and a turn-based Day Cycle System to limit the number of actions a player can perform each day, necessitating daily planning and execution of their contingencies.*

With the stage set, the focus is turned to the target audience. Here is where the end user is defined by asking questions like, but not limited to: What is the learner's role/position in regard to their training? What level of training are they currently at? What is the age group of the audience? The targeted audience is the most critical component to consider since they should always be advocated for throughout the design of the product. It may guide the design of the content that is displayed to the player, the interfaces that they interact with, and the prior knowledge assumption going into the content being covered in the game. In ECS's GDD, the target audience was identified as potential and current JPO personnel seeking IMI 4-level training. Identifying the targeted audience as both potential and current JPO at IMI 4-level training early on helped to inform the team and design the expected level of knowledge and experience the player has before going into the game.

Next is a subsection that the training and simulation world is very familiar with and is an element more unique to the serious games' world than entertainment games: Learning Objectives. What are the outcomes the end user will attain from playing through this game? These points help ground the designs and systems to maintain their focus on the end goal, which is ensuring the user is meeting those outcomes. Similar to the results above that helped to determine the level of learning from the targeted audience, identifying the learning object up front ensured that the design to follow remains focused on the goal of the training. In the case of ECS's turn-based game, the Learning Objectives were specified as critical decision-making, calculated decision-making, situational awareness, contingency execution, and resource management.

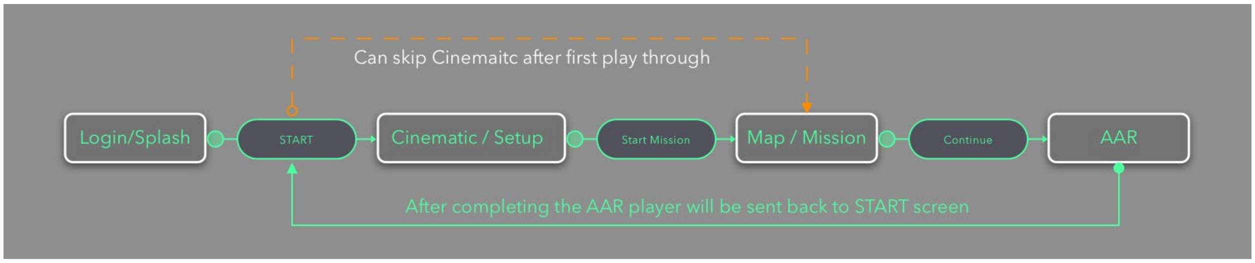
The subsection of Look and Feel explains the visual aesthetic and overall UI/UX (user interface/user experience) feel of the game. When it comes to presenting the gameplay and information to the user, what aesthetic and impression should they expect? Familiarity might be the goal with the overall UX to make the visuals feel recognizable, or maybe simplicity is ideal to minimize distractions from the content. Remember to keep the target audience in mind when deciding on the look and feel of the game. Since DLA-E has an existing aesthetic, ECS decided to maintain that look and establish the look and feel by using a minimalist color scheme that uses the DLA-E website as a reference baseline for the visual vocabulary and branding with the intent to create familiarity for the end user.

Gameplay mechanics are the boundaries that define what the game is, and it's in the Core Features that introduce the systems that create those mechanics. Core Features also describe overarching elements that detail the game. While the Game Concept subsection teases the prominent selling features, it's here where a list of all features should be aggregated in a concise, high-level appellative name. What systems are used to support the mechanics and rules of the game? What features make the game stand out? For ECS's development team, answering questions like these created a jumping point to expand and explore the mechanics of the game and the relationship between the systems. The following is a breakdown of the core features ECS defined in GDD:

- *Action Point System – Spend points to make moves*
- *Day Cycle System – Execute queued moves and simulate turn-based gameplay*
- *Reports System – Gives daily report briefings*
- *Resource Management – Maintain fuel levels and equipment*
- *Random Obstruction Event System – Introduces random events for the user to overcome*
- *PC Standalone & WebGL*
- *Realistic environment with 2D icon overlay*
- *Top-down Camera Game*

- *Based on the JP 4-03 scenario/simulation*
- *AAR – Gives a breakdown of the player's performance*

The last subsection under Game Overview is the Game Flow Summary. This subsection is a good example of how graphs are a great visual tool to use throughout the creation of the document. The game flow helps to establish a high-level understanding of how the player moves through the game from start to finish. What is the landing page the user is first greeted to? At what point does the player enter into the core gameplay? Where does the user go after completing the game? By solving these challenges, it starts to paint a picture of the flow progression of the game and creates the base structural foundation of the user experience. In ECS's design document, it is established that the user first lands on a login page and then progresses to a cinematic, followed by the gameplay mission, and ends in an AAR to return them back to the start, as seen here:


Can skip Cinemaitc after first play through
Login/Splash
START
Cinematic / Setup
Start Mission
Map / Mission
Continue
AAR
After completing the AAR player will be sent back to START screen

Narrative and Setting

The Narrative and Setting core section helps to give background and atmosphere to the game and can be broken down into two specific elements: Narrative Concept and Game Setting. The narrative concept assists in staging the world in which the game is taking place. Some games or simulations may have a more involved narrative, while others leave out an overarching story. The narrative concept's goal isn't necessarily to explain the story but rather to explain why the player, as a character in the world, is there. In ECS's game, DESC Capstone Exercise, since it is driven by a pre-existing scenario, the lore had already been established in the form of a situation and mission. However, even if the product was, for example, a maintenance trainer, defining a narrative on why the player is there performing the maintenance can help to drive the design mechanics and setting of the world.

Following the narrative concept, the game setting constructs the world in which the narrative takes place. What and where is the physical world the game takes place in? What does the environment look and feel like? Are there multiple locations? Just like the narrative, the game setting can also drive the mechanics that exist in the world. For example, in the DESC Capstone Exercise game, the setting takes place in a fictional world that is a play on African countries and, given the concept of the game, features like the Random Obstruction Event System allowed the introduction of a weather event mechanic that can appear in the world causing environmental effects. Something to also consider in the Narrative and Setting section is the approach in which storytelling is introduced and told. This could be in the form of text, narration, or even in a cinematic. Based on the identification of both the narrative and the setting, in the case of the DESC Capstone Exercise game, the team decided to create a cinematic to both introduce the narrative and setting to the player and used it as a vessel to familiarize the player with icons and elements they will be exposed to throughout the game.

Gameplay

With the overview of the game established and the narrative and setting grounding the world in which the game exists, next is to adjust the lens beyond the surface level and start to explore what components comprise the gameplay. The assembly of elements that make up the Gameplay core section reveals the goals the player is trying to achieve, the path needed to achieve the goal, and the challenges they can encounter. To help interpret the gameplay section, it can be broken down into the following subsections:

- Objective
- Game Progression
- Challenge/Puzzle Structure
- Core Gameplay Loop

The objective defines what the destination looks like for the player. What does the player need to do to reach the final goal or complete the game? What type of actions will the player make to progress to that point. What does the end look like to the player, is there a conclusion to the narrative, or does it provide analytics and stats to the player based on their performance? These are all questions that should be answered by the objective, and those answers will guide the mechanics to ensure they meet the requirements of the objective. That is how ECS derived the objective statement for the DESC Capstone Exercise game. The objective focused on the actions and decision-making the player must perform and a description of what the after-action review (AAR) recounts:

> *The player needs to effectively maintain a steady fuel supply to each military base to support the mission and prevent the supply in any given location from reaching 0. The player will also need to use critical decision-making to deal with unexpected obstructing events preventing fuel flow. The win/lose state isn't explicit and is more of a performance over the duration of the mission with an AAR breakdown of their fuel management for each location during the mission.*
>
> - *Successful Performance: The player successfully maintained fuel levels and prevented any fuel levels from hitting 0 throughout the mission.*
> - *Poor Performance: The player failed to maintained fuel levels and fuel levels hit 0 one or more times at any given location.*

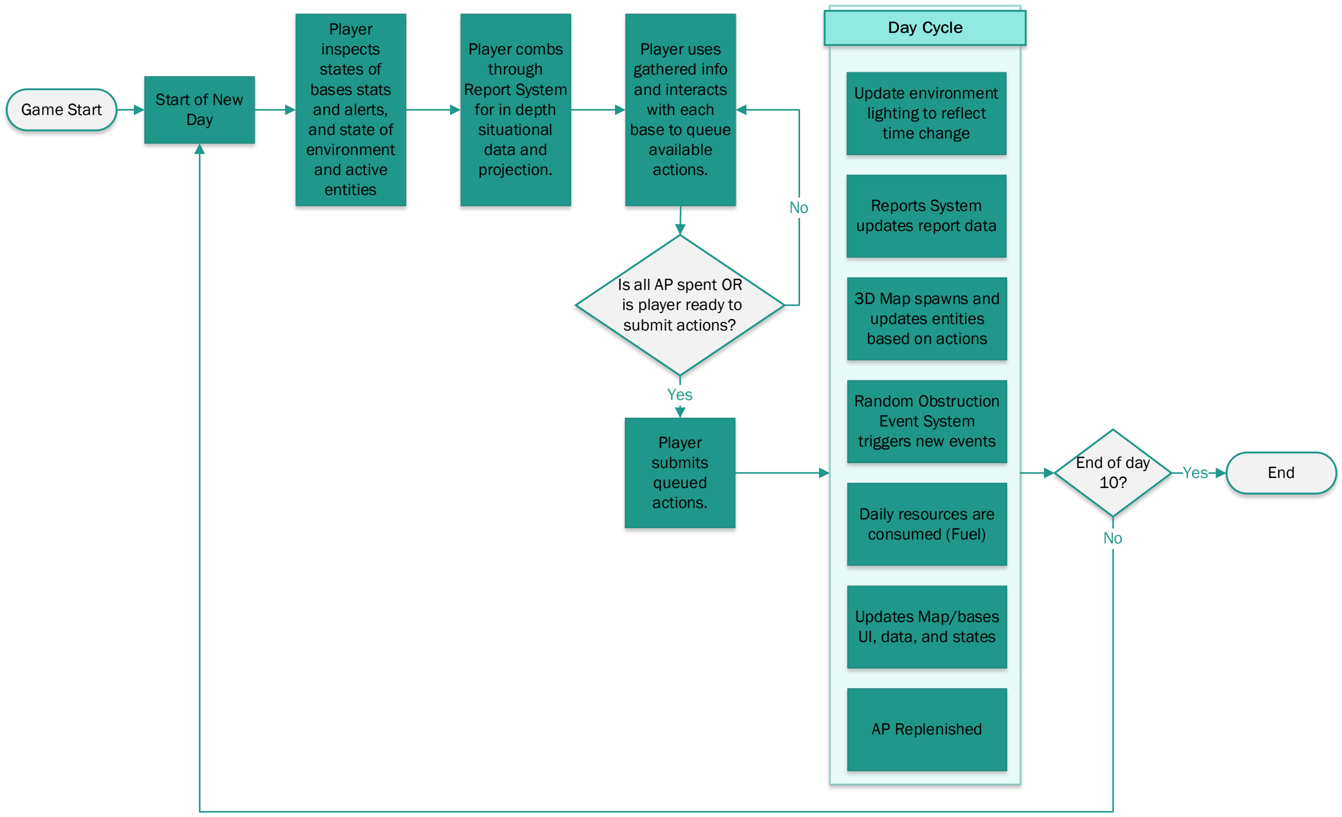
Tied to the gameplay objective is the game progression, an element that explains how the player advances time while playing. For example, it may be real-time based, where the world moves alongside the player's actions. It might be smaller scenarios tethered by a UI based on a sequential list the player must complete chronologically. Progression can even be tied to the mechanics and systems design for the game. This is the case of ECS's game, which uses strategic turn-based progression where a day cycling system is used to give the player control over when they want to progress the game and introduce challenges with each cycle the player advances. By identifying the game to be a turn-based progression type, this allow the ECS development team to focus on designing the mechanics to be center around requiring strategical critical thinking which align with the learning objective determined earlier on.

With reference to challenges the player faces during the progression, the challenge/puzzle structure subsection can start to be molded. Some questions to help design and detect the challenges are: As the user plays the game, what hurdles are introduced to them? Is there a core challenge that persists through the entirety of the gameplay? Are there more minor challenges in passing? Are there any restrictions put on the player? Here, there might be references to some of the systems mentioned in the above sections, which speaks to the value of scaffolding the information in the design document. For instance, in the DESC Capstone Exercise game, the challenge/puzzle structure is stated as follows:

> *The core puzzle structure for the player is managing the fuel levels of each military base based on daily consumption and player-driven fuel transport requests. This core puzzle is challenged further by introducing a Random Obstruction Event System that can create delay, cancelation, or request rejections for fuel as the player maintains each base's fuel levels.*
>
> *The player will also be restricted by the number of moves that can be made per turn by an Action Point System. In each turn, the player will use their action points to perform actions and must prioritize those actions per turn based on the number of action points available and the cost of an action.*

Where the persisting core challenge is the management of fuel levels, the smaller challenges are the injections of random obstruction events, and the restriction is the number of moves, in reference to the action point system, a player can make per turn. Having these challenges and puzzles defined will help when thinking about the game's flow state, a balance between difficulty and capability, and promote ideal engagement.

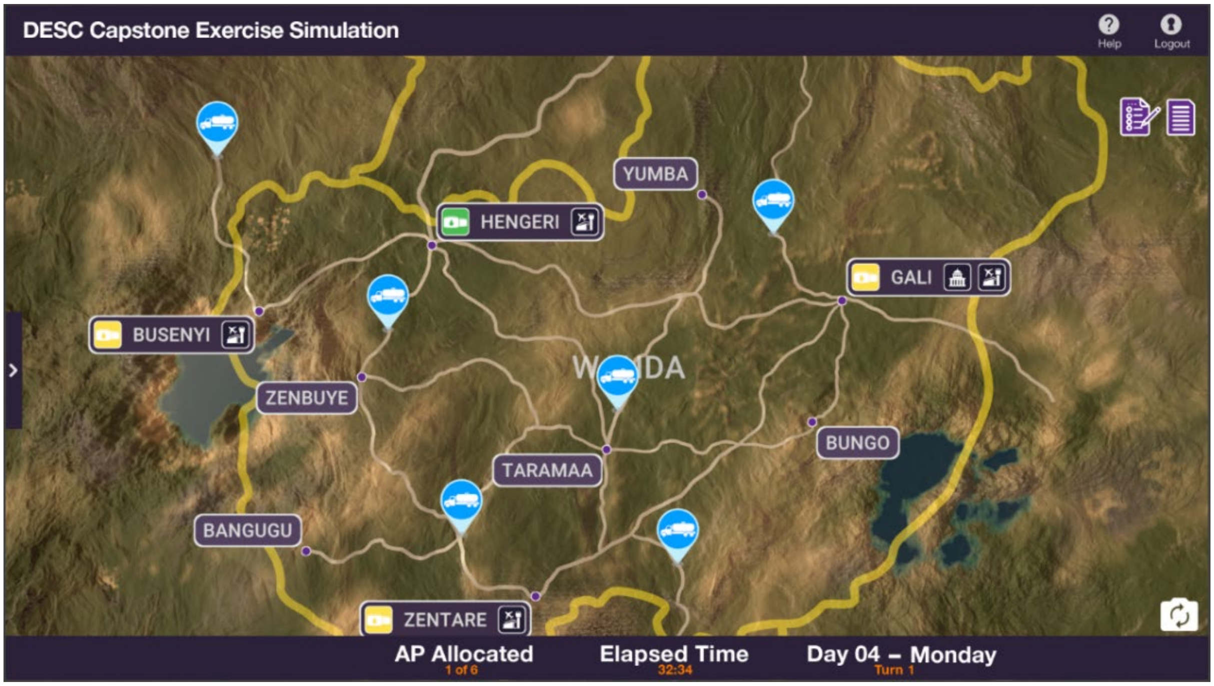
Now that the objective has been defined, the progression laid out, and a structure for the challenge has been created, the Core Gameplay Loop can be arranged. Much like the Game Flow Summary comportment in the Game Overview, this subsection undoubtedly benefits from a flow graph showcasing the literal gameplay flow from start to finish. This should be a high-level diagram that focuses on the core actions and decision-making points that progress the game and the state of any system that also changes with progression. Using the information established so far, ECS's development team is equipped to iterate through a series of gameplay flows until determining the ideal flow. The following image includes a diagram for ECS's game that demonstrates the gameplay loop for each cycle of a day until the final day of the game is reached:


Game Start
Start of New Day
Player inspects states of bases stats and alerts, and state of environment and active entities
Player combs through Report System for in depth situational data and projection.
Player uses gathered info and interacts with each base to queue available actions.
Is all AP spent OR is player ready to submit actions?
No
Yes
Player submits queued actions.
Day Cycle
Update environment lighting to reflect time change
Reports System updates report data
3D Map spawns and updates entities based on actions
Random Obstruction Event System triggers new events
Daily resources are consumed (Fuel)
Updates Map/bases UI, data, and states
AP Replenished
End of day 10?
Yes
No
End

Game World

Game World is a much more physical in-depth look into elements of the Look and Feel subsection of Game Overview and the subsections of Narrative and Setting that verbally describe the world. This section is comprised of two main components: Game World Aesthetic and Level Design. In the Game World Aesthetic, the style of the environment, both the physical world and interface integrated into the gameplay, are characterized. Is the game targeting a realistic environment or electing to go more stylized? Is the world 2D or 3D, or is it a mix of both? Does the interface reflect the same style of the environment, or is there a contrast between the two? This section might even include concept art or image references for inspiration. Defining the aesthetic of the world is the first step in understanding the art direction for the game.

While aesthetics is the decision on the type of paint and colors to use, the Level Design is the canvas and utensils used to paint. It is here where the structure of the world is built. How are objects related to one another? How does the layout affect the player's experience in how they interact with the world? Does the level design partake in the challenges and puzzles? These are all key questions to think about when designing the level. Even if the gameplay is relatively stationary or static, there should be logic to support the layout decision of objects in the world to interact with. While both are subsections of the same section, both aesthetic and level design can operate in parallel while both respecting each other. This means that the level design structure can be built without the full realization of the world's aesthetics. Often, a level design can start as two-dimensional and, if applicable, can be blocked out, or gray boxing, in three dimensions (using three-dimensional primitive shapes to build out a level). The value in this allows thorough testing of a level design without the initial need for detailed art assets and direction, although it is valuable to keep any elements from the aesthetic of the world that can affect the level in mind, for example, topography or land/cityscapes, etc. In the case of the DECS Capstone Exercise game, the scenario called for a map of the fictional area. The aesthetic that was agreed upon was realistic terrain with DLA-E themes, interfaces, and icons overlaying on the map. The level design came in the form of architecting a road formation that connected major and minor cities and a system that allowed the developers to adjust the distance and time it takes to travel across the roads to balance the challenge of transporting resources seen here:



Mechanics & Systems

Starting with the baseline information from the Game Overview, understanding the world established by the Narrative and Setting and Game World core sections, and adhering to the governance described in the Gameplay, the mechanics of how everything operates can begin to be flushed out. It is the Mechanics and Systems that make up the major substance of the game design document. It is akin to unfolding the Monopoly board game manual and deciphering all the rules on how to move around the board, how the different mechanics operate, like property transactions and chance cards, and how the players interact with each component and system on the board. When thinking about what the mechanics are that make up the gameplay and, more specifically, how to translate training steps into a mechanic, it is best to break down all behaviors into their own uniquely defined system and rules they conform to. The structure of this section to help with dissecting the rules and mechanics are as follows:

- Rules
- Movement
- Interactions
- Name of Mechanic/System (this subsection would be created for each mechanic/system in the game)

The rules of the game are a set of instructions that both delimitate and empower the player's actions. This subsection will be the most dynamic since the rules are constantly being calibrated, not just during the design phase but also during development and even after the release of the product. Traditionally, balancing the rules is the key to a good user experience, but it can often be particularly difficult when it comes to translating traditional training into a serious game. This is due to the content and material at hand and is often the learning objective that may create its own set of rules restricting the designs. ECS encountered a similar issue with the DESC Capstone Exercise game, where it was requested that the learner review external documents at the start of the game to reflect the situation in the real world. This request required the team to adjust the rules that defined the starting state of each base location and to create solutions to ensure the player understood that starting state and why the rules are defined in this way. Ultimately, the key thing to remember is the targeted audience, advocating for both them and the objective they are attempting to learn.

When fabricating the rules of the game, it might be useful to start by referencing the Core Features subsection of the Game Overview and breaking down the rules for each feature type. What are the rules that the player is bound by for each feature, and how do they relate to one another? While the rules are being created for the game, detailing said rules will be much more comprehensive than what the end user may be exposed to since the GDD's target audience is someone invested in the development of the game. Since the game design document will be the point of reference for both the rules and mechanics of the game, it's essential to be as detailed as possible, including both data and states presented to the player going on under the hood. Always be asking, "If I read these rules, do I have a complete understanding of both the boundaries and the toolsets to play the game?"

Both the Movement and Interactions subsections help to define the input methods of the player to interact with the world and the interfaces. Does the game require a click to move, or is it using gaming conventions of WASD keyboard or gamepad movements? What are all the input actions the player can perform to interact with the world or the UI? If applicable, it might be useful to define a key mapping table here to map inputs to gameplay actions. Movement is not just limited to the player's movement but can also define how other entities traverse the world. After deciding on the movement, the player would click and drag in the horizontal axis with camera movement dollying in the vertical. This second point on deciding the movement for all other entities was determined to be valuable to explain in this section as well. The team decided that there was a collection of units that moved on their own per day cycle and were given a detailed description of how they would travel along their paths and the distance they could cover per turn. It is necessary to consider all types of interactions and movement types when unraveling these two subsections.

The following section will be a deep dive into all the mechanics and systems in the game. Where the Core Features introduce the titles of the systems, and the Rules explain the instructions for the player to interact with the systems, this subsection breaks down how exactly each system/mechanic works. To help get a surface-level understanding of each system, first, start with a description of what the system is, what content, data, and entities it manages, and what its purpose is in regard to gameplay and/or learning objectives. Then, follow with the dynamics of the system. How does the system interact with the rest of the world and other systems? What are all the states of each system, and what triggers their state change? Like in the Rules definition, each system may manage its own set of data, times, or values that should also be established here and balanced throughout development. It should be expected that this section will change and update over time while the game is play-tested and iterated upon. Keeping that in mind helped the team test, iterate, and update the sections like the Daily Report System and, in particular, the Weather Reports. Throughout development, the team made many adjustments to the number of days displayed on the report to optimally balance between informing the player enough to make calculated moves while also giving a bit of ambiguity to keep them vigilant.

Entities

Much like the list of systems in the previous section, this section is a comprehensive characterization of all entities in the game. In this instance, entities are defined as any presence having objects or actors that engage with the player. This could be other characters in the game, objects or tools the player can interact with, or units or objects that move around the play space. If it's a character, maybe explain the background of the character and how it relates to the gameplay. Consider the environment/locations where an entity can be found. How does it interact with the player? Is the entity related to the systems listed above? What are the unique mechanics of the units? In the case of the DESC Capstone Exercise game, all entities were able to be identified upfront and were determined to be dynamic units located on the map, including all traveling vehicles and aircraft types, and each city and base the player can interact with.

User Interface

The final section covers all the types of interfaces the player engages with. The main focus of the User Interface section is the structure of the interfaces, the type of information they provide, and how they present that information to the player. This section may have some crossover with the art equivalence of the GDD and the ASG (art style guide). However, where the ASG focuses on creating a guide around the depiction of UI elements, think of the GDD section as the verbal depiction of the UI that gives their design decision reason. The section should initially be comprised of wireframes of the UI and primitive skeletonized diagrams that block out the interfaces. These wireframes should give each element an explanation and show its relationship to other wireframes. Just like every other section, the UI diagrams should also evolve to include completed UI elements to showcase the achieved design to the reader of the GDD. The User Interface section can be broken down into the following subsections:

- Game Menus
- Gameplay Interfaces
- After Action Review

The game Menu should include all non-gameplay-related menus. This can include, but is not limited to, login menus, main menus, loading screens, pause menus, help menus, and options menus. Blocking out each menu and creating a diagram relationship between each one will help to understand how each menu flows to the next and will give structure to the progression of the application as a whole. With each menu, consider all the interactive elements existing in it, how the player interacts with different elements, the expected results when the element is interacted with, and any logic tied to that interaction. For example, for the login menu, the player might be presented with a username, password, registration, forgot password, and exit button. What happens with each element when the player interacts with it? Does clicking the registration switch the state of the UI? Without getting into the technical architecture and reserving that for the TDD/SDD (technical design document/software design document), how is the login process handled? What does the feedback look like for the player when they perform a successful and failed interaction in a menu? These are all things to consider when blocking out the game menu diagrams. By taking this iterative approach to building the menus and focusing on the structure and interactions, the team was able to make both small and large incremental changes upfront with minimal impact.

Once the player is in-game, they may be presented with a series of all-new gameplay-based interfaces. This might include HUD (Heads-up Displays), popups, equipment/inventories, or maps. These interfaces can be broken down into four types of in-game UI:

- Non-Diegetic – Interfaces that exist in the game space but not in the world.
    - Example: 2D-based HUD in the screen space
- Diegetic – Interfaces that exist in the game space and in the world.
    - Example: Immersive HUDs that are part of the world space
- Meta – Interfaces that exist in the world in screen space but are not tied directly to the game space UI.
    - Example: Immersive overlays giving the state of the player like low health indicators.
- Spatial – Interfaces that exist in the world but not in the game space.
    - Example: World-based racing lines or character nameplates or icons.

Calling back to the main focus of the User Interface section, understanding what category the UI elements classify under will help design their states and how they provide information to the player. In the example of the DESC Capstone Exercise game, the team was able to identify both a spatial representation of UI data as well as an expansive non-diegetic UI of the same object. Base icons exist as spatial UI elements that are locked to their location on the map and indicate the current fuel levels at that base through a color state indicator. However, clicking on that UI element would open the non-diegetic UI popup that would give the player additional information about that base. Just like the Game Menu subsection, consider all the interactions with each element, the feedback it provides the player, and the state in which it might change.

An argument can be made that the After-Action Review (AAR) can be nested within either the Game Menus or Gameplay Interfaces. However, given that it exists as both a menu whose data is a direct correlation to the gameplay results and its degree of importance to training and serious games in regard to evaluating a player's performance to achieve learning level objectives, it deserves its own dedicated section in a GDD. Like the previous sections, the AAR should also be blocked out using a wireframe, however, understanding what data is being collected and analyzed and how it will be displayed to the player should be considered first since that information will dictate the design and structure of the AAR UI. The key thing to always remember is the targeted audience. What data is valuable to the player to view for them to understand their performance and how, if, and where they need to improve? Traditionally, in many training applications, the AAR tends to target the instructor, which is still very valid for evaluating the learner. Where the AAR might vary in a serious game is the data is ideally targeted for the player’s review in order for them to improve as they progress the game or when replaying it. In the case of the DESC Capstone Exercise game, the AAR came in the form of displaying the average fuel levels of each base location and a performance score based on the player’s achieved averages, challenging them to improve their averages with the following playthrough.

DISCUSSION/FUTURE RECOMMENDATIONS

The GDD is an essential tool that allows both developers and clients to gain a thorough understanding of the game's design, mechanics, and benefits. By creating a comprehensive GDD, everyone involved in the project can stay on the same page and work towards a common goal, resulting in a successful and satisfying end product. A fully realized GDD can also reduce onboarding time for any future developers by creating a turnkey document for all information in regard to the design of the game. It is recommended to ensure that the client is informed about the document's requirements and provides feedback to help the developers understand their goals. Additionally, it is important to thoroughly outline all the proposed information to ensure that the entire team is on the same page. This will allow clients to make requests based on the contract and enable developers to understand what is expected of them, avoiding any surprises on either side. Often, clients have difficulty articulating their expectations to the Principle Investigator or Project Manager, who then has to explain them to the developers, leading to misunderstandings. Requiring a GDD in any simulation for serious game development will eliminate miscommunication or unmet expectations from a contractual perspective.

CONCLUSION

In conclusion, the creation of a game design document for serious games serves as a tool to bridge vision and realization. The GDD is a tool to help break down traditional training methods and reconstruct it using a structure designed to immerse and captivate the learner. This allows for the fusion of entertainment and purpose. As mentioned at the start, the GDD is a living breathing document. While the game evolves and is iterated upon, the GDD should reflect those changes, this is a philosophy the ECS team maintained throughout the development cycle. The team also found that giving ownership of the GDD to all developers involved, meaning anyone could contribute to the design or description of the designs, ensures a collaborative and supportive development environment. By detailing the elements that shape immersive experiences, from narrative intricacies to gameplay mechanics, developers not only harness the power of engagement but also leverage the potential for impactful learning and meaningful change. As the final notes are placed in the comprehensive document, it bridges the gap between intention and execution, ensuring that the resulting game not only captivates players but leaves an indelible mark on the landscape of serious gaming, transforming it into a catalyst for education, empathy, and social impact.