Game Engineering: Formalizing Experience Based on Pacing & the Topology of Play

Anonymous Author
organization
city, country
email

Anonymous Author
organization
city, country
email

Anonymous Author
organization
city, country
email

***Abstract*—Central to game design is the player experience, a relation characterized by the inherently indirect nature of how the experience emerges through play. This second-order design problem of games introduces a significant challenge for the game development process. In this paper, we propose a game engineering process centered on the concept of *formal experience*. We detail a semiformal approach to the planning of game experiences. Specifically, we introduce three critical methodologies: the *Experience Description*, the *Pacing Diagram*, and the *Experience Chart*. The proposed approach enables developers to more effectively align their creative efforts with the intended play experiences within a game engineering context.**

***Index Terms*—Game Engineering, Game Design, Design Engineering, Play, Player Experience, Pacing.**

I. The Play Experience

In the domain of game design, the centrality of the player's experience cannot be overstated. This experience is a subjective and human encounter that arises from interaction with the game–by playing the game. When designers and engineers create games, the game only plays a secondary role. The primary focus is the experience a player has when interacting with the game. "When people play games, they have an experience. It is this experience that the designer cares about. Without the experience, the game is worthless" [1, p. 10]. This insight is of utmost importance and bears repeating: "The game is not the experience. The game enables the experience, but it *is not the experience*" [1, p. 11].

This central role of experience is not unique to games. It is the case for all types of entertainment: books, movies, poems, plays, music, rides [1]. The experience arises from interaction with the medium. Weinberger highlights the essence of this perspective: "... every reading of every poem, regardless of language, is an act of translation: translation into the reader's intellectual and emotional life. As no individual reader remains the same, each reading becomes a different—not merely another—reading. The same poem cannot be read twice" [2, p. 43]. The same is true for games: playing a game is an act of translation into the player's intellectual and emotional life–a subjective experience. As game developers, "... we never truly see the output of our work, since it is an experience had by someone else and, ultimately, unsharable" [1, p. 11-12].

This indirect nature of game experience and the corresponding challenges presented towards the design process is identified as a **second-order design problem** [3] "...meaning that designers are communicating with players indirectly through their games" [4, p. 165]. "...[Y]ou are never directly designing the behavior of your players. Instead, you are only designing the rules of the system" [3, p. 168]. Game design in the context of the second-order design problem becomes "...the process by which a game designer creates a game, to be encountered by a player, from which meaningful play emerges" [3, p. 80].

To understand this indirect relation between artifact and experience, we can use a perspective from psychology: **mental models**. The idea of mental models is as follows: Our mind creates a mental version of relations between objects and events to perceive reality [5]. By interacting with the artifact, the player creates a mental model. "...[T]he structure of mental models 'mirrors' the perceived structure of the external system being modeled" [6, p. 16].

Two aspects are of special importance for the relation of artifact and experience: (i) the perceived structure and (ii) the process of modeling. The first point (i) is related to the fact that the mental model is not modeling the system but what is perceived. *Perception* influences the mental model by introducing error. The second aspect (ii) is related to the nature of models in general: models are a *simplification* of a situation. That means some but not all aspects of a system are relevant and represented. A model is not an exact representation of the external system. Players filter the perceived information, and only some aspects end up in their mental model. This can be interpreted as follows: from the perspective of the system, the model has some errors, and the truth lies in the system. The error is introduced by *perception* (i) and *simplification* (ii). This, however, is not of special importance to the player: the mental model based on perception and simplification is their truth. The experience corresponds to the mental model, not the external system. This is what we describe as the "act of translation into the player's intellectual and emotional life."

II. The Game Engineering Problem

The goal of the game development process is to create an experience that is planned (to varying degrees). We have a specific idea in mind–the plan–and try to shape the reality–the experience emerging from playing the game–to come as close as possible to our plan. This process is known as engineering, with the plan as a conceptual model of the system.

Conceptual models are representations that describe systems and are used by designers, scientists, and engineers. Models are a simplification of a situation. For conceptual models, that means some but not all aspects of a system are of interest. The challenge is to design the simplification so that the model's behavior corresponds to the system's behavior and it is possible to perform analysis on it. [7]

”The *engineering method* starts by producing a model (for example, a blueprint, or a specification) of what we want to build, and proceeds by building it. We then aim to show that what we built is in fact an implementation of the model. Such a verification process compares the outcomes of the system to the predicted outcomes of the model, by testing and model checking, which is in many ways similar to scientific experimentation...A main feature of the engineering method is that the “truth” is in the model, while the system is in principle never fully correct” [7, p. 80].

The challenge in game development lies in shaping the player's subjective experience through the game artifact, as direct creation of this experience is not possible. The engineering process must facilitate a comparison between the system under construction and a conceptual model, encompassing the entire play system—human, machine, and their interaction—despite the subjective experience remaining beyond direct access. This necessitates innovative modeling approaches to bridge the gap between game development and player experience. The challenge lies within the comparison of the model with the experience. This involves two core challenges: (i) accessing the experience by measuring it and (ii) modeling experience as part of our model for the system in a form that allows comparison with the measurements of the experience.

Aspect (i) is an active field of research [8]–[12]. Measuring player experience ranges ”...from more objective approaches such as physiological measurements and in-game behavior analytics to more subjective techniques including interviews, focus groups, in-game probes, and questionnaires” [11, p. 4]. For the second aspect (ii), we need a formal specification for our model that allows us to relate the measurements that indicate the game experience to our model. As part of the model, this planning artifact must be experience-oriented, allow for the creation of the game artifact, and allow developers to enrich the planning artifact with data that matches the format of our measurements for comparison. We believe that both aspects can be solved with formal experience as a foundation for modeling design artifacts.

III. Goal of this Paper

Drawing on our background in game design and engineering, we advocate for a game development process that evolves around the perspective of formal experience. This paper aims to present an approach to designing experiences in a semi-formal way that helps developers align their creations with the intended experiences within an engineering framework. We introduce three critical methodologies: the **Experience Description**, the **Pacing Diagram**, and the **Experience Chart**.

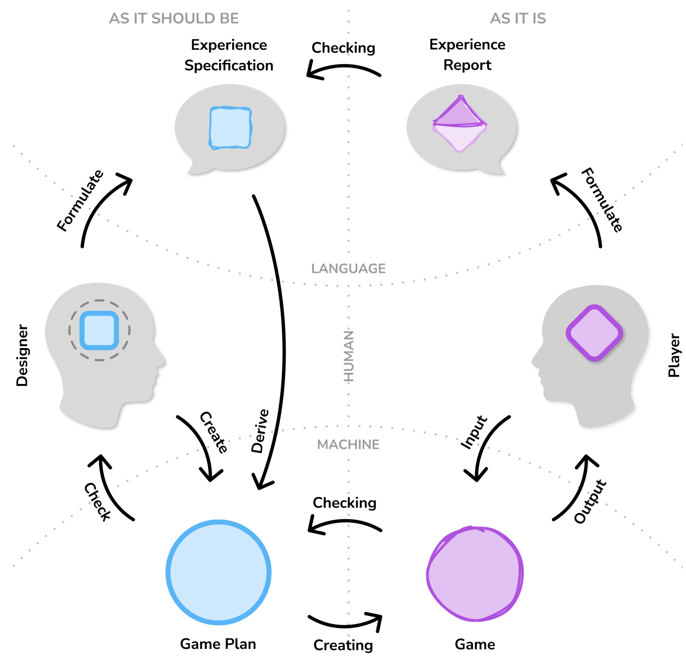

Fig. 1. A model of the game engineering process.

IV. Related Work

Understanding the player's experience by measuring it is a key challenge addressed by researchers and game developers [8]–[12]. In this context, user experience models are created and discussed [11]–[13]. This formal perspective is further supported by literature that looks at games as formal systems [1], [3], [14], [15] and literature that investigate the creation of games within studies on frameworks and methodologies for game design [1], [3], [4]. Of particular relevance is literature from science and game development that focuses on the player experience through the concept of pacing, including literature focusing on pacing as a concept for gameplay [16]–[20] and literature focusing on pacing as a concept for narrative [21]–[25].

V. Game Engineering

Our approach involves comparing the constructed system's experience with a model. Since direct measurement of experience is not possible, we formalize experience indicators within planning artifacts, aiming to bridge the gap between the current system and its model through a systematic engineering process. The proposed game engineering model depicts two parallel systems: 'as-it-should-be' and 'as-it-is.' The 'as-it-is' system consists of the actual game and the player, where an experience emerges from the player's interaction with the game. The as-it-is system can be measured by observing the game artifact and the player or by asking the player how they experience play. Conversely, the 'as-it-should-be' system begins with a designer's vision of the desired experience. This vision is crystallized into an experience specification, which guides the creation of the game plan. This specification mirrors the experience reports as somebody might formulate it after playing the game. Derived from the experience specification, the game model is completed and subsequently transformed into the actual game artifact by implementing the game. This artifact is subject to continuous evaluation against the model to ensure fidelity to the intended design following the engineering procedure. Developers can compare the system with the model by comparing the experience report with the experience specification and measurements of the player and the system with our experience planning artifact. Figure 1 illustrates this model. In the remaining paper, we illustrate the form and origin of these planning artifacts.

VI. The Experience Description

Game designers recognize games as second-order entities, focusing on archetypal play experiences rather than the game artifact itself. Through structured descriptions, designers formalize experience, creating models for the engineering process. Designers specify experience with experience descriptions. These formalizations allow for comparisons between actual play experiences and the intended design. These models vary, including High-Concept Statements or ”30 Seconds of Gameplay,” which highlight player activities and emotional responses, consistently placing the player in an active role. The High Concept Statement or Elevator Pitch is a one-sentence description of the game, focusing on what the player does in the game. ”30 seconds of Gameplay” describes the typical activities a player performs within a game session with the player in the active state. Common to all these approaches is the emphasis on describing experiences from the player’s perspective, using verbs to outline a sequence of player actions and system reactions mediated by input and output devices. This player-centric documentation aids in aligning the development process with the envisioned experience.

VII. A Formal View of Experience

Games are formal systems, precise in their technological execution [14], [15] and governed by formal rules foundational to their design and analysis [3]. Based on this formal nature, we apply a formal perspective on experience. We identify experience as something happening in time, with a start, a duration, and an end. Experience can be understood as a temporal concept: a sequence of activities described by verbs and their corresponding system reaction, as an exchange of action and reaction between the player and the game, translated into the intellectual and emotional mind of the player.

A. The Formal Nature of Play & the Game State

Applying a system theoretical perspective, we can identify the series of activities from which the experience emerges as a series of interactions with the game state: the player (user) is interacting with the game (system) via input and output devices (interfaces). ”When you play a game, you are simply interacting with the game state” [15, ch.”Abstract games ...”]. The essential aspect is the change of state as a consequence of player interaction. We can model a game as

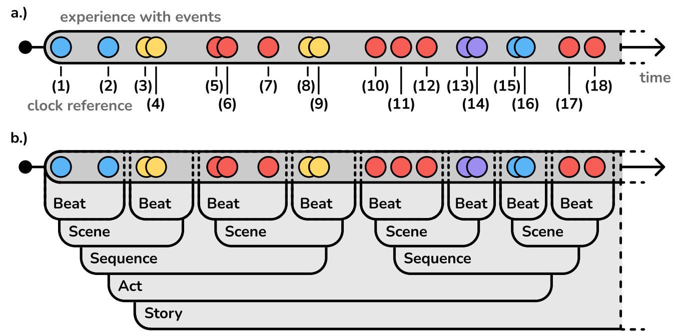

Fig. 2. A formal model of experience.

an abstract control system [14]. We follow the definition by Grünvogel [14, Definition 1] and define a game as follows:

> A game is a triple $(S, \mathcal{M}, F)$, where S is a set, $\mathcal{M}$ is a monoid and F is a (possibly partially defined) action of the monoid $\mathcal{M}$ on the set S, that is, a map $F : S \times \mathcal{M} \to S$ satisfying:
>
> - Identity: $F(s, \epsilon) = s$ for all $s \in S$ and ϵ the neutral element of the monoid $\mathcal{M}$.
> - Semi-group: $F(s, mn) = F(F(s, m), n)$ for all $s \in S$ and $m, n \in \mathcal{M}$. [14, p. 2]

$\mathcal{M}$ represents the input, S represents the state space, and F defines the rules defining the state of all entities in the game under the current state and the input [14]. Playing a game involves repeatedly invoking F with the parameters s and m.

B. Events

We can observe the evolution over time as a sequence of events. In the field of computing, an event refers to any occurrence that is detected and processed by a system. Here, occurrence means change in any quality impacting the system. Events can be triggered by various sources, including the system itself, user actions, or other external factors, and are usually managed or responded to by the system. In our example, these events result from input or internal changes covered by the rules implemented in F. For our case, we can define events as all changes within the state space S,i.e., changes between S_i and S_{i+1}. Here, it is important to note that S can include the player or input if we represent it within the state space (compare [14]). We discretize experience by identifying the key events for the play experience. An event is the atomic, most fundamental building block of the formal structure of the play experience. Atomic means that there is no smaller structural element in our schema to describe the formal structure of experience.

C. Experience as Temporal Concept

Events occur in the context of time. Here, we describe time as the scale of the continuous progress of existence. In the context of a play experience, existence may refer to any game element in S, as perceived **from the perspective of the player**. For the player’s experience, this progress can be perceived as events occurring in a one-way fashion from the past through the present into the future. To get more concrete when dealing with events, we can use a clock as a reference for the continuous progress of time. The structure and the function of this clock should not be of special importance here yet. Physical clocks, as well as logical clocks such as Lamport Clocks [26] or Vector Clocks [27], [28], are conceivable. With such a reference, we can set events in relation to our clock. We can assign a clock value to each event–we can identify an event's past, present, and future. With this, we arrive at our current formal understanding of experience: experience can be formalized as events ordered in time as perceived by the player. The experience is the translation of this sequence of events in time into the emotional and intellectual life of the player. Figure 2-a illustrates the experience with events placed on the arrow of time and a clock value assigned to each event.

With this formal perspective on events, we apply a functional view to experience. It describes *what* is happening and *how* it is *functionally* related with regard to the triple $(S, \mathcal{M}, F)$. Not how it is perceived on an aesthetical level. This model is not the experience. It helps us plan the experience and compare it with measurements, but *it is not the experience*.

D. Abstraction

An effective approach requires the ability to abstract. For this, we can draw inspiration from another field that involves the planning of temporal events: screenwriting for movies.

1) Game Beat: Events are the atomic building blocks of what is **happening**. Usually, events can be related to objects. When these objects interact, they create "an exchange of behavior in action/reaction" [29, p. 37]. From the perspective of the player, these sequences with a common integrity are usually perceived as coherent interactions. We call this a **game beat**. A beat, therefore, consists of a sequence of **events**, as illustrated in Figure 2-b.

2) Experience Structure: In screenwriting, beats are described as the smallest narrative unit [29]. These beats are then used within a hierarchical formalization of the story. Beats that share inherent qualities are further abstracted and combined to create a hierarchical structure to describe an experience. We follow the terminology used for story structure proposed by McKee [29]. A series of beats is a scene. "A scene is an action through conflict in more or less continuous time and space that turns the value-charged condition of a character's life on at least one value with a degree of perceptible significance..." [29, p. 35]. Story values are described as "...the universal qualities of human experience that may shift from positive to negative, or negative to positive, from one moment to the next" [29, p. 34]. A series of scenes forms a sequence of "generally two to five [scenes]–that culminates with greater impact than any previous scene" [29, p. 38]. Further intermediate levels for an effective use are conceivable. An act is a series of sequences "... that peaks in a climactic scene which causes a major reversal of values, more powerful in its impact than any previous sequence or scene" [29, p. 41]. "A story is a series of acts that build to a last act climax or story climax which brings about absolute and irreversible change" [29, p. 42]. Figure 2-b illustrates the experience as events ordered in time and their abstraction through grouping based on integrity.

3) A Discussion of Terminology: Games blend narrative and gameplay, creating procedural narratives even without pre-defined narrative through event sequences. We apply the terminology from screenwriting to games in general, albeit with nuances in usage by level designers who sometimes interchangeably use the term "beat" for what we have called "scenes." We argue for the screenwriting definitions where "beats" signify brief moments, contrasting with level design literature where "beats" may describe longer gameplay interactions lasting over a minute. The argumentation from the screenwriting literature is more conclusive.

E. Pacing

In order to abstract the formal sequence of events, we have to identify the integrity of events–previously described with indicators for human experience such as *story values*, *impact*, and *climax*. Given the event's properties, we can identify the change of identity over time by comparing the properties of events. This process is known as **pacing**.

Games can be understood as a narrative medium and a medium of gameplay. While games embody both aspects, this differentiation is visible in the concept of pacing. Some literature focuses on pacing as a concept for gameplay [16]–[20], while others focus on pacing as a concept for narrative [21]–[25]. We argue that the overall nature of pacing applies to the combination of both. Pacing is influenced by all sorts of components of a game, including music, visuals, narrative, difficulty, and gameplay elements [21].

Pacing describes the rhythm resulting from the recurring patterns of events in time with regard to their intensity and the temporal arrangement. This concept integrates both narrative and gameplay elements, where intensity is derived from player excitement due to conflicts and challenges encountered throughout the game. Gameplay, as well as narrative and aesthetical elements, contribute to this intensity, influencing the player's engagement and emotional response. The overall rhythm of the game is shaped by how the player perceives events, creating a dynamic interplay between the narrative progression and gameplay challenges, providing exciting variations in intensity. "Pacing is based on the concept that in order for gameplay action to seem exciting, it must be contrasted with moments of 'quieter' gameplay..." [4, p. 82].

VIII. The Pacing Diagram

This formal model of an experience is not just a formal artifact for the sake of theoretical discourse. In the context of pacing, we can find this formal perspective in practice: Designers use similar approaches to design and plan experiences in games. A common approach for designers is to use a visual graphic to describe the pacing of a game. While this approach is often used and talked about (compare pacing diagrams from [13], [16]–[18], [21], [25]), it needs to be standardized and formalized. We propose a minimal set as a standard so that the results are comparable and understandable to those

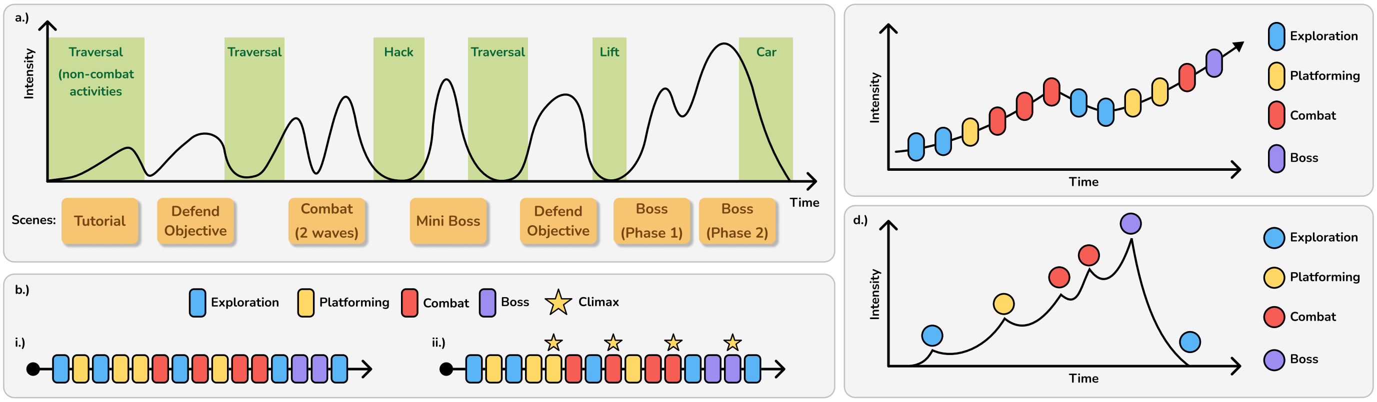

Fig. 3. Four examples of pacing diagrams. Figure a.) shows an example adapted from [30]. Figures c.) and d.) show two adaptions of a pacing diagram based on a tweet by Max Pears (Link: https://twitter.com/MaxPears/status/1161189416225968129). Figure b.) shows a variation of the standard pacing diagram.

familiar with the concept while allowing for a greater variety of expressions. Let us define pacing diagrams as follows.

A. Data Structure

The underlying data is a collection of experience units of an arbitrary level of abstraction. Any abstraction level from events to beats to scenes to the entire experience is conceivable. Each experience unit comes with a minimal set of four properties: a clock value for the start of the unit, an approximated duration, an intensity value, and some form of description of the unit's integrity, for example, with regard to the player's activity. These descriptions might be discrete and precise–represented as a category–or informal and intuitive–written in prose, e.g., in the form of experience descriptions, as introduced in VI. Enriching the units with data about their nature is crucial. Without this information, the data is meaningless. The hierarchical relation between units of different abstraction can either be derived from the clock values or modeled explicitly. Further properties are up to the developer. Technologies that are readable by humans and machines, such as XML or JSON, are conceivable for serialization.

B. Visualization

For the visualization, the experience units are drawn on the plane so that "... the correspondence on the plane can be established among all the elements of one component and all the elements of another component" [31, p. 193]. We model the time on the horizontal axis and the intensity of the game experience on the vertical axis. Experience units are represented by shapes or icons, possibly in combination with text. The icon's form can represent player activity categories if these exist. Unit representations are connected by a line to communicate the order and change of intensity over time. Examples are visualized in diagrams 3-b and 3-d. This is our basic specification for visualization. According to Bertin [31], this usage of the planar dimension of the graphic is the definition of a diagram. Thus, we call this graphic a **Pacing Diagram**. Variations are allowed to achieve a maximum in expression and efficiency.

1) Abstraction: The modeled experience unit can be of any abstraction level, as the designer wishes for communication. If we visualize higher abstractions, their properties are either modeled explicitly or derived from lower-level specifications.

2) Time Scale: We can differentiate between diagrams that model the experience units on the x-axis according to order (all symbols are placed evenly as seen in 3-c) or respecting their specific durations (they are placed on the x-axis according to their starting time as seen in 3-d). Another variation might involve a processed data structure that breaks down the entire sequence in units of the same time length, i.e., one minute. Each new entry receives the properties from their origin, and the new sequence is then visualized as normal.

3) Intensity Variations: Experience units might be projected on the x-axis so that the line indicating the intensity and the symbols are separated (compare figure 3-a.) A further simplification is to omit the intensity axis altogether and only look at the sequence of beat representations on the x-axis, as shown in Figure 3-b. (Local) maxima and minima can be marked with symbols, as shown in Figure 3-b-ii.

4) Annotations: Further annotations, such as clustering units by drawing a line around multiple units or written comments, are conceivable (compare Figure 3-a and 2-b.)

C. The Pacing Diagram and Measurements

A pacing diagram is a tool designers use to plan the pacing of a linear sequence of play activities. Pacing diagrams can be used for small-scale segments, such as levels, and larger segments, such as the entire game. With the pacing diagram, we have a formal description of a linear play experience. This artifact can be used as a reference when creating the game, and we can use it to compare the measured player experience with the original intended experience. By giving measurements a clock reference, we can link measurements to our formal representation within the context of the pacing diagram.

IX. Non-Linear Potential & Reactive Systems

Games distinguish themselves from classical media through their inherent interactivity and non-linear potential. Classical media's narrative structure with events presented in a fixed sequence is linear, and the narrative unfolds in a predetermined order. This sequence is the *plot* [29]. In classical media, the plot is decided by the creator, and the player experiences a static result. In games, the sequence of events is not fixed; it exists within a **possibility space**. This space is characterized by the potential for multiple outcomes at any given point–the potential for non-linear behavior–determined by player input and the game state. Games allow for "...free movement within a more rigid structure" [3, p. 304]. The creator creates the frame that is experienced by the player. Within this frame, the player creates the plot through interaction within the space of possibilities. This freedom is represented within the possibility space, created by the interactive nature of the game and the presence of choice. The boundaries of the rigid structure are formally defined by the specification of the game state. The linear sequence of events, as perceived by the player, can be interpreted as the result of traversing the space of possibility implemented within the game state. According to Nitsche [32], "On the formalist side, temporal structure is implanted in a game through conditional relationship depending on the game state" [32], allowing for a dynamic narrative and gameplay experience that can diverge significantly from linear media.

With the possibility space embodied within the game's behavior, we can identify games as **reactive systems**. A reactive system maintains an ongoing relationship with its environment rather than merely processing data in isolation [33]. These systems are characterized by a behavior model that defines valid sequences of input and output events, conditions, timing constraints, and actions [34]. This clear and rigorous behavioral description of what the system should do is essential for a reactive system [34]. For games, this behavior model essentially defines the game state. The form of this specification can be further concretized with "... a cue from computer science, saying that a game is actually a state machine: it is a system that can be in different states; it contains input and output functions, and definitions of what state and what input will lead to what following state" [15, ch."Abstract games ..."]. The use of state and event-based approaches for system design is widely recommended in the literature, presenting a natural method for capturing dynamic behaviors through abstraction in system design [34]. State machines are composed of a fixed set of states and a set of transitions between these states. They are depicted as directed graphs, with nodes representing states and arrows indicating transitions, typically labeled with the triggering events and conditions [34]. State machines are often visualized as state transition diagrams.

X. The Experience Chart

Harel's work [34], [35] underscores the applicability of state machines for modeling reactive systems, highlighting the potential of games to be conceptualized within this framework. By leveraging Statecharts, game developers can formally approach games' design and interaction complexity, mapping out the expansive possibility space that defines the play experience. We propose experience charts based on Statecharts [34] as a solution approach for the modeling of the non-linear potential of games. In its pure form, the experience chart is a set of nodes modeling experience units following the explanation for pacing diagrams and a set of relations between these experience units modeling possible transitions between experience units. The form of the new data structure follows the Statecharts specification. Statecharts offer an advanced framework for detailing the behavior of complex reactive systems [34]. They extend traditional state transition diagrams by introducing hierarchical structuring, differentiation between high-level and low-level events, and support for multi-level concurrency [34]. In [34], Harel introduces the visual formalism, which represents the heart of Statecharts. In [34] and later in [35], formal syntax and semantics are discussed.

With the non-linear potential between various experience units modeled as a graph, we model the topology of play. We can establish the correspondence on the plane among all components. Following the definition by Bertin [31], our graphic is a network as the plane does not have any meaning, and the information lies within the relations between the elements. An efficient arrangement of these elements in the plane produces a minimum number of intersections [31]. The form follows the Statechart specification. We call this graphic **Experience Chart**. Figure 4 illustrates an example.

XI. The Overall Picture

Experience charts and pacing diagrams have a dynamic, interdependent relationship. Pacing diagrams represent linear event sequences with intensity and time (i.e., a specific playthrough), serving as interpretations of the non-linear possibility space detailed in experience charts. Conversely, experience charts can be developed from pacing diagrams by forming a state machine capable of reproducing these sequences. This mutual relationship enhances understanding and planning of game experiences.

A. Planning Experience

The integration of experience descriptions, pacing diagrams, and experience charts with our game engineering model presented in V facilitates a nuanced game engineering process. Initially, game descriptions guide the creation of pacing diagrams and experience charts, enabling designers to visualize intended playthroughs and possible experiences. These artifacts are enhanced with attributes indicating the play experience, like intensity and duration, and linked with additional resources (e.g., mood boards and asset lists) to form a comprehensive planning model. During gameplay, we can create pacing diagrams that track events and interactions, incorporating observational data such as player inputs and biometric measurements to enrich our understanding of the player experience. (Compare the approaches presented in [8]–[11].) This approach allows for comparison of real gameplay and the corresponding experience against the planned model, assessing if actual play aligns with expected outcomes and ensuring the development remains true to the intended model.

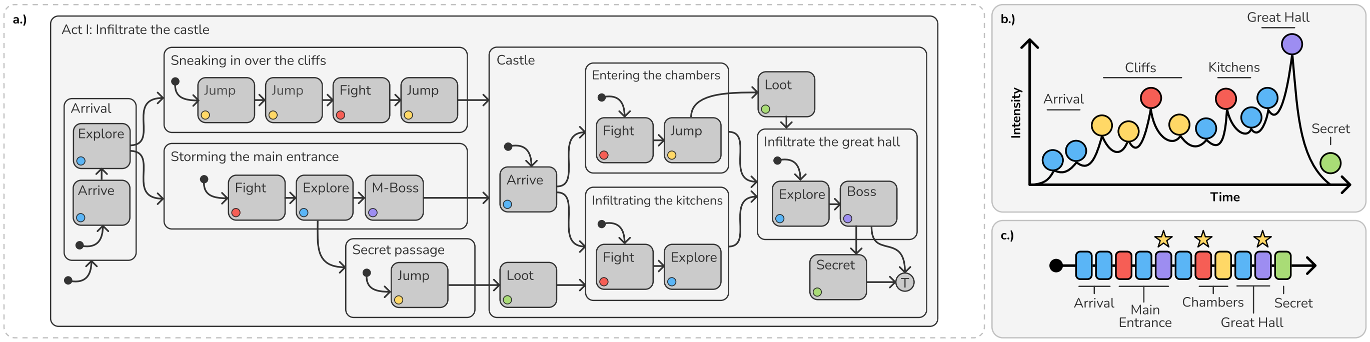

Fig. 4. a.) An experience chart for an example scenario of an adventure game with beats described by verbs as atomic building blocks. All transitions describe traversal without additional conditions. The labels are, therefore, omitted. Diagrams b.) and c.) show derived pacing diagrams of different paths from the experience chart with annotations.

This methodology offers a game engineering framework for bridging the game model and player experience.

B. Opportunities

We see many opportunities within the presented approach:

1) Experience Charts as System Specification: The close connection between the experience chart and game state suggests the chart's potential as a system specification, similar to the idea behind Statecharts [34].

2) Possibility as Game Feature: Formalizing the possibility space within the experience chart for runtime access enables adaptive game features based on the player's past, present, and potential future experiences. Once formalized, the possibility space can be used for the implementation of game features.

3) Shared Language: Game development is an interdisciplinary field that involves various domains and professions [1], [36]. It encapsulates elements from the visual arts, music, sound design, narrative creation, writing, product design, programming, mathematics, engineering, psychology, health care, architecture, business, law, ethics, and economics, among many others [1], [4], [36]. Given game development's interdisciplinary nature, the presented artifacts can unify various domains, serving as a central planning artifact. This facilitates a standardized approach to experience planning, improving adoption, consistency, and comparability across projects.

XII. Discussion

In our discussion, we focus on three aspects: Intuition in game development, the subjectiveness of play, and the spatial quality of play experience.

1) Intuition in Game Development: Our approach is based on the perspective of games as closed formal systems. This contrasts the imprecision inherent in games' relation to human experience. While games are precise as isolated artifacts, their design process–rooted in creativity and human interaction–often operates within the realms of intuition and imprecision. Formalism demands precision, but design thrives on flexibility and the freedom to be imprecise. This dichotomy is crucial during the interaction and interpretation of play, where the intuitive nature of the game becomes evident. The engineering process of games must accommodate both the formal, precise nature of games and the intuitive, imprecise aspects of human experience. We argue that our formal approach allows for the integration of intuitive artifacts within the development process. Using experience charts and pacing diagrams enriches the intuitive, creative process and helps to formalize it, ultimately providing a balance between both worlds.

2) Subjectiveness of Play: The subjective nature of play challenges objective measurement from an external standpoint. It seems like authentic insights into subjective experiences can only come directly from players, e.g., by comparing two experiences within one's own emotional world. Glimpses of this approach can be incorporated into the development process with artifacts such as mood boards and game references. For now, we want to leave this discussion for the field of measuring player experience. We argue that measuring experience is a valid approach to evaluate play for the engineering process. In our model for the game engineering process, we used mental models to explain the indirect nature of the experience. Mental models are discussed lively in literature. To accommodate the scope of this paper, we want to refrain from entering this discussion. Instead, we refer to the large body of research on mental models (compare [5], [6]). Staggers and Norcio [37] explore various questions around mental models as concepts for human-computer interaction research, including a comprehensive overview of conceptual and theoretical issues. Nevertheless, the theory of mental models helps us to deepen our understanding of experience within an engineering framework.

Spatial Quality of Experience: The presented methods are based on the arrangement of events in time that depends on the possibility space. For games, this implies both temporal and spatial relationships. Nitsche [32] conjectured that from a player-centered perspective, time and space in video games are closely interdependent in that traversing space is understood as progressing time, i.e., progressing *event space–time*. This perspective is supported by literature that researches player experience. Maram et al. [12] investigate player strategies and decision patterns based on spatial abstraction. The spatial quality of experience is also visible in artifacts that resemble spatial variations of the pacing diagram. The Nintendo Power Method is a design tool inspired by maps from the magazine

"Nintendo Power" published by Nintendo in the late 1980s. "Nintendo Power maps show a level from a macro-scaled point of view so that a reader can see a game level in its entirety. At the same time, it highlights important gameplay moments on the micro-scale so readers know what to look out for to reach secret rooms or beat difficult obstacles" [4, p. 84]. Used as a design tool, the spatial distribution allows conclusions about the density, and thus pacing, of the level. A similar relationship exists in the resemblance of form between experience charts and molecule diagrams as introduced by Azar [38]. Azar advocates for a topological approach to level design and presents molecule design based on a graph-based approach to the design of a game's space. This indicates a direct relation between space and our approach that is not yet fully understood and integrated into the concept.

XIII. Conclusion & Future Work

In this paper, we have presented a model for game engineering and identified a core challenge for the engineering approach for games. In this context, we presented three solution approaches: the experience description, the pacing diagram, and the experience chart.

For our next steps, we aim to engage both game developers and academics in discussions about our proposed approach, seeking to validate and refine our methods through application in real-world projects. On a theoretical level, we plan to delve deeper into the spatial qualities of the topic to further understand its implications and applications in game development.

References

[1] J. Schell, *The Art of Game Design: A book of lenses*, 3rd ed. CRC press, 2019.

[2] E. Weinberger, *Nineteen ways of looking at Wang Wei*, O. Paz, Ed. New York: Moyer Bell, 1987.

[3] K. Salen and E. Zimmerman, *Rules of play: Game design fundamentals*. Cambridge, Mass. [u.a.]: MIT Press, 2004.

[4] C. W. Totten, *An architectural approach to level design*. Boca Raton, Florida: CRC Press, 2014.

[5] P. N. Johnson-Laird, *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press, 1983, no. 6.

[6] J. K. Doyle and D. N. Ford, "Mental models concepts for system dynamics research," *System dynamics review: the journal of the System Dynamics Society*, vol. 14, no. 1, pp. 3–29, 1998.

[7] J. Fisher, D. Harel, and T. A. Henzinger, "Biology as reactivity," *Communications of the ACM*, vol. 54, no. 10, pp. 72–82, 2011.

[8] P. Cairns, A. Cox, N. Berthouze, S. Dhoparee, and C. Jennett, "Quantifying the experience of immersion in games," in *Cognitive Science of Games and Gameplay workshop at Cognitive Science*, 2006, pp. 26–29.

[9] A. Drachen and A. Canossa, "Towards gameplay analysis via gameplay metrics," in *Proceedings of the 13th international MindTrek conference: Everyday life in the ubiquitous era*, 2009, pp. 202–209.

[10] L. Nacke and A. Drachen, "Towards a framework of player experience research," in *Proceedings of the second international workshop on evaluating player experience in games at FDG*, vol. 11, 2011.

[11] D. Johnson, M. J. Gardner, and R. Perry, "Validation of two game experience scales: the player experience of need satisfaction (pens) and game experience questionnaire (geq)," *International Journal of Human-Computer Studies*, vol. 118, pp. 38–46, 2018.

[12] S. S. Maram, J. Pfau, J. Villareale, Z. Teng, J. Zhu, and M. S. El-Nasr, "Mining player behavior patterns from domain-based spatial abstraction in games," in *2023 IEEE Conference on Games (CoG)*, 2023, pp. 1–8.

[13] F. Coulianos, "Pacing and gameplay analysis in theory and practice," 2011. [Online]. Available: https://www.gamedeveloper.com/design/pacing-and-gameplay-analysis-in-theory-and-practice

[14] S. M. Grünvogel, "Formal models and game design," *Game Studies*, vol. 5, no. 1, pp. 1–9, 2005.

[15] J. Juul, "Introduction to game time," in *First Person: New Media as Story, Performance, and Game*, N. Wardrip-Fruin and P. Harrigan, Eds. Cambridge, Massachusetts: MIT Press, 2004, pp. 131–142.

[16] A. Bagus Harisa and W.-K. Tai, "Pacing-based procedural dungeon level generation: Alternating level creation to meet designer's expectations," *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 401–416, 2022.

[17] T. Hoffstetter, "The art and science of pacing and sequencing combat encounters," 2013. [Online]. Available: https://www.gamedeveloper.com/design/the-art-and-science-of-pacing-and-sequencing-combat-encounters

[18] C. Linehan, G. Bellord, B. Kirman, Z. H. Morford, and B. Roche, "Learning curves: analysing pace and challenge in four successful puzzle games," in *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*, 2014, pp. 181–190.

[19] M. Venturelli and K. Studio, "Space of possibility and pacing in casual game design–a popcap case study," in *VII Brazilian Symposium on Games and Digital Entertainment*, 2009.

[20] M. Davies, "Examining game pace: How single-player levels tick," 2009. [Online]. Available: https://www.gamedeveloper.com/design/examining-game-pace-how-single-player-levels-tick

[21] S. Costiuc, "Charting through pacing of uncharted 2. part 1 - pacing principles," 2019. [Online]. Available: https://www.gamedeveloper.com/design/charting-through-pacing-of-uncharted-2-part-1---pacing-principles

[22] K. Gyllenback, "Putting into play," 2020. [Online]. Available: https://katarinagyllenback.com/

[23] E. Hill, "Tick, tick, boom!! - catharsis and pacing in games: The new dramaturgy pt.2," 2013. [Online]. Available: https://www.gamedeveloper.com/design/tick-tick-boom---catharsis-and-pacing-in-games-the-new-dramaturgy-pt-2

[24] V. Ho, "Game pacing: Differences between mediums," 2017. [Online]. Available: https://www.gamedeveloper.com/design/game-pacing-differences-between-mediums

[25] M. Lopez, "Gameplay fundamentals revisited: Harnessed pacing & intensity," 2008. [Online]. Available: https://www.gamedeveloper.com/design/gameplay-fundamentals-revisited-harnessed-pacing-intensity

[26] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," in *Communications of the ACM, Vol.21 (7)*, 1978, pp. 558–565.

[27] C. J. Fidge, "Timestamps in message-passing systems that preserve the partial ordering," in *Australian Computer Science Communications*, vol. 10, no. 1, 1988, pp. 56–66.

[28] F. Mattern, "Virtual time and global states of distributed systems," in *Proceedings of the Workshop on Parallel and Distributed Algorithms*, 1988, pp. 215–226.

[29] R. McKee, *Story: style, structure, substance, and the principles of screenwriting*. Harper Collins, 1997.

[30] P. Ellis, "Single player level design pacing and gameplay beats (part 1-3)," 2015. [Online]. Available: https://www.worldofleveldesign.com/categories/wold-members-tutorials/peteellis/level-design-pacing-gameplay-beats-part1.php.

[31] J. Bertin, *Semiology of graphics*. University of Wisconsin press, 1983.

[32] M. Nitsche, "Mapping time in video games." in *DiGRA Conference*, 2007.

[33] D. Harel and A. Pnueli, "On the development of reactive systems," in *Logics and models of concurrent systems*. Springer, 1984, pp. 477–498.

[34] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of computer programming*, vol. 8, no. 3, pp. 231–274, 1987.

[35] D. Harel and A. Naamad, "The statemate semantics of statecharts," *ACM Transactions on Software Engineering and Methodology*, vol. 5, no. 4, pp. 293–333, 1996.

[36] IGDA Education Special Interest Group, "Igda curriculum framework," International Game Developers Association (IGDA), Tech. Rep. v3.2beta, 2008.

[37] N. Staggers and A. Norcio, "Mental models: concepts for human-computer interaction research," *International Journal of Man-Machine Studies*, vol. 38, no. 4, pp. 587–605, 1993.

[38] N. Azar, "The metrics of space: Molecule design," 2013. [Online]. Available: https://www.gamedeveloper.com/design/the-metrics-of-space-molecule-design