VIOLETA Framework: Enhancing Real-World Skill Mastery Through Educational Games

Kristijan Đokic

August 3, 2025

Contents

Chapter 1

Introduction

1.1 Background and Motivation

Throughout my youth, I engaged extensively with video games, often spending more than two hours per day playing. In retrospect, I find myself questioning who I might have become, and what skills I might have developed, had I invested that time differently. While I do not consider playing video games a mistake—indeed, I attribute much of my cognitive development to the complex puzzles and strategic challenges I encountered—the benefits of gaming appeared to diminish beyond a certain point. In my view, pursuing a broader range of activities would likely have been more advantageous.

This reflection prompted a central question: why are many productive activities less engaging than games, and could games be designed to foster real-world skills while maintaining their inherent appeal? From this question emerged a fundamental idea: games should not only entertain but also ensure that the more an individual plays, the more they master real-life competencies. The possibility of learning complex subjects, such as quantum physics, with the same enthusiasm typically reserved for casual games among friends, became a driving motivation. This vision led to the creation of a dedicated framework, named VIOLETA (Versatile Interactive Ontology-based Learning Environment for Teaching & Assessment), which is built entirely around this principle—that gameplay and real-world skill mastery must be directly and systematically linked.

1.2 Problem Statement

1.2.1 Context

Automation is progressively transferring simpler work tasks to machines (Bughin et al., 2018). As a result, a growing proportion of the population will need to engage in more complex, cognitively demanding jobs. Continuous learning will be essential to adapt to these changes. If learning could be made faster and more intuitive, it would significantly improve employee training, accelerate research activities, and strengthen overall societal innovation capacity.

1.2.2 Missing Links Between Pedagogy and Play

Although several educational game design frameworks have been proposed—such as the Game Object Model (GOM) (Amory, 2007), (Hart et al., 2021), Universal Design for Learning (UDL) (CAST, 2018), and Let's Get On-Board (Nautiyal et al., 2024)—many fail to clearly illustrate how learning elements and gameplay mechanics influence each other. For example, UDL provides valuable guidelines for making learning accessible to all students by offering multiple means of engagement, representation, and expression. However, it offers limited guidance on how to translate these pedagogical strategies into interactive gameplay mechanics. Similarly, the Game Object Model emphasizes the alignment of narrative, gameplay, and learning objectives but remains abstract in its implementation, often lacking specific, actionable methods for integrating complex educational content into game systems.

Instead, many of these frameworks provide vague recommendations such as "incorporating learning material into gameplay" (Nautiyal et al., 2024) without detailing operational strategies—that is, step-by-step processes, decision-making tools, or mapping techniques that would help designers systematically link gameplay actions to educational outcomes. In practice, this means that the connection between what the player does in the game and what they are supposed to learn often remains implicit or incidental. Designers are left without structured guidance on how to build this connection or evaluate whether it exists.

Additionally, some frameworks restrict creative freedom by prescribing a limited set of predefined mechanics or emotional outcomes (Xin, 2022)—for example, assuming designers must work within fixed categories like the six core emotions in Dillon's 6-11 Framework. While such models aim to guide emotional design, they can unintentionally constrain creativity by discouraging designers from defining their own experiential goals. A more flexible approach would allow designers to express their intended aesthetics in just two or three of their own words—such as calmness, urgency, or resilience—and then design mechanics that support those experiences. Meanwhile, other frameworks are overly complex (Spieler et al., 2017) or narrowly focused (Silva, 2020), introducing gaps and blind spots based on the authors' limited knowledge of game mechanics.

Another critical issue is the insufficient accommodation of diverse learner knowledge levels; many models assume a homogenous learner profile, which is unrealistic in real-world educational contexts. Furthermore, research shows that serious games often fail to achieve pedagogical objectives and struggle to sustain player engagement when the gaming and learning aspects are not properly integrated (Jaccard et al., 2021).

1.2.3 Consequences

Without a structured, well-grounded approach to educational game design, developing effective educational games becomes both resource-intensive and risky. A structured framework could significantly reduce the time and effort required, while increasing the likelihood of success by embedding best practices into the development process. For example, instead of relying on a conventional, teacher-centered format—where the instructor explains merge sort through a one-way presentation with minimal student interaction—a teacher could design a reusable educational game that actively engages students in exploring the sorting algorithm themselves. This game-based approach could not only increase engagement but also improve long-term retention by combining entertainment with instruction (Jaccard et al., 2021).

1.2.4 Specific Need

There is a need for a framework that systematically links game mechanics with educational knowledge, bridges varying levels of learner difficulty, and cultivates intrinsic player motivation to maintain engagement over time. As discussed in the previous sections, existing frameworks often fail to provide operational strategies for aligning gameplay with learning outcomes, constrain creative freedom through rigid design assumptions, and overlook the diversity of learner profiles. Empirical studies reinforce these shortcomings: Annetta argues that a serious game "must first be fun, or the player will not want to play it," showing how inadequate attention to enjoyment erodes sustained participation (Annetta, 2010). Likewise, Silva concludes that an educational game "must be catchy for the player to want to play it multiple times," yet most prototypes he reviewed fall short, leading to limited replayability and shallow engagement (Silva, 2020). Collectively, these limitations contribute to the high development costs, inconsistent learning outcomes, and low player engagement observed in many serious games. Without a structured approach that directly addresses these gaps, educational game design remains inefficient, inaccessible to many educators, and unlikely to consistently produce meaningful learning experiences.

A new framework must not only offer clarity and flexibility, but also empower designers to build experiences where deeper gameplay naturally leads to deeper understanding. This need extends far beyond individual classrooms or training sessions. In today's world—characterized by rapid technological change, global interconnectivity, and growing cognitive demands on workers and citizens alike—learning is no longer a phase of life, but a permanent, dynamic process. As highlighted in a McKinsey report (Bughin et al., 2018) on the future of work, the demand for cognitive skills and higher-order thinking is expected to rise significantly in response to increasing automation and workplace complexity. In my view, societies that invest in more effective learning systems—particularly those that make learning intuitive, engaging, and lifelong—will be better equipped to meet these challenges with adaptability, innovation, and resilience.

Educational games, if properly designed, can play a transformative role in this process. By embedding real-world competencies into intrinsically motivating environments, they can help individuals not only learn faster, but develop the kind of flexible, systems-level thinking needed in modern life. The societal impact of such a shift would be profound: improving workforce readiness, enabling more equitable access to quality learning, and ultimately fostering a population better prepared to navigate and shape the world we are building.

1.3 Objectives of the Thesis

The main objective of this thesis is to develop and present the VIOLETA framework, a structured methodology for designing educational games where increased gameplay leads directly to the mastery of real-world competencies.

The specific objectives are as follows:

- To analyze existing educational game design frameworks and identify their key limitations regarding the integration of gameplay mechanics and knowledge development.
- To define the principles, components, and construction methodology of the VIOLETA framework.
- To illustrate how the VIOLETA framework ensures systematic linking between gameplay mechanics, learner progression, and real-world skill acquisition.
- To critically evaluate the strengths and limitations of the VIOLETA framework in comparison to existing models.
- To reflect on broader implications for educational game design and propose future directions for refining the framework.

1.4 Structure of the Thesis

This thesis is organized into eight main chapters:

- **Chapter 1: Introduction**
  This chapter provides the background and motivation for the research, formulates the problem statement, defines the objectives of the thesis, and outlines its overall structure.
- **Chapter 2: Theoretical Background**
  This chapter presents the psychological and theoretical foundations of the VIOLETA Framework, including Self-Determination Theory, Cognitive Load Theory, and insights from evolutionary learning theories such as those discussed in *The Play of Animals*. It examines how intrinsic motivation, cognitive load management, and intuitive mapping of complex topics to gameplay are leveraged to design educational games that systematically foster real-world competencies.
- **Chapter 3: The VIOLETA Framework**
  This chapter introduces the motivation behind the creation of VIOLETA, outlines its core principles and philosophy, and describes its key components, such as game objective setting, real-life skill mapping, feedback loops, and mechanisms ensuring that increased gameplay leads to increased mastery.

- **Chapter 4: Evaluation of the Framework**
  This chapter provides a logical evaluation of the VIOLETA framework, discusses its strengths and unique contributions, and highlights potential limitations and risks.
- **Chapter 5: Discussion**
  This chapter reflects on broader aspects of educational game design, explores possibilities for improving the VIOLETA framework, and suggests areas where the framework could be expanded or adapted.
- **Chapter 6: Conclusion**
  This chapter summarizes the main contributions of the thesis, offers closing thoughts on the relationship between games and real-life skill development, and suggests directions for future research.
- **Chapter 7: References**
  This section contains all cited books, articles, research papers, and frameworks used throughout the thesis.
- **Chapter 8: Appendices**
  The appendices include optional supplementary materials such as diagrams of VIOLETA components and examples of skill mapping.

Chapter 2

Theoretical Background

VIOLETA Framework is based on Self-Determination Theory, Cognitive Load Theory, lessons taken from *"The Play of Animals"*, and my own rationale.

2.1 Intrinsic Motivation and Play

As Self-Determination Theory states, humans perform best when there is intrinsic motivation behind their actions. To explain why so many people find games intrinsically motivating, we can take lessons from *"The Play of Animals"*. "Animals must practice their instincts during a safe time (youth) to ensure that, when real challenges come, they can survive. Play is thus a tool of evolution to prepare animals for the tasks of adult life." (Groos, 1898) Given this, we can reason that humans intrinsically enjoy learning through play as well, because we have evolved to do so.

Given this fact, the question arises: why didn't we implement more play in our education? The problem of introducing play into learning is the complexity of the subjects involved. While wolf puppies instinctively engage in rough play to practice their muscles and instincts (Groos, 1898), when task complexity rises, there isn't always an intuitive way to introduce play. While motoric skills can be easily practiced through sports like basketball, teaching quantum physics using only a ball and one's body remains a difficult task. Consequently, complex topics have traditionally relied on a conventional, teacher-centered format.

2.2 Games as Educational Tools

The VIOLETA Framework introduces structured methods for integrating game-based activities into teaching, thereby leveraging the evolutionary intrinsic motivation linked to play. As games can create their own worlds, the complexity of topics can be mapped onto knowledge already understood by students, effectively reducing cognitive load.

For example, in the board game *"Power Grid"* (BoardGameGeek contributors, 2004) players must build factories to sell electricity. Various factory types operate on different energy sources (renewable, coal, oil, nuclear, gas). The finite supply of these resources (excluding renewables) affects pricing dynamically. Thus, if a player observes that many coal-based factories exist, they can predict a rise in coal prices and adjust their strategy accordingly. Here, the basics of economy, supply, and demand are intuitively mapped onto game mechanics. Mastery of the board game inherently demands mastery of fundamental economic principles.

2.3 Psychological Needs in Games

This mapping of complex principles into intuitive gameplay leads naturally into an analysis through the lens of Self-Determination Theory. According to SDT (Ryan & Deci, 2000), intrinsic motivation is high when three psychological needs are fulfilled: competence, autonomy, and relatedness.

2.3.1 Competence

Competence is a need that games generally support well due to their nature of offering instant feedback to players' actions. *"Power Grid"* fosters a strong sense of competence by providing players with immediate and intuitive feedback on the consequences of their decisions. As players make strategic choices regarding resource management, factory investments, and supply-demand balancing, they experience visible progress and setbacks, clearly observable through fluctuations in their earnings, which directly reflect the effectiveness of their strategic decisions. The clear mapping between strategic thinking and in-game success satisfies the psychological need for competence, as described by Self-Determination Theory.

2.3.2 Autonomy

Autonomy is another psychological need that games are well suited to fulfill. Many games support autonomy by setting goals but allowing players the freedom to decide how to achieve them. This naturally leads to player experimentation and testing in safe environments, akin to wolf puppies playing with each other. In *"Power Grid"*, players decide when to buy a factory, which type to choose, where to build it, how much money to spend, and how to manage resources. The wide range of meaningful choices supports the autonomy need.

Even in more constrained games like puzzles, autonomy is preserved through strategic decision-making approaches, albeit in a narrower form.

2.3.3 Relatedness

Relatedness, the third psychological need, is also fulfilled in *"Power Grid"*. Being a board game, it is typically played in person, usually among friends or family, which fosters social connections. Other games, such as massively multiplayer games like World of Warcraft, also support relatedness through features like guilds or factions (Przybylski et al., 2010).

However, not all games necessarily fulfill this need. While the VIOLETA Framework does not directly address relatedness, it can indirectly support it. As stated in Self-Determination Theory, "many intrinsically motivated behaviors are happily performed in isolation, suggesting that proximal relational supports may not be necessary for intrinsic motivation, but a secure relational base does seem to be important for the expression of intrinsic motivation." (Ryan & Deci, 2000)

In the classroom, the instructor can use the time freed up by game-based learning to engage more deeply with students by creating a two-way dialogue. Instead of solely delivering information, the teacher can listen to students' experiences, encourage them to share how they perceived and approached the learning challenges, discuss their different strategies and outcomes, and jointly reflect on lessons learned. This transforms the classroom into a dynamic environment where learning is not a one-way transfer of information, but a shared exploration, strengthening both autonomy and relatedness.

This enhances autonomy, as "teachers who support students' autonomy begin by attempting to understand, acknowledge, and where possible, be responsive to students' perspectives" (Ryan & Deci, 2000), and it indirectly supports relatedness by strengthening the teacher-student relationship.

2.4 Managing Cognitive Load

Given that optimal cognitive engagement is crucial to maintaining competence and preventing frustration, the VIOLETA Framework incorporates a dynamic system for monitoring and adjusting cognitive load: the STACK process.

Cognitive Load Theory (Sweller et al., 1998) distinguishes between two memory systems—working memory and long-term memory—and introduces the concept of schemas. "A schema can be anything that has been learned and is treated as a single entity. If the learning process has occurred over a long period of time, the schema may incorporate a huge amount of information." (Sweller et al., 1998) For example, a schema for a restaurant includes extensive knowledge about food, money, architecture, furniture, and related processes.

Cognitive Load Theory argues that working memory can handle approximately seven elements (schemas) at a time; thus, learners who have not yet internalized schemas in long-term memory are at risk of cognitive overload. The STACK process quantifies the number of elements a player is managing at any given time and provides instructors with this information. This allows instructors to adjust the number of elements presented during each STACK phase, ensuring that games dynamically adapt to provide optimal challenges tailored to individual players. This careful management of cognitive load directly supports the competence need.

2.5 Novelty and Motivation

Although the detailed design of the thematic (novelty) component falls outside the scope of this bachelor thesis, its conceptual inclusion is important for illustrating the full motivational architecture envisioned by the VIOLETA Framework. The appeal of novelty, as suggested by Self-Determination Theory, can also bring intrinsic motivation, and some aspects of the framework will briefly touch upon it.

Chapter 3

The VIOLETA Framework

3.1 A Note on Motivation

The motivation behind the creation of the VIOLETA framework has already been discussed in detail in Chapter 1. There, I outlined the personal reflections and systemic challenges that led to the core question driving this thesis: Can we design games that not only entertain but also reliably foster real-world skill mastery?

Rather than repeating those points here, this chapter focuses on the structural and operational realization of that vision. The following sections present the core philosophy of the VIOLETA framework, its foundational components, and the principles that guide its application. Through this exploration, I aim to show how VIOLETA provides a practical yet flexible methodology for educational game designers—one that ensures gameplay is meaningfully and systematically linked to real-life learning outcomes.

3.2 Core Principles and Philosophy

The framework is grounded in five core principles:

- **Mapping gameplay mechanics to real-world skills:** : In VIOLETA, every meaningful player action should support the mastery of a bundled set of real-world skills or knowledge. This is achieved by directly integrating the target learning content into the design of core game mechanics—not merely overlaying educational material on top of gameplay.

- **Adaptable difficulty:** Recognizing that learners have diverse backgrounds and progress at different rates, VIOLETA supports scalable cognitive challenge. Game systems built under this framework are expected to adapt to the learner's current level of mastery, ensuring optimal engagement and avoiding both boredom and overload.

- **Intrinsic motivation as a design priority:** A core tenet of VIOLETA is that educational games must be intrinsically motivating—that is, enjoyable and engaging on their own terms, not because they promise external rewards like grades or approval. Drawing from Self-Determination Theory and evolutionary perspectives on play, the framework recognizes that humans are naturally driven to explore, master, and play when their psychological needs—competence, autonomy, and relatedness—are met. VIOLETA encourages designers to create systems that tap into this intrinsic drive by ensuring that challenges are meaningful, progress is visible, and player agency is respected. When learning feels like play, players are more likely to stay engaged, take creative risks, and sustain effort over time—factors that are essential for deep, lasting skill development.

- **Emotionally aligned learning arcs**: VIOLETA recognizes that durable learning is inseparable from feeling. Every major and micro loop of play should guide the learner through a purposeful emotional progression that mirrors the psychological journey required for the target real-world competency. Designers therefore articulate a desired macro-emotional arc early in the process (see Step 4: Define the Intended Emotional Experience and the subsequent LF/FAG stages) and bind moment-to-moment feedback to that arc. When the emotions evoked by play scaffold the same schemas that the mechanics train, intrinsic motivation deepens, memory consolidates, and transfer to real life accelerates. In VIOLETA, emotion is not decorative; it is the affective glue that makes "more play = more mastery" possible.

- **Trust in human intuition:** Rather than prescribing fixed aesthetics, mechanics, or themes, VIOLETA leaves space for the designer's creativity and personal vision. It acknowledges that meaningful games often emerge from the designer's unique perspective and emotional intent. Therefore, the framework encourages designers to express their learning goals, desired emotional experiences, and thematic choices in their own terms, while still offering a structured path to systematically link those elements to player progression and mastery.

In essence, VIOLETA is both operational and philosophical. It offers clear steps for novice designers while remaining abstract enough to respect the inherently artistic nature of game creation. The goal is not to control design outcomes, but to scaffold a process that ensures the resulting games are intrinsically motivating, educationally effective, and tailored to their intended context.

3.3 The Three Pillars of VIOLETA

The VIOLETA framework is built upon three foundational components—**Atomic Unit**, **Theme**, and **Mechanics**—each representing a crucial dimension of the game design process. These pillars serve as the structural anchors that guide designers in aligning gameplay with learning objectives while preserving flexibility and creativity.

3.3.1 Atomic Unit

The atomic unit represents a bundled set of real-world knowledge, skills, or behaviors that the designer decides should be learned as a cohesive whole. In traditional educational terms, this is roughly equivalent to a "learning objective," but VIOLETA emphasizes **grouping related skills or concepts that reinforce one another**. For example, negotiation skills, emotional regulation, and active listening could form a single atomic unit in a game about conflict resolution.

Rather than treating each individual fact or skill in isolation, VIOLETA encourages designers to think holistically—identifying natural clusters of competencies that make sense to learn together, and designing gameplay that reflects this cohesion.

3.3.2 Theme

The theme defines the narrative context, setting, and emotional atmosphere of the game. It can be realistic (e.g., running a restaurant), fictional (e.g., pirates exploring unknown islands), or fantastical (e.g., time travelers solving paradoxes). More than aesthetic decoration, the theme also supports the learning process by:

- Making abstract skills concrete and relatable through metaphor.
- Triggering curiosity and emotional engagement.

- Providing continuity and coherence across different game scenarios.

Additionally, the theme encourages designers to consider the emotional journey they want the player to experience—similar to how the MDA framework (Xin, 2022) uses "aesthetics"—and to craft situations that evoke feelings such as urgency, calmness, or resilience, depending on the learning goals.

3.3.3 Mechanics

Mechanics are the core systems and rules that govern gameplay. They define how players interact with the game world, make decisions, overcome challenges, and receive feedback. VIOLETA treats mechanics **not only as sources of engagement, but as the primary vehicle through which real-life competencies are practiced and internalized.**

For instance, if the atomic unit involves teamwork and resource sharing, then game mechanics might include cooperative problem-solving, pooled inventories, or communication constraints. The goal is to ensure that mastering the game naturally requires practicing the desired skills.

3.4 Integration Between Pillars

While each of the three pillars—Atomic Unit, Theme, and Mechanics—serves a distinct purpose, the strength of the VIOLETA framework lies in how these elements are meaningfully **integrated**. It is through thoughtful integration that gameplay becomes more than entertainment—it becomes an engine for real-life skill development. Below are the key combinations and their pedagogical significance:

3.4.1 Atomic Unit and Mechanics: Skill-Mapped Gameplay

The most essential integration is between the **Atomic Unit** and **Mechanics**. In this connection, game mechanics are deliberately designed so **that success in the game requires the same competencies** the player is meant to develop in real life. For example, if the atomic unit includes time management and prioritization, a game might include scheduling dilemmas or conflicting tasks that the player must organize efficiently under pressure.

This mapping ensures that learning is embedded in action. The player is not just exposed to educational content—they must actively use and refine it in order to succeed. Research from frameworks like co.LAB (Jaccard et al., 2021) and UDL (CAST, 2018) supports this principle: learning is most effective when embedded within authentic decision-making, not treated as an add-on.

As the co.LAB framework notes, “the successful alignment between game mechanics and learning mechanics is [...] an essential feature for the success of serious games” (Jaccard et al., 2021). Similarly, UDL guidelines emphasize that “barriers can be reduced when options are available that support learners in connecting relevant prior knowledge, or link to the prerequisite information elsewhere” (CAST, 2018).

3.4.2 Theme and Mechanics: Cognitive Support and Emotional Engagement

Integrating the **Theme** with **Mechanics** serves two important roles. First, it supports learning by anchoring new concepts in familiar, intuitive experiences. This technique—often used in teaching math through concrete examples like apples or coins—can be extended to more complex subjects using thematic metaphors. For example, a game about managing a spaceship crew may use power distribution mechanics to teach resource allocation or systems thinking.

Second, thematic integration boosts engagement through emotional resonance. Players are more likely to persist through difficulty when the game world is interesting or meaningful to them. As Csikszentmihalyi’s theory (Ackerman, 2021) of flow highlights, attention and deep learning are enhanced when individuals are immersed in personally relevant, goal-driven experiences.

This dual effect is supported by several findings. As cited in the Model-Driven Framework for Educational Games (Roungas, 2016), Malone (1981) argued that effective games often provoke fantasy—“the mental images evoked by the user through the game environment”—and curiosity. Garris et al. (2002), also quoted in the same work, later refined this idea by introducing mystery as a key motivational driver, stating that “mysteries presented in a game can evoke curiosity in the individual, pushing a desire for discovery.” Similarly, like previously mentioned the Universal Design for Learning (UDL) framework highlights the importance of helping learners access relevant prior knowledge by embedding new information in familiar contexts (CAST, 2018). These insights reinforce the educational value of well-integrated themes—not merely as decoration, but as cognitive and emotional scaffolds for deeper learning.

3.4.3 Atomic Unit and Theme: Conceptual Framing

The final integration—between **Atomic Unit** and **Theme**—creates a bridge that aligns the player's cognitive focus with the designer's educational goals. A well-chosen theme can shape how players think about the skills they are using, even before specific mechanics are introduced. For instance, presenting conflict resolution within a pirate negotiation narrative can make abstract social skills feel **tangible** and **situationally grounded**. This conceptual framing helps players recognize and internalize the relevance of their in-game actions to real-life contexts.

3.4.4 Triadic Integration

Ultimately, the goal of VIOLETA is not just pairwise alignment but full **triadic cohesion**. When Atomic Unit, Theme, and Mechanics are harmonized, the result is a learning environment where every element reinforces the others. Players are emotionally invested, cognitively supported, and continuously practicing real-world skills—without needing to leave the game world to "switch into learning mode."

This vision of interconnected design elements is well-supported by the literature. As noted in the co.LAB framework (Jaccard et al., 2021), "a successful design depends as much on the quality of each element as on the relevance and adequacy of the links between these elements". It is the strength of these relationships—not just the components themselves—that gives a serious game its coherence and effectiveness. The co.LAB authors even propose that systemic modeling approaches are particularly well-suited for educational game design, given that such games function as complex systems where mechanics, narrative, and pedagogy must be fully integrated.

3.5 Structured Guidance Without Creative Constraint

Creating a structured methodology for game design—one that ensures educational effectiveness without stifling creativity—is inherently challenging.

The more structure a framework imposes, the fewer degrees of freedom it leaves for designers to experiment, invent, and express personal vision. In this sense, structure and creativity can be seen as occupying opposite ends of a single spectrum. Any attempt to design a framework must choose a hypothetical position along this line, balancing clarity and constraint with openness and inspiration.

This is not a novel dilemma. Similar trade-offs appear in many creative domains. Consider color theory: beginner designers are often introduced to color harmonies such as complementary, analogous, or triadic schemes. These structured rules help ensure aesthetically pleasing combinations and avoid jarring visual results. However, experienced designers know when—and how—to break these rules deliberately, to evoke discomfort, tension, or uniqueness. For example, deliberately clashing colors might be used in a game to signal chaos or urgency, creating a visceral emotional response that would be lost in a conventional palette.

The same applies to game design. Novices benefit from scaffolding—a structured process that narrows the vast design space into manageable, comprehensible decisions. Experienced designers, on the other hand, may find such constraints limiting, especially if they have a strong vision or intuitive sense of what the experience should be. The VIOLETA framework addresses this tension by drawing on the cognitive architecture proposed by Nobel Prize winner Daniel Kahneman in *"Thinking, Fast and Slow"* (Kahneman, 2011).

Kahneman distinguishes between two cognitive systems. System 1 operates automatically, quickly, and **often subconsciously**, generating impressions, intuitions, and emotional reactions. In contrast, System 2 represents our **conscious, deliberate decision-making system**—it is slower, more deliberate, and capable of analytical reasoning. As Kahneman writes, "System 1 runs automatically and System 2 is normally in a comfortable low-effort mode, in which only a fraction of its capacity is engaged. System 1 continuously generates suggestions for System 2: impressions, intuitions, intentions, and feelings." (Kahneman, 2011)

I would say the VIOLETA framework structures the components that can be used as **tools** for designers. Imagine a music producer working on a new song. There are tools that help him shape the sound—like visual indicators showing how loud a sound is or how frequencies are distributed. These tools don't write the song for him, but they provide objective information that help him make informed decisions. Similarly, the components in the VIOLETA framework store information—such as mappings between skills and mechanics—and sometimes also include structured links that define how different parts of the system interact. Since these interactions and data representations are objective, they do not require creativity and are ideal targets for System 2 processing.

This includes, for example, processes that align cognitive difficulty with player skill level, or track how different gameplay elements reinforce the mastery of an atomic unit. These structural elements form the backbone of the framework, **enabling designers to focus their creative energy elsewhere**.

Importantly, VIOLETA achieves a hybrid balance by framing prompts that activate creative intuition while still embedding structure. Consider a step in the design process that simply asks: *"Find a theme in which the skills of your atomic unit would naturally be used."* This **preserves complete freedom** in how the designer imagines the theme—whether it's a pirate negotiation, a courtroom drama, or a sci-fi expedition—yet the structure of the question ensures that the theme is **pedagogically meaningful**. It nudges System 1 to autocomplete an answer in a way that inherently supports integration, without limiting creative options. In this way, VIOLETA doesn't dictate the final design, but orchestrates the cognitive environment in which good designs are more likely to emerge—regardless of the designer's level of experience. Structured where it helps, intuitive where it matters.

3.6 Practical Application of VIOLETA: A Step-by-Step Walkthrough

While there are multiple ways to integrate the three core pillars of the VIOLETA framework—Atomic Unit, Theme, and Mechanics—this thesis follows a specific order of integration. This sequence, developed through the author's design experience and pedagogical reasoning, is believed to be the most intuitive and structurally coherent. What follows is a detailed, step-by-step walkthrough of how the framework can be used in practice, using a concrete example to illustrate the reasoning and outcomes at each stage.

3.6.1 Step 1: Define the Atomic Unit

The first step in applying the VIOLETA framework is to answer a deceptively simple question:

"What is the atomic unit that the game should be based on?"

An **atomic unit**, as defined in the framework, refers to a bundled set of real-world knowledge, behaviors, or skills that should be learned together as a cohesive whole. It is the educational "nucleus" of the game—what the game aims to teach or train, not as isolated trivia, but as a functional, applicable cluster of competencies.

Atomic units can vary greatly in complexity. They may be **as narrow as a single soft skill**—such as assertive communication or basic mental arithmetic—or as broad **as an entire interdisciplinary domain**, such as systems thinking or quantum physics. The choice of atomic unit depends on the designer's educational goals, target audience, and contextual constraints.

For the purposes of this walkthrough, time management will be used as an example of an atomic unit.

3.6.2 Step 2: Deconstruct the Atomic Unit Extract Skill Kernels

Break the atomic unit into its component competencies **and** crystallise the logical core (the "kernel") of each competency so later design steps can re–skin it with perfect structural fidelity.

Once the atomic unit has been defined, the next step is two–fold:

1. **Generate the candidate skill list** by asking: *"What knowledge, actions, and/or skills are necessary to master this atomic unit?"*
2. **Extract the kernel** of every skill by writing a one–sentence, verb–driven causal formula (*Input* $\rightarrow$ *Transformation* $\rightarrow$ *Output*).

This dissection serves multiple purposes. First, it provides granular control in later stages of the design process, particularly during theme and mechanics integration. Second, and more subtly, it ensures **internal cohesion**: by identifying sub–skills that all contribute to a common overarching goal, the designer avoids the risk of mixing unrelated competencies simply for the sake of variety.

Cohesion is particularly important in serious games because it allows thematic elements, player decisions, and feedback systems to all point in the same conceptual direction. This not only strengthens the educational impact but also improves player immersion. A player who senses that all game elements are "pulling in the same direction" is more likely to perceive their actions as meaningful and transferable to real life. In essence, this alignment embodies **structural isomorphism**: the in-game cause-and-effect chains mirror the real-world logic of the skills being taught, further reinforcing both immersion and learning transfer.

2–A Generate the Candidate Skill List

List every *knowledge, action, or skill* a fluent practitioner must command—no matter how small. Each item becomes a potential hook for mechanics, narrative beats, and assessments, while the complete list guarantees conceptual cohesion.

2–B Extract the Kernel

For every skill on your list, distil its causal logic into a single sentence using plain language. Strip away domain–specific jargon; keep only the core transformation:

- **Format:** `active-verb` + object + outcome qualifier
- **Example (Encryption):** Transform readable data into unreadable data with a reversible rule."
- **Example (Time–tracking):** Record how time is allotted so actual usage becomes visible."

Why this matters

- Serves as a structural checksum: if a future fantasy or sci–fi analogue violates the kernel, the mapping must be refined.
- Primes the metaphor search in Step 3 by exposing the bare mechanics that need re–skinning.
- Provides diagnostic language for play–testing ("Did players actually transform X into Y with rule Z?").

How to write a kernel

1. **Name the Input:** What is acted upon?
2. **Specify the Transformation:** What happens to the input?
3. **State the Desired Output/Goal:** How do we know we're done?

2–C (Optional) Cluster Label Skills

When the list grows unwieldy, group related kernels into thematic buckets to reduce cognitive load during later mapping stages. These clusters are scaffolding only; adapt or discard as needed.

Example buckets for the *Time Management* atomic unit

- **Use Productivity Tools:** Time Tracking; Task Management; Calendar and Scheduling; Knowledge/Note Systems; AI–powered Productivity; Focus Distraction–Blocking Tools
- **Apply Established Techniques:** Goal Setting; Minimise Context Switching; 80/20 Principle; Energy Management
- **Develop Personal Skills:** Learning to Say No; Prioritisation Frameworks; Planning and Delegation; Focus and Concentration; Delayed Gratification Control
- **Build Sustainable Habits:** Adequate Sleep; Avoiding Procrastination; Consistent Work Environment

2–D Quality Gate

Before advancing to Step 3, verify that **every** sub–skill:

1. Directly advances the atomic unit's mastery.
2. Possesses a clear kernel sentence.
3. Can be observed in–game (even if abstractly).

If an item fails any check, refine or remove it. The tighter this list, the smoother your theme–mechanic mapping will be.

Outcome: A vetted catalogue of sub–skills, each with a one–sentence kernel, optionally grouped for clarity. This artefact becomes the hand–off into Step 3 (*Theme Analogy Mapping*), where each kernel will be matched with an isomorphic scenario.

3.6.3 Step 3: Commit to a Theme & Validate Every Kernel

Once the atomic unit and its component skills are defined, the next step is to select a theme—a narrative setting and emotional context in which the skills of the atomic unit **are naturally relevant and continuously engaged**. The process is deliberately simple but uncompromising:

Guiding question (Step 3A):

"In what kind of world or situation would someone need to use the skills from my atomic unit regularly?""

Choosing the theme first taps the designer's System 1 creativity; the subsequent kernel-fit pass (Step 3B) brings System 2 rigour, ensuring structural isomorphism and pedagogical integrity.

3A Pick a Candidate Theme

- Write a two–sentence mood blurb that captures the world's flavour, stakes, and typical activities.
- No constraints on realism vs. fantasy, so long as the theme **naturally motivates** the learner to practise the atomic unit's skills.

Example Theme (high-magic academy) "You are an apprentice archivist at the Arcane University, safeguarding knowledge through spells that conceal, transform, or reveal ancient texts while rival houses scheme to steal them."

3B Kernel-by-Kernel Mapping—100 % Coverage Required

Guiding question *Inside my chosen theme, what in-world element replaces each kernel's* input*, what action expresses the kernel* verb*, and what tangible result stands in for the* output*?*

Create a concise table (or spreadsheet) with one row per kernel from Step 2. Mark the mapping with a simple Y/N in the final column only if it preserves the Input → Transformation → Output logic.

Proceed to design step 4 *only when every row is marked "Y"*.

Real-world Input	Theme Input	Verb	Theme Output	Fit?
file (plaintext)	parchment scroll	transform	glowing runic script	Y
measure task time	track spell casting	record	visible time-sigils	Y
...	...	...	...	...

- **Coverage rule:** Proceed to the next design step *only if every kernel row is succesfully transformed.*
- Any failure means the theme *must* be revised (swap objects, add a micro-locale) or a new theme chosen altogether.
- Aim for the simplest swap first—usually replacing the Input and Output nouns—before discarding the theme.

Outcome. A *Kernel Mapping Table* that proves 100% structural isomorphism between the chosen theme and every competency in the atomic unit. With this artefact locked, designers can move confidently into mechanic prototyping, knowing that each gameplay interaction will echo a real-world skill without conceptual drift.

This formulation is designed to trigger System 1 intuition. Instead of forcing a rational, top-down selection of theme, the questions gently activate the designer's subconscious ability to associate real-life scenarios with the skills involved. It encourages creativity while keeping the design anchored in pedagogical relevance.

While there are no restrictions on the type of theme—realistic, fictional, metaphorical, or fantastical—the key is that the **theme provides a meaningful context for the atomic unit** to manifest and develop. A good theme not only helps players emotionally invest in the game world, but also ensures that practicing the target skills feels logical and engaging within that context.

3.6.4 Step 4: Define the Intended Emotional Experience

With the atomic unit and theme selected, the next step is to articulate the **emotional experience** the player is intended to go through while engaging with the game. This is not merely about moment-to-moment emotions (e.g., surprise, joy, frustration), but the overall **emotional arc** that emerges through gameplay. To surface this, the designer can ask:

"If someone truly lived inside this theme, while actively using the atomic unit skills, what would they gradually feel—and why?"

This question helps uncover what kind of internal experience the game should elicit. By framing it as a hypothetical lived scenario, it activates intuitive reasoning (System 1) while still focusing the designer on emotional relevance and authenticity.

At the end of this step, the designer is encouraged to write a **short description of the emotional journey**, followed by **a concise list of the most important feelings** they want the player to experience. This not only supports intuitive thinking but also serves as a reference point in later steps—especially when tuning mechanics, pacing, and feedback systems to reinforce these emotional states.

In the case of our Entrepreneur-Student-Athlete theme, the following emotional trajectory is envisioned:

Feeling: Gradual Control Amid Chaos

The player begins in a state of **constant pressure**—drowning in expectations and to-do lists, feeling like it's almost fantastical to accomplish everything. Initially, success seems out of reach, not because of laziness but because of overwhelming complexity. As the game **progresses**, the player starts identifying key levers: what to prioritize, how to schedule better, what to abandon. **A sense of control** emerges, not because the world became easier, but because the player became more skilled. This emotional journey—from chaos and pressure to increasing clarity and mastery—parallels the real-life experience of learning time management.

Summary of Key Emotional States

- Constant pressure and overwhelm
- Gradual control of life
- Satisfying sense of progress

These emotional arcs are naturally subjective and depend on the designer's perspective, but they should be plausible outcomes of the chosen theme. To help illustrate the range of possibilities, here are several additional examples showing how this step could be approached in different contexts:

Additional Examples

Example A: Psychological Resilience in a Post-Apocalyptic Survival Theme

Atomic Unit: Psychological resilience and adaptive coping.
Theme: A lone survivor navigating a ruined world, trying to rebuild a small community.

Feeling: Fragile Hope Amid Harsh Reality

The game starts with a bleak tone—scarce food, dangerous weather, and distrustful people. Yet each small success builds hope: planting crops, saving a life, repairing a shelter, gaining someone's trust. Over time, the player develops emotional resilience—not through dramatic victories, but through persistence, quiet growth, and care for others. The core transformation lies in finding meaning amid hardship.

Summary of Key Emotional States

- Helplessness and vulnerability
- Grit and determination
- Subtle, fragile hope
- Meaning through small victories

Example B: Strategic Thinking in a Space Colony Theme

Atomic Unit: Strategic planning and long-term decision-making.
Theme: A team of pioneers establishing the first human colony on Mars.

Feeling: Calm Urgency and the Weight of Decisions

The player is constantly pulled between short-term survival and long-term prosperity. Every decision feels high-stakes: whether to ration water, invest in unstable energy tech, or expand too soon. Gradually, the player shifts from reactive to proactive thinking, enjoying the challenge of forecasting consequences and designing resilient systems.

Summary of Key Emotional States

- Tension between immediate and long-term priorities
- Responsibility and pressure to make high-impact choices
- Growing confidence in complex planning
- Satisfaction from achieving sustainable stability

Example C: Empathy Development in a Medical Internship Theme

Atomic Unit: Empathy and patient communication.
Theme: A young medical intern navigating their first rotation in a chaotic hospital.

Feeling: Overwhelm Turning into Human Connection

The experience begins with anxiety: too many patients, not enough knowledge, and intense supervision. Mistakes happen. But slowly, the player realizes that listening and connecting with patients leads to clarity, cooperation, and better outcomes. Emotional intelligence becomes as important as clinical accuracy, leading to moments of unexpected human warmth.

Summary of Key Emotional States

- Anxiety and performance pressure
- Discomfort after early missteps
- Calm and emotional clarity through empathy
- Fulfillment from relational trust and care

It is crucial to emphasize that at this stage, we are **not designing a game**. We are not concerned with mechanics, rules, win conditions, or how the game will function at the systemic level. This step is focused entirely on defining a **compelling and cohesive theme**, along with the **emotional states** it naturally evokes when the atomic unit is regularly used within that world. The short scenario we write is not meant to define gameplay, but rather to build a consistent story—a narrative context where the atomic unit is naturally embedded and emotionally charged.

Just as we first defined an atomic unit and only then broke it down into its supporting skills—ensuring internal cohesion—the same logic applies here: we first define the thematic context, and only then extract the main emotional states. This order is intentional. It allows the feelings we identify to be **cohesive with each other**, as they all emerge from the same thematic experience. By breaking the experience down into **granular emotional components**, we gain clarity and structure that will be essential during later integration with game mechanics.

To restate: **we are not defining the game in this step**. Our goal is to define a story-world that is emotionally rich and conceptually aligned with the atomic unit. The mechanics will emerge later, but only if the emotional and narrative foundation is strong. Prioritizing this foundation ensures that gameplay, when introduced, will support and reinforce the learning goals, rather than being forced to fit them retroactively.

3.6.5 Step 5: Layer Feelings (LF) Process

Before transitioning to mechanics, designers may choose to perform a **Layer Feelings (LF)** process—a lightweight analytical step that introduces structure into the emotional design space. The LF process takes the key emotional states identified in the previous step and explores whether any hierarchical or causal relationships exist among them.

For example, in the case of our time management game, the three identified emotional states were:

- Gradual Control of Life
- Progress
- Constant Pressure

By applying the LF process, we might identify a structure such as:

- Progress
  - Gradual Control
- Constant Pressure

Alternatively, one could argue for:

- Gradual Control
  - Progress
- Constant Pressure

Or even:

- Gradual Control
- Constant Pressure
  - Progress

These arrangements are entirely **subjective** and open to interpretation. Designers are encouraged to develop their own reasoning for why certain emotions build upon or lead to others. In some cases, there may be **no clear hierarchy**, and the designer may choose to treat all emotional states as parallel.

While this step is subjective and creative, it becomes valuable in the next phase—**mechanics integration**—where emotional states and gameplay elements will be linked structurally. Having even a loose hierarchy or grouping at this stage provides a **bridge from intuition to structure**, ensuring that the emotional journey defined earlier is carried forward in a way that is **logically consistent and pedagogically meaningful**.

3.6.6 Step 6: Form A Game (FAG) – Integrating Emotions into Mechanics

In this step, we begin the process of turning the abstract results of the creative phase into concrete game systems. We use the previously defined Layer Feelings (LF) structure as the foundation for **integrating mechanics**—the rules and interactive systems through which players engage with the game.

The core task is to **find or invent game mechanics that reflect, evoke, or reinforce each emotional state** defined in the LF hierarchy. To guide this process, the designer can ask:

"What kind of mechanic would make the player feel this emotional state naturally, through gameplay—not through narrative or dialogue?"

This is not a question about how to represent a feeling, but how to **build a system where that feeling emerges as a natural consequence of interaction.**

To carry this out:

- Begin at the root of the LF hierarchy and move downwards.
- For each emotional state, define at least one mechanic that can evoke that feeling in the player.
- When dealing with child emotions, ensure their mechanics also support or feed into the emotional state of their parent—preserving the emotional logic of the LF structure.

This is a creative step with structural guidance. Designers are free to use mechanics from any medium—video games, board games, tabletop RPGs, even analog systems—as long as the emotional effect is preserved. Unlike some existing frameworks that prescribe specific lists of mechanics, VIOLETA remains fully open-ended, allowing maximal flexibility for experienced designers.

That being said, one of the potential weaknesses of this approach is that inexperienced designers may struggle to generate appropriate mechanics without guidance. A potential extension of the framework could include a curated, non-prescriptive mechanics library, where designers can browse examples categorized by emotion, pacing, or player interaction mode.

Example: Applying FAG to the Time Management Game

Recall our Layer Feelings structure:

- Progress
  - Gradual Control
- Constant Pressure

Mechanics:

- **Progress:** Achieved through **deck-building** and **skill tree**. As the player collects more cards or unlocks new branches in the skill tree, their overall capacity and efficiency increase, reflecting meaningful development over time.
- **Gradual Control:** Reinforced through **increasing player agency (amount of options available)**, such as the number of available actions or branching tactical choices. The more the player masters time management techniques, the more flexible and precise their available tools become (e.g., more action slots, more powerful cards, better combo chains). This also feeds into the feeling of Progress.
- **Constant Pressure:** This can be achieved through **tight margins of success**. The game is tuned so that players can only complete all their goals if they make a very limited number of mistakes—perhaps no more than 5%. As a result, players will often fall just short of full completion, creating a persistent sense of pressure. They are constantly aware that slipping up even slightly could prevent them from reaching their objectives, which mirrors the real-life stress of juggling too many tasks with too little time.

Now that mechanics have been assigned to each emotional state in the Layer Feelings (LF) structure, the next step is to **transform that emotional hierarchy into a mechanical structure**. This process creates what we call the **Base Mechanics Tree (BMT)**.

To build the BMT, we replace each emotional state in the LF with the game mechanics that were selected to evoke it. If a feeling is supported by multiple mechanics, each mechanic becomes a separate branch under the parent. Additionally, if a mechanic supports more than one parent node (for example, if it connects both a child and parent emotional state), it can appear in multiple places in the tree.

Example Base Mechanics Tree (BMT)

Original LF

- Progress
  - Gradual Control
- Constant Pressure

Chosen Mechanics

- Progress → Deck Building, Skill Tree
- Gradual Control → Amount of Options Available
- Constant Pressure → Tight Margins of Success

Final BMT

- Deck Building
  - Amount of Options Available
- Skill Tree
  - Amount of Options Available
- Tight Margins of Success

In this case, the emotional state Progress is represented by two core mechanics—deck building and skill tree. The child emotion Gradual Control is realized through the increasing number of meaningful options available to the player. The emotional state Constant Pressure is translated directly into the Tight Margins of Success mechanic.

The Base Mechanics Tree serves as the **structural foundation** for translating the emotional design into gameplay systems. It preserves coherence and intent by aligning emotional states with player interactions. While it does not directly guide difficulty tuning, pacing, or mastery systems, it provides the conceptual groundwork that is later expanded upon through tools like the Triadic Integration Table (TIT), the List of Schemas (LoS), and cognitive load matrices.

We have now came to the **crucial moment** of the design. The thing we want to do is to create **concrete game rules/mechanics/systems** at this point. As for BMT we (could) have used very abstract wording, it could be very difficult to come with specific things.

For example choosing the "Deck Building" as game mechanic for the feeling progress, is a general term. What it also includes that was not noted is that to have a deck building, you need deck (cards) and then options to buy and/or sell cards in order to build your own deck. For me personally it also implies playing with cards and drawing them.

On the other hand if we take the "Skill Tree" there are no additional elements that I would declare. Skill tree consists of (possibly) branching options and we simply have to select an option that would unlock something for us in game. So while Deck building contains additional elements, skill tree, for me personally, does not, and doesn't need any extension.

Tight Margins of Success is the most general/abstract of the three. It could be implemented in any way. It's not a single element like Skill Tree (again, personally), nor is it like Deck Building which has a couple of extra elements. Tight Margin of success is so open, that we could come up with many systems that would support it. The BMT wording was done in this way with this exact purpose to let the designers come up with their own vague mechanics that later can be created into concrete systems.

So this was done on **purpose**. As the whole point of the BMT is to maintain the **integration of theme feelings with the game mechanics**, the vagueness of the game mechanics is not questioned or frowned upon. This is very delicate and maintaining this integration was of **utmost importance**, thus the creativity allowed. However, because of this creativity it is very hard to find a proper structure that would lead to a MVP (Minimum Viable Prototype) of the game. Hence there isn't a single question that could be asked to utilize System 1 at this moment.

3.6.7 Step 7: From Base Mechanics Tree to MVP – Building a Playable Prototype

With the Base Mechanics Tree (BMT) completed, the next phase in the VIOLETA framework is the development of a Minimum Viable Prototype (MVP). This prototype serves as the critical foundation for all further steps, including triadic integration and difficulty alignment. Although this prototype is intentionally minimal and incomplete, it should already contain the core mechanics and emotional structure that define the game's intent.

This stage represents a shift from abstraction to implementation. The designer moves from selecting emotionally aligned mechanics to **defining the concrete components, rules, and systems** needed to bring those mechanics to life. While experienced designers may already have an intuitive workflow, VIOLETA offers an **optional structured, iterative, and recursive method** to help less experienced designers translate their BMT into a coherent MVP.

Purpose and Outcome

- **Goal:** To produce a playable game prototype that expresses the chosen mechanics and emotional experience.
- **Outcome:** A complete **List of Schemas (LoS)**—a catalog of all gameplay-relevant components, systems, and information structures.

Suggested Recursive Approach

The framework provides the following guided process for building the MVP from the BMT:

1. Pick an entry from the BMT. Start with any major mechanic from the BMT (e.g., Deck Building, Skill Tree, Tight Margins of Success).
2. Ask:

    > *"Which concrete game elements does this mechanic consist of?"*

    This question helps you break down abstract mechanics into gameplay-relevant systems, such as components, actions, or rules.
3. For each new game element identified, ask:

    > *"Given our atomic unit, what would this element represent in the theme, and how would it function within the game mechanics?"*

    This step ensures that every element remains consistent with the original skill-mapping and emotional design.

4. Repeat Steps 2–3 recursively for each new element. Continue the process until no further decomposition is necessary.

5. Once complete, move to the next BMT entry and repeat. This recursive approach ensures that each mechanic is expanded fully and all gameplay elements are aligned with the emotional and educational goals.

Example (Partial Breakdown for Time Management Game)

Starting Mechanic: Deck Building

1. **Deck Building consists of:**
   - Playing Cards
   - Drawing Cards
   - Buying/Selling Cards
2. **Cards represent:**
   - Actions taken in everyday life (productive or unproductive).
   - Each card occupies 1 hour on a Daily Board.
3. **Daily Board:**
   - A component with 24 time slots representing a full day.
   - Cards must be placed in sequence, mimicking the passage of time.
4. **Play Cards:**
   - The core daily activity.
   - Players play all cards from their hand in sequence.
   - This simulates real-life scheduling and prioritization.
5. **Draw Cards:**
   - At the start of each day, players draw a hand from their personal deck.
   - Cards include both productive actions (e.g., study, exercise) and negative ones (e.g., distractions, procrastination).
6. **Buy/Sell Cards:**
   - Productive cards can be acquired through spending progress points.

- Negative cards can be removed to improve future turns.
- Represents the real-world ability to form habits or eliminate bad routines.

Submechanic: Amount of Options Available This submechanic plays a vital role in evoking the emotional trajectory of gradual control. The goal is to guide players through an initial experience of constraint and frustration, eventually allowing them to unlock increasing degrees of agency and mastery. To apply this, each child element under a major mechanic (like Deck Building) must be evaluated in terms of how it supports or evolves the amount of strategic options available to the player over time.

Let's illustrate this through a focused example on **Draw Cards**, a subelement of Deck Building:

Focus: Draw Cards

Originally defined as:

- At the start of each day, players draw a hand from their personal deck.
- The deck includes both productive (e.g., *Study*, *Exercise*) and unproductive (e.g., *Procrastinate*, *Distractions*) cards.

To reinforce the emotional arc of control:

- **Initial Constraint:**
  At the start of the game, players have a **hand size limited to 2 cards**. This restriction severely limits planning and forces players to react with minimal agency. Furthermore, players draw all cards at once, and only after they have played all of their cards; reinforcing the feeling of being pulled along by circumstance.
- **Forced Play:**
  Since they cannot skip cards or selectively hold cards for later turns, players must **play both good and bad cards**, simulating the emotional reality of chaotic time management where unproductive behaviors cannot be easily avoided.
- **Unlocking Control:**
  As the game progresses, the player gains access to upgrades (via the Skill Tree) that **increase hand size** and allow more flexibility in drawing and playing cards. With a hand size of 4, players can now plan a

half-day in advance, skip or delay unproductive actions, and experiment with optimized sequences.

- **Emotional Outcome:**
  This mechanic transition—from having no real choices to meaningful control—evokes a shift from **frustration to satisfaction**. Initially, players feel **disoriented and restricted**, unable to meaningfully plan their day or avoid bad actions. This mirrors the emotional chaos of poor time management in real life.

However, as their hand size grows and upgrades unlock more agency, players begin to plan ahead, structure their days, and optimize card sequences. Importantly, the emotional payoff here is not tied to big achievements or dramatic successes—**it emerges simply from being able to plan at all**.

This moment is crucial.

In the context of the Atomic Unit for time management, **planning** is one of the central real-life skills the game aims to foster. By designing early gameplay to be frustratingly unplannable, the framework creates a baseline that **amplifies the emotional satisfaction of unlocking planning ability**. The player does not just appreciate planning—they crave it, struggle for it, and feel genuine relief when it becomes possible.

This transforms planning from a background utility into a **rewarded, emotionally anchored skill**, tightly linking game mechanics to real-world mastery. It exemplifies how the VIOLETA framework operationalizes the principle that "**more play = more mastery**"—not only through content, but through carefully crafted emotional journeys embedded in mechanic design.

This analysis shows how "Amount of Options Available" is not just a balance or pacing issue, but a **deliberate emotional design lever**. Each mechanic, when examined through this lens, can be refined to produce satisfying player arcs of growth.

Recursive Reminder:

Repeat this same reflective process for other child mechanics under the BMT (e.g., Buy/Sell Cards, Goal Systems).

Some example questions for "Amount of Options Available":

- *Does this mechanic start in a constrained form and gradually expand?*
- *Does its expansion feel meaningful to the player?*
- *Does it support the intended emotional arc (e.g., autonomy, mastery, pressure)?*

Repeat the same process for the **Skill Tree** and **Tight Margins of Success**, expanding each into its full game system.

What Is the "List of Schemas" (LoS)?

The LoS is the **end product of this recursive definition process**. It includes:

- **Components** (e.g., cards, boards, tokens)
- **Player Actions** (e.g., play card, upgrade skill)
- **Information Schemas** (e.g., available mana, stress levels, progress points)
- **Passive Systems** (e.g., stat boosters, fatigue modifiers)
- **Structural Elements** (e.g., daily/weekly/monthly loops, upgrade trees)

This list is what allows later phases of the framework—such as difficulty balancing and educational evaluation—to operate on a clear, testable foundation.

Why This Step Matters

This is a generative step. The goal is not to prescribe the final game form but to **ensure every mechanic has meaningful, playable representation** that supports the original emotional and educational design.

Furthermore, the process stimulates System 1 creativity: by following a series of small, intuitive questions, the designer gradually builds complexity without becoming overwhelmed. This approach also ensures that every rule, component, and interaction is purposeful—either supporting the original mechanic or contributing to the intended emotional journey.

Summary

The Prototype Development phase is a flexible, creative expansion of the BMT into a working MVP. While it lacks rigid structure, it is bound by a simple logic: **only include mechanics that support the original mechanics and emotions**.

The goal is not to create a polished or final game, but to produce a **minimal, functional prototype**—a core version that enables early playtesting and evaluation. Importantly, this prototype is **not fixed**: many mechanics, systems, and components defined here may change in the following steps. As integration and balancing processes begin, adjustments will be necessary to maintain alignment between learning goals, emotional experience, and gameplay dynamics.

What matters most at this stage is having a **starting point**—a playable foundation that anchors the upcoming work. This MVP serves as the entryway into two of the most critical phases of the VIOLETA framework:

- **Triadic Integration:** systematically linking the Atomic Unit, Theme, and Mechanics.
- **Difficulty Alignment:** adapting the game to match varying learner skill levels.

Without a concrete prototype, these stages cannot be meaningfully applied. Therefore, while the MVP may still be rough or incomplete, it plays a strategic role—bridging creative design with structured educational impact.

It is important to note that the iterative process presented in this step is **not mandatory**. It is a **suggested approach**, especially helpful for designers who may be unsure where to start. The process leverages System 1 thinking by prompting action through small, incremental design decisions—each rooted in the BMT. However, experienced designers may reach a working MVP through entirely different workflows. What matters is not how the prototype is created, but that it remains **coherent with the emotional structure and mechanical intent** defined in earlier steps.

3.6.8 Step 8: Creating the Scaling Influence Table (SIT)

In this step we take the List of Schemas (LoS) and use its mechanics to construct a triadic link between *atomic-unit skills*, *emotional design*, and *gameplay*. The tool for this purpose is the **Scaling Influence Table (SIT)**.

Structure.

- **Column 1** lists every *atomic-unit skill* you wish to map.
- **Subsequent columns** list *all emotions in the game's emotional palette—including parent and child emotions.*[1]

Marking direct influence. For each cell, place a "+" if the skill *directly* influences that emotion. Leave the cell empty (or mark "-") if the influence is indirect or non-existent.

- **Example.** *Planning* → *Progress* receives "+", because careful planning immediately reinforces a sense of progress. *Planning* → *Constant Pressure* remains blank, since pressure in this prototype stems from *Goal Setting.*

This step is inherently *creative and interpretative.* Different designers may arrive at different mappings; what matters is that each "+" can be justified in clear, pedagogically sound terms.

Note. Mark "+" only for direct causation. Indirect chains (e.g. a skill → intermediary → emotion) are *not* recorded in the SIT.

Example SIT

Atomic Unit Skill	**Progress**	**Gradual Control**	**Constant Pressure**
Planning	+		
Energy Management	+		
Task Management	+	+	
Calendar Use	+		
Goal Setting	+	+	+

Once the SIT is complete we build a dedicated **Triadic Integration Table (Kernel-level) – TIT-K** *for each (skill × emotion) pair marked with "+".* The TIT-K embeds every kernel of that skill into candidate mechanics drawn from the BMT branch of the same emotion, ensuring that real-world mastery, emotional intent, and gameplay remain tightly interwoven.

[1] Listing child emotions up-front makes later automation simpler; no additional expansion step is required when filtering for mechanics.

Constructing the TIT-K

For every *(atomic skill × emotion)* pair that carries a "+" in the Scaling Influence Table (SIT), create a separate **Triadic Integration Table – Kernel (TIT-K)**:

- **Column 1 – Kernel ID & Label**
  List each kernel that belongs to the selected atomic skill (e.g. `K_22b Align task to peak e`
- **Columns 2. . . N – Candidate Mechanics**
  Add *one column for every concrete mechanic or component* found under the BMT branch that corresponds to the chosen emotion. *Example for* Progress*:* [`Skill-Tree Entry` `Play Card` `Draw Card` `Sell Card` `Buy Card`]
- **Final Column – Result**
  Mark each kernel–mechanic pairing as **Accepted**, **Revised**, or **Rejected**, and add a brief rationale.

Evaluating Each Row

1. **Kernel–Purpose Fit**
   "What real-life micro-action does this kernel train, and how can the mechanic enact that purpose?"
2. **Result Rules**

   - purpose realised → **Accept** & copy to LoS
   - partial fit (tweak viable) → **Revise**
   - purpose contradicted → **Reject**

This lean layout lets you scan, per skill–emotion pair, which specific mechanics best express each kernel, while the Result column keeps the LoS and subsequent design layers in sync.
This table must be built for each atomic unit skill and its influenced emotions. Although time-consuming, this guarantees strong **internal consistency** and maintains the framework's principle that more play = more mastery. In future work, this manual process could be semi-automated through intelligent tooling or game design assistants.

With this, the **triadic integration** process is complete. The game now contains a tightly interwoven structure where **real-life skills are meaningfully represented** through game mechanics. At the same time, **emotions and theme are deeply embedded** to support a smooth cognitive transition between prior knowledge and new learning content. This design ensures that gameplay is not only engaging but also developmentally aligned with the player's growth.

3.6.9 Step 9: Calibrate Challenge – Maintaining Competence Through Adaptive Difficulty

The next step focuses on introducing a mechanism that allows developers to **dynamically align the difficulty of the game with the player's evolving capabilities**. This functionality is grounded in **Cognitive Load Theory** (Sweller et al., 1998), which forms the theoretical basis for the **STACK** process (**Structured Tiered Adaptive Core Knowledge**). STACK provides a structured method for adjusting game complexity in real-time to prevent both cognitive overload and under-stimulation.

According to Self-Determination Theory (SDT), the psychological need for **competence** is a key prerequisite for sustaining **intrinsic motivation**. Players must feel capable of mastering the challenges presented to them in order to remain engaged and motivated. To support this, the STACK process is introduced. STACK dynamically adjusts the difficulty of the game to maintain an optimal challenge level—ensuring that the player is **neither under-stimulated** (which would lead to boredom and disengagement), nor **overburdened** (which would obscure their sense of competence and create frustration).

SDT also notes that when challenges exceed a player's current abilities, high-quality feedback can help mitigate the negative effects of cognitive overload. In line with this insight, the VIOLETA framework includes an additional support mechanism called the **3E** (**Exposure, Experiment, Evaluate**), which will be introduced later. This process is designed to scaffold player growth by encouraging exploration and reflection, even when direct success is not immediately achievable.

To fully understand the function of STACK, we must first revisit how learning and memory are conceptualized within Cognitive Load Theory (CLT). According to CLT, the human mind relies on two distinct memory systems: **working memory** and **long-term memory** (Sweller et al., 1998). Working memory is extremely limited—capable of actively processing only about **seven elements or fewer** at once (Miller, 1956 as cited in (Sweller et al., 1998)). When these elements interact or require manipulation, such as in problem-solving tasks, the effective limit is often much lower (Sweller et al., 1998).

In contrast, long-term memory has a virtually unlimited capacity and stores **schemas**—structured bundles of knowledge. Schemas allow us to treat many interacting pieces of information as a single unit in working memory. For instance, the concept of a "restaurant" is just one schema in long-term memory, but it encapsulates a wide array of associated concepts such as food, money, service staff, social rules, and architectural layouts. Once learned, this complex bundle is treated as a single "element" in working memory, thereby **reducing cognitive load** and improving our ability to reason and perform tasks efficiently.

CLT emphasizes the importance of **schema construction and automation** in instructional design. The key challenge is managing **intrinsic cognitive load**—the mental burden imposed by the inherent complexity of the material, especially when elements are **interdependent** and must be processed simultaneously (Sweller et al., 1998). For example, learning a mathematical operation like multiplication relies heavily on having already internalized simpler schemas such as addition. It would be inefficient and overwhelming to teach multiplication to a learner who has not yet mastered addition. This is precisely where the STACK system comes in. STACK ensures that learners are exposed only to challenges appropriate to their current cognitive state by continuously monitoring the number of **schemas active in working memory**. A core principle of STACK is that a player should not be asked to handle more than their working memory can reasonably manage. It does this by enforcing **tiered progression**, only allowing advancement once the current tier's schemas are sufficiently internalized and automated. Consider a mathematical example. In STACK's tiered system:

- **Tier 1** might involve learning and mastering addition and subtraction.
- **Tier 2** would only be unlocked after demonstrating fluency in Tier 1 and might include multiplication and division, which build on those foundational operations.

Trying to teach multiplication without solidified knowledge of addition would result in **cognitive overload**, reducing both performance and long-term retention. STACK prevents this by calibrating the difficulty based on each player's demonstrated competence.

What makes STACK particularly powerful is that it respects **learner variability**:

- A highly skilled player might complete a level on the first attempt due to prior knowledge or fast learning. In that case, STACK immediately escalates the difficulty, introducing new mechanics or atomic unit skills. This keeps the experience **fresh, novel, and intrinsically motivating**—a crucial factor in sustaining engagement.
- Conversely, a struggling player will remain at the current level, receiving just enough challenge to stimulate learning without becoming overwhelmed. This careful balance nurtures the player's sense of **competence**, a key tenet of Self-Determination Theory (SDT).

In this way, STACK aligns cognitive demands with player ability at all times, ensuring:

- **No player is left** frustrated by an insurmountable wall of complexity.
- **No player becomes bored** due to repetitive or overly simplistic tasks.

By dynamically adjusting the presentation of atomic unit skills and game mechanics, STACK transforms the game into a self-regulating educational system. It builds upon CLT's insights by embedding a **diagnostic and adaptive difficulty manager** directly into gameplay. This allows the VIOLETA framework to uphold its core promise: the more the player engages with the game, the more real-life skills they master—without ever violating cognitive constraints that would otherwise inhibit learning.

3.6.10 Step 10: Temporal Structuring and Schema Graph Construction

Before defining STACK tiers, the VIOLETA framework introduces two structuring tools that guide the tiering process: the **Time Table (TT)** and the **Schema Graph (SG)**.

Defining the Time Table (TT)

Not all educational games are explicitly time-phased; some operate in continuous flow. However, when time-structured gameplay is present, it's important to make these phases explicit. The Time Table (TT) captures all distinct temporal segments in which game actions occur. In the time management prototype presented in this thesis, two primary phases have been defined:

Time:

- Play Phase
- Upkeep Phase

These phases form a repeating daily loop. The Play Phase represents the portion of the day when the player executes cards (i.e., actions), while the Upkeep Phase serves as a reset or status evaluation period where new cards, and progress points are managed.

Extracting Schemas from the LoS

The List of Schemas (LoS) represents the concrete gameplay components defined during prototype construction. Each schema reflects a mechanic or system that the player must interact with, internalize, and eventually automate. These schemas are the foundational blocks from which STACK tiers will be constructed. Below, we list and describe each schema as it currently appears in the time management prototype.

- **Draw Cards:** At the start of each in-game day (Play Phase), the player draws a number of cards equal to their current hand size. Initially, this hand size is very limited (e.g., 2 cards), reflecting early-game constraint and low agency. As gameplay progresses and the player unlocks skills such as Task Management or Calendar Use, the hand size increases, allowing more complex planning and decision-making.
- **Play Cards:** Players must place drawn cards sequentially into hourly slots on a Daily Board. For example, if the player begins with a start time of 9:00 and a hand of 2 cards, they may play into 9:00 and 10:00. With 4 cards, those slots extend to 11:00 and 12:00. This mechanic models real-life scheduling, encouraging the player to think in terms of temporal allocation and sequence.
- **Sell Cards:** The player may discard one card from their deck. This mechanic reflects the long-term optimization of one's routine by removing unproductive or negative habits (represented as cards), thereby increasing the likelihood of drawing better options in future turns.

- **Buy Cards:** The player selects one new card from a limited pool of X options. This introduces the concept of selective improvement—adding useful actions or habits to the deck in a deliberate, choice-driven manner. It ties into skill development and strategic planning, as players must evaluate not just the immediate value of a card but its long-term utility.
- **Pick a Progression Skill:** Players unlock a new skill from a progression tree. This schema introduces branching decisions and delayed benefits. Each skill has meaningful consequences for future play, such as modifying card behavior, altering draw/play rules, or enhancing feedback. It also serves as a narrative indicator of growth and mastery.
- **Energy Management:** This mechanic reflects the real-world phenomenon of fluctuating energy, motivation, and biological rhythms. Cards have an associated energy requirement to be played at full productivity. If the player lacks sufficient energy at the time of play, the card's effectiveness is diminished. The player has access to visual feedback on their current energy state, allowing them to align demanding tasks with energy peaks. Mastery of this mechanic enables strategic optimization and deep reflection on one's scheduling patterns.
- **Consider Goals:** At every decision point, the player must account for their goals—both short-term (daily) and long-term (weekly/monthly). These goals are not merely decorative; they influence scoring and success conditions. The cognitive demand here lies in maintaining these targets within working memory during gameplay. In real life, this mirrors the challenge of acting in accordance with one's priorities amid distractions and opportunity costs.
- **Task Management:**Unlocking this skill increases hand size by 3 and allows the player to play cards 3 additional hours ahead. This significantly expands strategic flexibility and deepens the temporal planning layer. It enables players to construct intricate routines and better accommodate multiple priorities. Mechanically, it represents the structuring and batching of tasks—a key concept in productivity science.
- **Calendar Use:** Adds 1 more card to the hand and extends play time by 1 hour. Additionally, it introduces inter-day planning: the player may now schedule cards into future days by burying them X cards deep into the deck. This requires mental calculation and forethought, rewarding players who engage with future-oriented strategy and capacity planning.

- **Planning:** Adds 2 cards to hand size and allows playing 2 hours ahead. More importantly, it reveals all upcoming goals for the month, offering transparency and foresight. This schema enables the player to act in a highly informed manner, connecting day-to-day actions to long-term ambitions. Planning thus becomes a cognitive skill modeled within the game, rather than a passive background.

- **Goal Setting:** Grants the player the ability to adjust the goals for each day, week, and month. While the total sum of required effort (points, hours, etc.) must remain constant, the player can redistribute focus areas—e.g., shifting emphasis from fitness to academics. This mechanic reflects both flexibility and constraint in real-world planning, fostering higher-order thinking about resource allocation and life balance.

- **Cards:** Each card represents a real-life activity. Cards encode multiple dimensions: time cost, energy cost, and productivity yield. They may represent either productive actions (e.g., exercise, study) or distractions (e.g., social media, aimless wandering). The player must evaluate not just the face value of a card, but how it fits into their energy curve, available time, and goal structure. Cards are the primary decision-making unit and the medium through which all gameplay unfolds.

*Note: The schema descriptions presented here are **partial example** of time management game. This simplification is intentional and serves to keep the example readable and focused, allowing the underlying VIOLETA processes to be more easily followed.*

Constructing the Schema Graph (SG)

With the List of Schemas (LoS) finalized, the next step in the VIOLETA framework is to visualize the **logical and functional dependencies** among schemas. This results in the **Schema Graph (SG)**—a directed graph where nodes represent schemas and arrows indicate prerequisite or enabling relationships. The SG is not merely a visual aid; it is a **strategic tool** for defining STACK tiers, ensuring that complexity is introduced gradually and pedagogically.

The goal of the Schema Graph is to answer two core questions:

1. **Vertical progression**:

"What must the player already understand before they can meaningfully interact with this schema?"

2. **Horizontal reinforcement**:

"Are there other schemas at the same level that mutually support or deepen understanding when learned together?"

To keep the diagram readable, we are working with a simplified version of the schema set.

Key Enhancements

- Bidirectional horizontal arrows (↔) between Task Management, Calendar, and Planning indicate mutual reinforcement. These schemas exist at the same tier and address overlapping problem spaces: scheduling, foresight, and temporal optimization.
- Learning one of them makes the others cognitively easier to grasp. For example:
  - Calendar introduces future planning via deck manipulation.
  - Planning expands goal visibility and strategic reach.
  - Task Management increases breadth of execution.
- The combination of these systems accelerates the construction of a **meta-schema**: the player's mental model of "*strategic time control*."

Why Horizontal Connections Matter

While vertical progression defines **what must come first**, horizontal connections acknowledge that **schemas are not always isolated units**. In reality, once one concept is internalized, similar or adjacent ones are acquired faster through:

- **Shared interfaces** (e.g., all three time-management tools affect hand size and play range),
- **Common mental models** (e.g., "what does future planning require?"), and

- **Transfer effects** (e.g., understanding energy tradeoffs in one schema helps forecast them in others). These lateral connections also help inform the composition of STACK tiers, allowing certain schemas to be introduced as bundles if their joint learning curve is smoother than learning them in isolation.

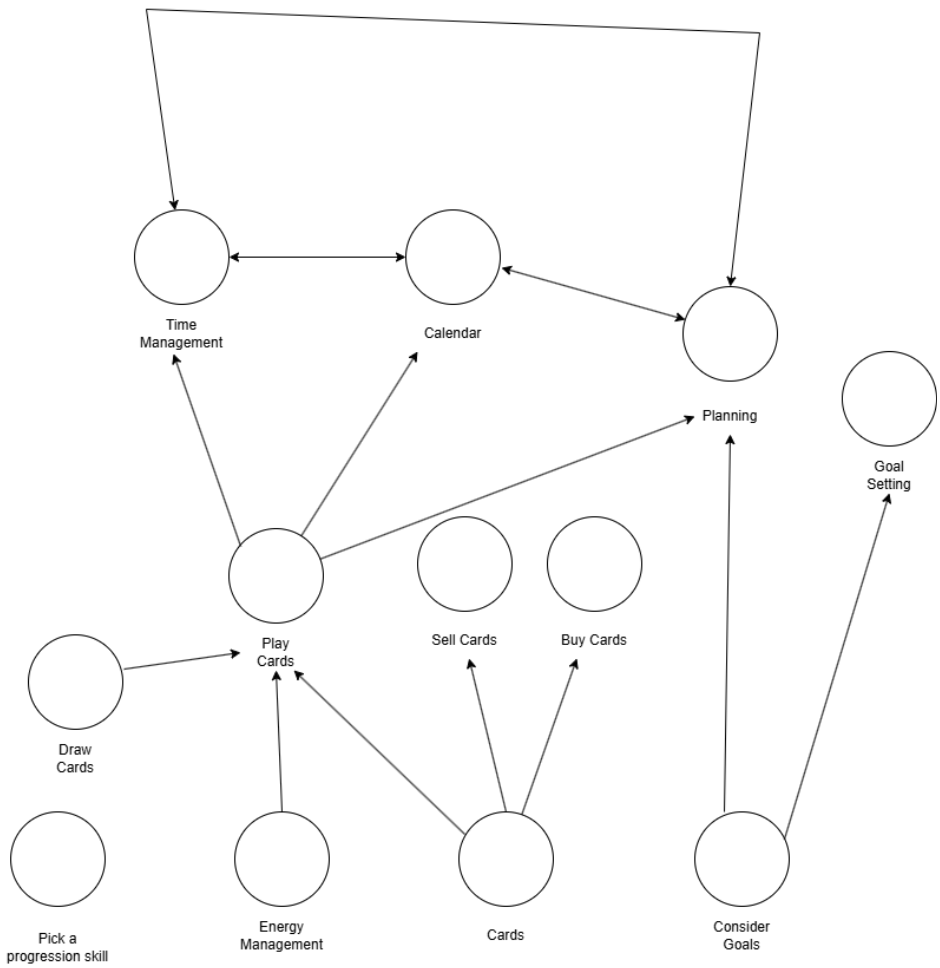

Figure 3.1: Example of Schema Graph.

On the Subjectivity of Schema Graph Design

It is important to emphasize that constructing a Schema Graph is not a mechanical process. The selection of connections—both vertical and horizontal—is inherently **subjective** and **creative**. While VIOLETA provides a structured approach to organizing gameplay and learning progression, it **does not prescribe a fixed schema layout**. Instead, the SG acts as a **canvas for interpretation**. The strength of the framework lies in how it balances rigor with expressiveness:

- A skilled instructor will use the SG to better align real-life skills with meaningful game behaviors.
- A thoughtful designer will identify elegant relationships between mechanics that enhance both learning and play.

Better educators will recognize which dependencies reflect true skill acquisition. Better designers will argue convincingly for how specific mechanics reinforce each other. In this sense, the SG is not only a design artifact—it is also **a statement of educational and creative intent**. By allowing this flexibility, the VIOLETA framework affirms its foundational philosophy: **Game design is not the removal of human judgment—it is the amplification of it through useful tools**.

3.6.11 Step 11: Constructing the Schema Adjacency Matrix (SAM)

Once the Schema Graph (SG) has been established, the next step is to translate its structure into a **Schema Adjacency Matrix (SAM)**. The SAM is a square matrix that formalizes the directional relationships between schemas, making the network easier to quantify, analyze, and later transform into STACK tiers.

Each row in the SAM represents a **source schema** (i.e., where a directed connection begins), and each column represents a **target schema** (i.e., where the connection ends). A value of 1 indicates that there is a directed edge from the row's schema to the column's schema. A 0 indicates no connection.

To support deeper analysis, we include:

- A final column labeled **Total Outgoing**: the number of schemas that each row-schema directly enables.

- A final row labeled **Total Incoming**: the number of schemas that point into each column-schema.

This matrix helps us determine:

- **Foundational schemas:** high outgoing, low incoming.
- **Integrative or strategic schemas:** high incoming and outgoing.
- **Capstone or dependent schemas:** high incoming, low outgoing.

Example: First Row and Column

Let us demonstrate this using two specific cells:

- **Row: Draw Cards**
  From the Schema Graph (SG), *Draw Cards* connects only to *Play Cards*. Therefore, the row for Draw Cards contains a **1** in the *Play Cards* column, and **0**s elsewhere.
  Total Outgoing: 1
- **Column: Play Cards**
  Play Cards has incoming edges from *Draw Cards*, *Cards*, and *Energy Management*. This means that the *Play Cards* column contains a **1** in the rows for those schemas.
  Total Incoming: 3

Schema	**Draw**	**Play**	**Sell**	**Buy**	**Progression**	**Goals**	**Task Mng.**	**Calendar**	**Planning**	**Energy**	**Goal Set**	**Cards**	**Total**
Draw Cards	–	1	0	0	0	0	0	0	0	0	0	0	1
Play Cards	0	–	0	0	0	0	1	1	1	0	0	0	3
Sell Cards	0	0	–	0	0	0	0	0	0	0	0	0	0
Buy Cards	0	0	0	–	0	0	0	0	0	0	0	0	0
Pick Progression Skill	0	0	0	0	–	0	0	0	0	0	0	0	0
Consider Goals	0	0	0	0	0	–	0	0	1	0	1	0	2
Task Management	0	0	0	0	0	0	–	1	1	0	0	0	2
Calendar	0	0	0	0	0	0	1	–	1	0	0	0	2
Planning	0	0	0	0	0	0	1	1	–	0	0	0	2
Energy Management	0	1	0	0	0	0	0	0	0	–	0	0	1
Goal Setting	0	0	0	0	0	0	0	0	0	0	–	0	0
Cards	0	1	1	1	0	0	0	0	0	0	0	–	3
Total Incoming	0	3	1	1	0	0	3	3	4	0	1	0	

Table 3.1: Interaction matrix of schema elements across gameplay phases.

3.6.12 Step 12: Estimating Cognitive Load

Schema Adjacency Matrix (SAM) completed, we now proceed to assess the **Intrinsic Cognitive Load (ICL)** of each schema. While the SAM captures how interconnected a schema is, the ICL attempts to answer a more direct question:

"How mentally demanding is this schema to learn or execute in isolation?"

Methodology

Cognitive Load Theory (CLT) distinguishes between different types of cognitive load, of which **Intrinsic Load** is the most structurally grounded. It is primarily determined by the **element interactivity** inherent in a learning task—i.e., how many information elements must be held and processed simultaneously in working memory (Sweller et al., 1998).

However, despite its theoretical clarity, intrinsic cognitive load **is not empirically fixed or measurable with precision**. In practice, the cognitive burden of a schema depends on numerous fluid factors:

- How familiar the learner is with similar mechanics.
- How well its components have been internalized as schemas.
- How many of its subcomponents feel novel or automatic.

This uncertainty becomes especially relevant in **games**, where schemas can be **rich, layered, and composed of multiple sub-schemas**. For example, the action "Sell Cards" may involve filtering, valuation, and discard logic. To a veteran player, this might feel like a single schema—already chunked and automated. To a first-time player, it may feel like **three interwoven decisions**, each requiring conscious attention.

Because of this inherent variability, the VIOLETA framework treats ICL estimation not as a rigid calculation but as **a creative and interpretative step**. It is a moment where the **instructor's insight and heuristics** come into play. By making an educated assumption about how complex a schema will feel to the player—especially at first exposure—we can calibrate a **"ground-level" cognitive estimate** to guide tier construction.

Creative Estimation of Load

To support instructors in this task, VIOLETA suggests a primary heuristic: **Use the number of meaningful choices as a proxy for intrinsic cognitive load**. For example, take the schema Play Cards:

- At the beginning of the game, a player draws 2 cards and must assign them to 2 available time slots.
- One way to frame this is: the player only has to decide which card goes first—because the second card automatically fills the remaining slot. Under this model, only **one meaningful decision** is made, yielding an ICL of **1**.
- Alternatively, a player may evaluate all possible permutations—e.g., "What if Card A goes at 9 AM? What if Card B goes at 9 AM?"—resulting in **two comparative evaluations** and an ICL closer to **2**.

Some designers may even argue for **2 $\times$ 2 = 4 combinations**, depending on how deep the analysis goes. All of these are valid perspectives. What matters is that:

- The instructor **reflects on how the decision is likely experienced**,
- **Articulates their reasoning**, and
- **Assigns a value** that reasonably reflects the expected working memory load for a novice player.

Acknowledging Schema Partiality

It is also important to note that **schemas may be partially known** to the player already. While we treat each schema in the table as a discrete entity, real learners do not arrive with tabula rasa minds. Many will recognize elements from previous games, learning contexts, or even life experiences. Because we cannot assess prior knowledge with certainty, we must assume a cautious baseline:

- Assume minimal internalization.
- Use ICL to define **a starting point** where the schema is challenging but not overwhelming.

This value will later help us:

- Establish entry conditions for STACK Tier 1.
- Create a difficulty slope where low-performing players remain engaged and high-performing players are quickly moved to novel content.

Final Thought

In sum, the intrinsic load of a schema is not a fixed property—it is **a situational estimate**. VIOLETA's approach empowers instructors to make these estimates with confidence and clarity. Rather than pretending precision where none exists, it encourages transparent, reasoned subjectivity. Future research may refine this process and yield more standardized ICL metrics, but until then, the creativity, judgment, and pedagogical insight of the instructor **remain essential**.

Intrinsic Load Table (ICL Table)

The following table presents a subjective evaluation of intrinsic cognitive load for each schema in our game prototype:

Schema	**Description**	**Intrinsic Load**
Play Cards	2 comparative choices: Player evaluates card A in slot 1 vs. card B in slot 1 before finalizing. Two comparative evaluations.	2
Sell Cards	Choose 1 card from 126, but cards are independent; decision space is wide but not complex.	1
Buy Cards	Depends on how many cards are offered to the player.	X
Cards	Each card encodes time and energy cost + productivity value. Since players are unaware of the energy cost until they unlock the skill, it's 1 + 1 = 2.	2
Consider Goals	Player sees 5 goals, but they're independent—no interaction needed between them.	1
Draw Cards	No decision to be made; only awareness of how many are drawn.	1
Pick a Progression Skill	Depends on how many skills we offer the player to choose from.	Y
Task Management	Adds 3 cards and 3 hours of planning. Not all cards are different, and card-playing mechanic already introduced. Reasonable estimate: 4.	4
Calendar	Adds 1 card and enables playing 1 card into the future. 1 + 1 interacting elements.	2
Planning	Like Task Management but adds visibility into future goals. Equivalent decision weight.	4
Energy Management	Adds awareness of energy as a constraint. One new variable to consider in card play.	1
Total		**18 + X + Y**

Table 3.2: Estimated intrinsic cognitive load for schema elements in the prototype game.

Variable Intrinsic Loads and Schema Parameterizability

In the ICL table, certain schemas—such as Buy Cards and Pick a Progression Skill—have been assigned **variable loads**, marked by placeholders like **X** and **Y**. This is not a gap, but a **deliberate affordance** within the VIOLETA framework, designed to enable future tuning. Some schemas are **inherently parameterizable**—they are built around offering players a configurable number of options or layers. For example:

- In Buy Cards, the designer can control how many cards are presented for purchase at once (e.g., 3, 5, or 7).
- In Pick a Progression Skill, the number of skill upgrades offered may vary depending on game phase, difficulty level, or player performance.

This parameterizability enables these schemas to act as **adaptive levers** during STACK tier construction. By increasing or decreasing the number of choices, the designer directly influences **intrinsic cognitive load**. This makes it easier to ensure that the combined working memory load of active schemas remains close to the cognitive threshold (often estimated around 7 elements).

However, this kind of flexibility is **not universal**. Many schemas—like Draw Cards, Play Cards, or Energy Management—have fixed structures that do not lend themselves to runtime tuning. Their intrinsic load depends on the nature of the mechanic itself and cannot be easily adjusted without changing core gameplay dynamics.

Therefore, the parameterization of ICL is schema-specific: only those mechanics that inherently allow for modifying the quantity of choices can be used as "difficulty knobs."

This distinction is critical when designing STACK tiers. It ensures that:

- Instructors do not mistakenly expect all mechanics to be adjustable.
- Tuning remains focused and intentional, targeting the right type of schemas.

In summary, variable-load schemas add necessary flexibility, but that flexibility is **bounded** by the schema's internal design. VIOLETA highlights these opportunities but does not assume they are universally available.

Calculating Element Interactivity Load (EIL)

Once the intrinsic load of each schema is estimated, we must account for a second dimension of cognitive burden: **element interactivity**. Even schemas with low intrinsic load can generate significant cognitive strain when they interact with each other in complex ways. This is where the **Element Interactivity Matrix (EIM)** comes into play.

Each EIM is tied to a specific game phase, as defined in the Time Table (TT). In our case, we identified two distinct phases:

- Play Phase
- Upkeep Phase

Accordingly, we create one EIM for each phase. If a game has no such division, a single matrix is used.

Constructing the Element Interactivity Matrix (EIM)

The EIM is a symmetric matrix that encodes whether two schemas **interact cognitively** during the same phase. It helps estimate the Element Interactivity Load (EIL) for that phase.

Construction Steps

1. List all relevant schemas for the selected phase as both rows and columns.
2. Fill the diagonal with − (a schema doesn't interact with itself).
3. Black out all cells **below the diagonal** to avoid double-counting.
4. For each remaining cell, insert:
   - **1** if the two schemas interact and increase mental workload together.
   - **0** if they are independent.
5. In the Total Interactivity column, sum each row's interactions.
6. In the Total row and Total Interactivity column, sum all interactions to produce the **Element Interactivity Load (EIL)** for that phase.

Only the final total matters for the **Cognitive Load Matrix (CLM)**—not which schema initiated or received the interaction.

Partial Example: EIM for Upkeep Phase (STACK 1)

EIM Upkeep STACK 1	Buy Cards	Pick Prog. Skill	Consider Goals	Goal Setting	Total Interactivity
Buy Cards	–	0	1	0	1
Pick Prog. Skill	■	–	1	0	1
Consider Goals	■	■	–	1	1
Goal Setting	■	■	■	–	0
Total					**3**

Table 3.3: Element Interactivity Matrix (EIM) — Upkeep Phase, STACK Tier 1

Note: This is ***a partial example*** *constructed for clarity. A complete EIM for STACK 1 would include all active schemas, but this version demonstrates the process while remaining readable.*

Take the example of Buy Cards. On the surface, this might seem like a simple act of choosing one card from several options. However, when a player is also tasked with pursuing specific goals—such as improving health, advancing their startup, or maintaining relationships—every potential card must be evaluated **in the context of those goals**. For instance, a "Workout" card may be effective for physical health but consumes time and energy that could have been used for study or project tasks. This creates **a cross-schema evaluation problem**: the player is not just choosing the "best" card, but the card that **best aligns with their strategic intent**.

Because of this dependency, the schemas Buy Cards and Consider Goals cannot be cognitively isolated. They **interact directly** in the player's decision-making process, and thus contribute jointly to working memory load. As such, we assign a 1 in the matrix to indicate this interactivity.

Partial Example: EIM for Play Phase (STACK 1)

EIM Play STACK 1	Consider Goals	Task Mgmt	Calendar	Planning	Energy Mgmt	Total Interactivity
Consider Goals	–	1	1	1	1	4
Task Mgmt	■	–	1	1	1	3
Calendar	■	■	–	1	1	2
Planning	■	■	■	–	1	1
Energy Mgmt	■	■	■	■	–	0
Total						**10**

Table 3.4: Element Interactivity Matrix (EIM) — Play Phase, STACK Tier 1

Note: This is ***a partial example*** *constructed for clarity. A complete EIM for STACK 1 would include all active schemas, but this version demonstrates the process while remaining readable.*

On its own, Task Management increases the number of cards a player can draw and play—but its usefulness is **directly shaped** by how well the player can schedule those actions across time.

This is where the Calendar schema becomes crucial. The Calendar mechanic allows players to schedule cards for future turns, effectively extending the planning horizon. As such, the player must consider not only which tasks to manage, but **when** and **in which future context** those tasks will execute. This introduces a second layer of decision complexity.

Further, Planning adds an additional dimension by providing foresight into upcoming goals, which in turn influences how tasks and schedules are prioritized. Finally, **Energy Management** imposes constraints on when certain tasks can be optimally performed based on energy levels, further complicating the scheduling logic introduced by Calendar and Task Management.

Because these schemas—Task Management, Calendar, Planning, and Energy Management—all influence each other's effectiveness, we assign interactivity values accordingly. For example, Task Management has direct cognitive interdependencies with Calendar (when should tasks be scheduled?), with Planning (what's coming next?), and with Energy Management (will I have the capacity to complete this task effectively?). These dependencies raise Task Management's total interactivity score to **3**.

Why EIM Matters

Cognitive Load Theory stresses that learners can only handle **a limited number of interacting elements** in working memory before overload occurs. The Intrinsic Load of individual schemas tells us how heavy they are in isolation—but the Element Interactivity Load tells us how much heavier they become when used together.

VIOLETA incorporates both to calculate the **Cognitive Load Matrix (CLM)** in the next step, ensuring that the design:

- Stays within working memory constraints,
- Offers just the right challenge,
- And scaffolds schema acquisition across STACK tiers.

This step also serves as an opportunity for instructors to apply their pedagogical and domain-specific intuition by thoughtfully identifying which schemas are cognitively interconnected. Better instructors—those with deeper understanding of both the subject matter and player psychology—will be able to construct more accurate and effective interactivity mappings, resulting in a more balanced and impactful gameplay experience.

3.6.13 Step 13: Constructing the Cognitive Load Matrix (CLM)

With the Intrinsic Cognitive Load (ICL) and Element Interactivity Matrices (EIMs) completed, we can now unify all cognitive demands across phases using the **Cognitive Load Matrix (CLM)**. This matrix allows designers and instructors to see—at a glance—**whether each game phase within a STACK tier is cognitively manageable** for players.

How to Build the CLM

To construct the CLM, follow these steps:

1. Create a column for each schema, plus three additional columns:
    - Phase Name
    - EIMs (for the Element Interactivity Load)
    - TOTAL LOAD (sum of intrinsic and interactivity loads)
2. Create a row for each game phase defined in the Time Table (TT).
3. Fill in each schema's intrinsic load if it is active in that phase; use 0 if inactive.
4. Insert the EIM total for that phase.
5. Compute the TOTAL LOAD by summing across all schema values and the EIM total.

The result is a quantitative snapshot of **how cognitively demanding** each game phase is, making it easy to check whether phases fall near or far from the working memory threshold (typically 7 elements for novice learners).

Phase Name	Task Mgmt	Calendar	Planning	Energy Mgmt	Buy Cards	Consider Goals	Goal Setting	Pick Progression Skill	EIMs	TOTAL LOAD
Upkeep	0	0	0	0	X	1	1	Y	3	$5+X+Y$
Play	4	2	4	1	0	1	0	0	10	22

Table 3.5: Cognitive Load Matrix (CLM) — Example Values for STACK Tier 1

Partial Example: CLM for STACK Tier 1

Note: This is ***a partial example*** *constructed for clarity. A complete CLM for STACK 1 would include all active schemas, but this version demonstrates the process while remaining readable.*

Interpreting the Results

The CLM immediately reveals phase-level cognitive pressure:

- The Upkeep Phase results in a TOTAL LOAD of 5 + X + Y. This is useful, because X and Y are parameterizable (i.e., number of options for Buy Cards and Pick a Skill), which allows the designer to **tune the phase to stay within working memory limits**. If we set X = 1 and Y = 1, the total becomes 7, which is ideal for new players.
- In contrast, the Play Phase has a fixed TOTAL LOAD of 22, far above the working memory threshold. This indicates a clear overload, likely to result in frustration, reduced performance, or poor skill acquisition—unless the player is already highly experienced.

This reveals the CLM's critical value: it enables **design validation**. The designer can now make evidence-based decisions about which schemas to retain, remove, or delay for later STACK tiers.

Iteration: From Overload to Balanced STACK Tier

Constructing the CLM marks the end of one full design iteration. It gives the designer visibility into which phases are feasible and which are overloaded. From here, the process becomes **iterative**:

1. Use the CLM to **identify overloaded phases**.
2. Move or delay the most cognitively expensive schemas.
3. Rebuild the EIMs for the reduced set of schemas.
4. Update the CLM and check if the phase total is near or below the cognitive threshold (7).

5. Repeat as needed.

Once a phase falls within acceptable cognitive load limits—without compromising skill relevance—you have successfully defined a STACK Tier. Each tier introduces just enough new complexity to promote learning, while maintaining the player's sense of competence and engagement.

Refining the STACK Tier Using the Schema Adjacency Matrix

After completing the Cognitive Load Matrix (CLM), it often becomes clear that certain phases or tiers exceed the cognitive limits suitable for a particular target audience. At this point, the designer must **reduce the number of active schemas** to bring the total load closer to the working memory threshold—typically around 7 elements. To support this process, the VIOLETA framework recommends leveraging the Schema Adjacency Matrix (SAM). This matrix reveals how schemas are interdependent and helps identify which elements are **core, foundational, or peripheral**.

Start with Base Elements

Schema	Draw Cards	Play Cards	Sell Cards	Buy Cards	Pick a progression skill	Consider Goals	Task Management	Calendar	Planning	Energy Management	Goal Setting	Cards	Total Outgoing
Draw Cards	–	1	0	0	0	0	0	0	0	0	0	0	1
Play Cards	0	–	0	0	0	0	1	1	1	0	0	0	4
Sell Cards	0	0	–	0	0	0	0	0	0	0	0	0	0
Buy Cards	0	0	0	–	0	0	0	0	0	0	0	0	0
Pick a progression skill	0	0	0	0	–	0	0	0	0	0	0	0	0
Consider Goals	0	0	0	0	0	–	0	0	1	0	1	0	2
Task Management	0	0	0	0	0	0	–	1	1	0	0	0	2
Calendar	0	0	0	0	0	0	1	–	1	0	0	0	2
Planning	0	0	0	0	0	0	1	1	–	0	0	0	2
Energy Management	0	1	0	0	0	0	0	0	0	–	0	0	1
Goal Setting	0	0	0	0	0	0	0	0	0	0	–	0	0
Cards	0	1	1	1	0	0	0	0	0	0	0	–	3
Total Incoming	0	3	1	1	0	0	3	3	4	0	1	0	

Table 3.6: Schema Adjacency Matrix (SAM) — Time Management Prototype

Schemas with **zero incoming connections** in the SAM are considered **base elements**. These are mechanics that do not rely on any prior schema for understanding, making them ideal candidates for inclusion in early STACK Tiers. These schemas can be introduced with minimal scaffolding, as they form the cognitive ground floor of gameplay.
In the SAM for STACK Tier 1, we identified the following base elements:

- Cards
- Draw Cards
- Pick a Progression Skill
- Consider Goals

- Energy Management

While these provide a minimal cognitive foundation, they are likely not sufficient to reach the desired complexity for meaningful gameplay. Their combined load is likely too low to engage even novice players meaningfully.

Expand with High-Influence Schemas

To enrich the STACK Tier without overwhelming the player, the next step is to introduce **schemas with high outdegree**—i.e., schemas that have many **outgoing connections** in the SAM. These are often **mechanically integral** to the rest of the game; they “unlock” or “enable” many other features and systems.

In our SAM, the most influential schema by this measure is:

- Play Cards (3 outgoing connections)

Because Play Cards is a core mechanic around which many others revolve, it is a natural candidate to include in early gameplay—even if its intrinsic load is moderate to high. Introducing it early helps ensure that other schemas introduced later will be more easily understood and integrated.

Consider What’s Already Working

Recall from the CLM that the Upkeep Phase was already within acceptable load limits. This gives us the freedom to retain or even slightly expand that phase without exceeding the working memory threshold. As such, we can safely reintroduce other Upkeep-related schemas, such as:

- Sell Cards

- Buy Cards

These mechanics introduce controlled decision-making opportunities and help round out the player’s first experience with card economy, without placing too much burden on planning or future-state reasoning.

Revised Schema Set for STACK Tier 1

Based on this reasoning, the refined schema set for STACK Tier 1 is:

- Play Cards

- Draw Cards
- Sell Cards
- Buy Cards
- Pick a Progression Skill
- Consider Goals
- Energy Management
- Cards

This collection strikes a balance between simplicity and functional richness. It includes foundational schemas, introduces at least one high-connectivity mechanic, and fills out both Play and Upkeep phases with meaningful but manageable interactions.

EIM Upkeep STACK 1	**Buy Cards**	**Pick a Progression Skill**	**Consider Goals**	**Goal Setting**	**Total Interactivity**
Buy Cards	–	0	1	0	1
Pick a Progression Skill	■	–	1	0	1
Consider Goals	■	■	–	1	1
Goal Setting	■	■	■	–	0
Total					**3**

Table 3.7: Element Interactivity Matrix (EIM) — Upkeep Phase, STACK Tier 1

EIM Play STACK 1	**Draw Cards**	**Consider Goals**	**Cards**	**Energy Management**	**Total Interactivity**
Play Cards	0	1	1	1	3
Draw Cards	■	0	0	0	0
Consider Goals	■	■	0	0	0
Cards	■	■	■	1	1
Energy Management	■	■	■	■	0
Total					**4**

Table 3.8: Element Interactivity Matrix (EIM) — Play Phase, STACK Tier 1

Phase Name	**Play Cards**	**Draw Cards**	**Cards**	**Energy Mgmt**	**Buy Cards**	**Consider Goals**	**Sell Cards**	**Pick Progression Skill**	**EIMs**	**TOTAL LOAD**
Upkeep	0	0	0	0	X	1	1	Y	3	$5 + X + Y$
Play	2	1	2	1	0	1	0	0	4	11

Table 3.9: Cognitive Load Matrix (CLM) — Full Schema Overview for STACK Tier 1

Re-iteration: Adjusting Schema Sets to Achieve Cognitive Balance

The CLM revealed that the Play Phase still exceeds the target cognitive load. With a total load of 11, it remains above the commonly cited working memory threshold of 7 elements. To bring it within acceptable bounds, we must iterate further.

One effective adjustment is to remove the Energy Management schema:

- This reduces Intrinsic Load by **1**.
- It also eliminates **2 interactivity points** in the EIM, as Energy Management was involved in multiple schema connections.

This single change brings the Play Phase's total cognitive load from 11 down to 8, which is much more reasonable—especially for early STACK tiers targeting novice players. Importantly, this adjustment **aligns perfectly with an earlier design decision**: the energy system continues to function **in the background**, affecting outcomes without requiring player awareness or decision-making in the initial stages. This preserves systemic realism while delaying cognitive engagement with the energy mechanic until later tiers, when the player has more working memory resources and contextual understanding to handle it.

Crucially, removing Energy Management at this stage does not compromise the integrity of the core loop—*Play Cards, Draw Cards, Consider Goals, and Cards* remain intact, preserving the fundamental rhythm of planning and execution.

On the 7-Element Anchor: A Working Heuristic

The traditional limit of 7±2 elements in working memory has served as a guiding principle for cognitive design. However, the exact number is:

- Empirically variable, depending on the player's familiarity with the material,
- Influenced by schema chunking and prior knowledge,
- And affected by attention, fatigue, and engagement levels.

Because Intrinsic Load values and EIM scores in VIOLETA are also based on heuristics and approximations, the threshold of "7" should be seen as a **design anchor**, not a hard boundary. In some cases, raising the working memory ceiling to 8 or 9 elements may better reflect real player tolerance.

Flexibility and Future Research

This highlights another area where the VIOLETA framework invites future work:

- Through case studies and empirical testing, it may be possible to refine the cognitive anchor used for STACK Tier design.
- Different genres, age groups, and subject matters may benefit from different anchor points.
- VIOLETA's flexibility ensures that instructors and researchers can adapt it to their evolving understanding of cognitive dynamics.

By blending structured reasoning with creative heuristics, VIOLETA encourages iterative design that remains responsive to both theory and practice.

Preparing for the Next STACK Tier

With the first STACK Tier now finalized, we prepare to design the next tier in the sequence. The core purpose of the STACK system is to ensure that **schemas are not just introduced—but also mastered—before more complexity is added**. Each tier thus serves as a structured layer of cognitive scaffolding, preventing overload and promoting schema automation.

Identifying Mastered Schemas

Before designing the next tier, we must evaluate which schemas from the current CLM are considered "mastered." This decision is **subjective** and must be made based on:

- The nature of the schema,
- The intensity and frequency of its use in the current tier,
- The extent to which its cognitive burden is likely to decrease over time.

Consider the following example from STACK Tier 1:

- **Play Cards:** While this mechanic becomes more familiar with practice, it always involves active decision-making. The cognitive load associated with evaluating card-slot combinations is persistent. Therefore, we do not mark this schema as mastered, and it will remain in the next STACK tier.

- **Draw Cards:** This mechanic is passive and simple. No decision-making is required, and the player's understanding stabilizes early. We can confidently say this schema has been mastered, and it will be removed from further ICL and SAM considerations.
- **Cards:** As the foundational component of the game, cards will continue to play a central role. However, given the extended exposure in Tier 1, the intrinsic load of the schema can be reduced (e.g., from 2 to 1) to reflect partial automation.
- **Consider Goals:** Although players become more comfortable tracking goals, this schema involves ongoing juggling of priorities. Because this information must be repeatedly held in working memory, it is retained in the next tier.

This schema-level reasoning is again a point where the instructor's insight plays a critical role. Better instructors will make more nuanced decisions, adjusting their approach based on domain knowledge, playtesting, and player profiles.

Updating the Design Process

Once mastery decisions are made, the following updates must occur:

- Remove fully mastered schemas from:
  - The **ICL table** (they no longer contribute to intrinsic load),
  - The **SAM** (they no longer affect adjacency relationships),
  - Any **EIM or CLM** constructed for the next tier.
- Update intrinsic load values for **partially mastered schemas** in the ICL (e.g., reduce from 2 to 1).

This ensures that cognitive calculations remain accurate and reflective of actual player capacity.

Begin the Next Iteration

With updated cognitive assets (SAM, ICL), the VIOLETA process begins again. Designers:

1. Reassess which schemas to introduce next based on updated SAM (prefer schemas with low incoming edges or high outgoing edges).

2. Rebuild EIMs for each phase based on the new schema set.

3. Construct a new CLM and check total loads.

4. Iterate until cognitive load per phase stabilizes near the working memory threshold.

This cycle repeats until **all game schemas have been introduced and scaffolded** through one of the STACK tiers. At that point, the STACK process is complete. The player has been brought from foundational understanding to full-system fluency through a sequence of optimally loaded cognitive steps.

Beyond STACK: The 3E Process and Future Directions

While the core of this thesis focuses on the construction and iteration of STACK Tiers, it is important to briefly introduce the intended usage model for these tiers—particularly how they are meant to be experienced by the player and evaluated by the system. This is where the **3E process—Exposure, Experiment, Evaluate**—comes into play.

Although 3E lies outside the formal scope of this bachelor thesis, it represents the **natural continuation** of the STACK concept. Once a STACK Tier is defined and implemented, the player must move through a carefully designed experience cycle that:

1. **Exposes** them to the relevant schemas,

2. Encourages them to **experiment** with choices, and

3. Provides meaningful feedback to **evaluate** whether mastery has occurred.

For example, take the atomic unit skill of **batching similar tasks**—a principle rooted in time management. In a video game context, a player's decisions could be tracked programmatically: are they grouping similar actions together to minimize context switching? In a board game context, the same could be evaluated more simply—e.g., through level design where sufficient points can only be earned **if tasks are batched**, thus encouraging the correct behavior without the need for explicit instruction.

Importantly, VIOLETA does **not require players to "fail" in order to learn**. While repeated failure is a valid design choice in some contexts (especially when **frustration** is a target emotion), the framework instead encourages flow-preserving systems that enable mastery through gradual improvement. For instance:

- Procedurally generated levels with slight variations can reinforce the same schema without creating a feeling of repetition.
- In board games, randomized tile draws or shuffled decks can achieve the same effect.

This design pattern also leverages psychological principles like the Near Win effect, discussed in Addiction by Design (Schüll, 2012), which shows how players are more likely to remain engaged if they feel they are *almost succeeding*. Applied properly, this can help maintain motivation without compromising learning outcomes.
Ultimately, the most critical principle of the STACK system is:

No schema from a higher STACK Tier should be introduced until the current tier's schemas have been demonstrably mastered.

This guarantees that each layer of complexity rests on a solid cognitive foundation, ensuring that learning is **progressive, integrated, and retention-oriented**. Future work could expand on this by building automated systems to monitor schema mastery, integrating adaptive content delivery, and refining mastery thresholds across player profiles.

Beyond Mastery: Introducing SPARK

The final step in the VIOLETA framework—though beyond the scope of this bachelor thesis—is the **SPARK** phase, short for **System for Player Action, Reward, and Knowledge**. While STACK ensures that gameplay content is delivered in a cognitively aligned sequence, SPARK focuses on the moment-to-moment experience of play—specifically, how player actions are reinforced and perceived. SPARK is designed to enhance the core gameplay feel by drawing on standardized game design principles that reward correct actions through sensory feedback and player-driven feedback loops. The idea is to create gameplay that not only teaches but also feels satisfying to interact with.

In many popular genres—such as hack-and-slash or action RPGs—players often describe the "quality" of gameplay in terms of **impact feedback**: for example, does hitting an enemy produce a satisfying sound? Does the animation convey weight and force? These effects, while seemingly cosmetic, serve a critical purpose: they **reward the player's action** and reinforce the decision loop. SPARK seeks to standardize such techniques in the context of educational games. In the context of our time management game, a simple SPARK implementation might include:

- A distinct sound cue when the player successfully batches similar tasks together.
- A visual animation that reinforces that this was an efficient or productive decision.
- A recurring "flow meter" or momentum bar that builds as effective actions accumulate.

Such feedback loops can create **positive emotional associations** with correct decision-making, helping the player internalize real-world skills through **implicit reinforcement** rather than only extrinsic rewards like points or badges.

That said, the impact and effectiveness of these methods are not yet empirically validated within educational gameplay contexts. As such, SPARK remains an experimental concept, and its future development represents a promising avenue for further research—particularly in the intersection of cognitive design and game feel. It is anticipated that future work will explore how to measure the impact of sensory reward systems on long-term schema retention and player motivation.

Chapter 4

Evaluation of the Framework

This chapter consolidates the evaluations embedded throughout the previous design walkthrough and presents a comparative assessment of the VIOLETA framework. Rather than introducing new material, it offers a higher-level synthesis of the design logic and structural innovations already articulated, while contrasting them with other educational game design methodologies.

4.1 Synthesized Strengths and Principles

From its inception, the VIOLETA framework was built to address a key educational design gap: the lack of explicit and actionable linkage between gameplay and real-world skill acquisition. The strengths of VIOLETA are not isolated features but stem from a cohesive architectural philosophy. Key strengths include:

- **Triadic Integration Structure:** The systematic mapping between atomic unit skills, emotional trajectories, and gameplay mechanics—operationalized through tools such as the Scaling Influence Table (SIT) and the Triadic Integration Table (TIT)—ensures that gameplay elements are not arbitrarily designed but emerge from pedagogically meaningful relationships. This tight integration strengthens transferability and maintains thematic coherence.

- **Emotional Anchoring as a Design Lever:** Emotional experiences (e.g., "gradual control amidst chaos") are structurally embedded using the Base Mechanics Tree (BMT). Emotions guide both game pacing and mechanic selection, ensuring that gameplay evokes emotionally resonant experiences that support learning.

- **Cognitive Load-Aware Progression:** STACK (Structured Tiered Adaptive Core Knowledge) introduces schema-based adaptive progression that respects Cognitive Load Theory, preventing overload and scaffolding learning through tiered complexity.
- **Balance of Freedom and Structure:** VIOLETA's methodology leverages System 1 (intuitive design) and System 2 (analytical structuring) thinking. This creates a workflow that is both expressive and reproducible, particularly when compared to overly prescriptive or purely abstract frameworks.
- **Open Mechanic Philosophy:** VIOLETA does not prescribe a fixed list of valid mechanics. Instead, it encourages designers to invent or adapt mechanics, as long as they logically align with emotional states and skill mappings. This flexibility preserves originality and domain fit.
- **Front-Loaded Design, Streamlined Development:** While VIOLETA emphasizes an intensive design phase, it delivers a finished, logic-consistent design blueprint. This reduces downstream development costs by minimizing confusion, rework, and vague design specs. The resulting clarity can significantly shorten the prototyping and implementation timeline.
- **Guided in Learning Science:** Built upon well-established theories such as Self-Determination Theory and Cognitive Load Theory, giving it a strong pedagogical foundation.
- **Step-by-Step Design Process:** Designers are never left guessing. From atomic unit selection to MVP and difficulty calibration, each step builds on the previous with logical progression.
- **Modular Outputs:** Components like the SIT, TIT, BMT, CLM, and SAM can be reused, adapted, or extended—ideal for large educational portfolios or sequels.
- **Challenges the "Fun vs. Learning" Dichotomy:** Rather than treating engagement and education as trade-offs, VIOLETA blends them through design. It ensures that what is fun is also educationally purposeful, and what is educational feels meaningful and rewarding.
- **Framework Agnostic to Medium:** The framework can be applied to board games, card games, digital games, or hybrids. This medium-agnostic approach increases its versatility and reach across different

production environments. VIOLETA does not enforce any technological constraint.

- **Prepares for Tool-Based Future:** Its systematic use of matrices and tables makes it ideally suited for future digital tooling. Potential extensions include AI-assisted design tools, editor interfaces, or dynamic validation systems, which could dramatically lower entry barriers.
- **Potential for Long-Term Impact:** By bridging educational theory and creative production, VIOLETA has the potential to redefine collaboration between educators and designers. If scaled, it could serve as a common language or standard for educational game design.

4.2 Limitations and Design Trade-offs

VIOLETA is not without limitations. Several trade-offs and constraints emerged during the construction of the walkthrough and prototype:

- **Designer Cognitive Load:** The recursive and multi-step nature of the framework can be cognitively demanding, especially for novice designers. Without dedicated tools or templates, designers may feel overwhelmed.
- **Subjectivity in Mapping:** Emotional and mechanical mappings (e.g., which mechanics evoke "progress" or "autonomy") rely on designer interpretation. This allows creative freedom but can lead to inconsistency if applied carelessly.
- **Tooling Dependence for Practicality:** Tables such as the TIT, SAM, CLM, and EIM are immensely helpful but hard to manage manually. The practical implementation of VIOLETA may require digital support tools.
- **Requires Strong Abstract Reasoning:** Designers must be comfortable abstracting real-life skills, linking them to emotions, and then converting those into mechanics.
- **No Final Game Guarantee:** VIOLETA delivers a structured design, not a complete game. The quality of implementation—UX design, narrative polish, rule clarity—still depends on development expertise outside the framework's scope, which could affect learning efficacy.

- **Weak Support for Collaborative Design:** VIOLETA is optimized for a single lead designer or author. It lacks scaffolding for distributed teams (e.g., dividing tasks across narrative, mechanics, pedagogy).
- **Lack of Digital Tool Support:** Managing tables and matrices manually is time-consuming and error-prone. Without software support, designers may struggle to maintain consistency.
- **Unvalidated Skill Transfer:** The principle "more play = more mastery" is theoretically justified but has not yet been empirically validated. Future research is needed to confirm its effectiveness across different skill domains and learner types.
- **Framework Outputs Are Not Easily Standardized:** Because of its open-ended nature, VIOLETA-designed games may be hard to compare, rate, or certify without a separate rubric or quality assurance layer.

4.3 Comparison with Existing Frameworks

Feature	UDL	Game Object Model	Let's Get On-Board	VIOLETA
Supports diverse learners	✓	✓	✓	✓
Connects gameplay with skills	×	✓(abstractly)	×	✓(explicitly)
Step-by-step mapping process	×	×	×	✓
Emotional anchoring	×	×	×	✓
Adaptive cognitive design	×	×	×	✓(via STACK)
Encourages designer creativity	✓	✓	✓	✓

Table 4.1: Framework feature comparison between UDL, Game Object Model, Let's Get On-Board, and VIOLETA.

Supports diverse learners: All four frameworks support learner diversity. UDL excels here, with a central focus on accessibility and flexible engagement. VIOLETA contributes through its adaptive STACK system, which accommodates varying learner progression rates by adjusting cognitive load and challenge.

Connects gameplay with skills: UDL and LGOB do not offer direct methods to embed educational skills into gameplay. GOM aims to align learning with gameplay conceptually but offers little in terms of concrete steps. VIOLETA explicitly maps skills to in-game actions using structured tools like SIT and TIT, enabling systematic integration.

Step-by-step mapping process: Only VIOLETA includes a detailed design workflow. UDL, GOM, and LGOB focus on goals and principles without prescribing how to transform ideas into game-ready mechanics.

Emotional anchoring: GOM touches on emotional tone via aesthetics, but does not operationalize emotion in the gameplay-learning relationship. VIOLETA uniquely formalizes emotional design, treating it as central to skill transfer and engagement.

Adaptive cognitive design: This is a unique strength of VIOLETA. None of the other frameworks dynamically regulate difficulty based on player cognitive load. VIOLETA's STACK structure explicitly scaffolds learning through schema-paced progression.

Encourages designer creativity: All frameworks leave space for creativity. VIOLETA distinguishes itself by structuring creativity—offering tools that preserve freedom while ensuring educational alignment.

4.4 Evaluation of Core Hypothesis: "More Play = More Mastery"

The central hypothesis of the VIOLETA framework is that extended engagement with the game should result in proportional increases in real-life skill mastery. This claim can be re-evaluated through three lenses:

Structural Integrity: The triadic mapping ensures that each skill is explicitly embedded into game actions. There is minimal opportunity for players to succeed in the game without exercising the intended competencies.

Progressive Complexity: The STACK system gradually introduces schemas based on mastery, ensuring cognitive pacing. This increases retention and reduces the likelihood of shallow learning.

Emotional Motivation: Emotional journeys are tied to skill progression, making the experience both rewarding and reflective of real-world growth. As players develop strategies that work within the game, they simultaneously internalize those strategies as life skills.

In conclusion, the hypothesis stands validated within the prototype scope and in the theoretical design space. The VIOLETA framework demonstrates that entertainment and learning can be structurally linked through its triadic integration of skills, emotions, and mechanics. Rather than relying on incidental learning or surface-level engagement, it constructs layered game systems where extended gameplay naturally supports deeper real-world skill development. However, while this theoretical foundation is strong, empirical research and playtesting are needed to validate whether these designed systems consistently lead to real-life mastery across diverse contexts and player profiles.

Chapter 5

Discussion

5.1 Reflections on Educational Game Design

The development of the VIOLETA framework has been both a response to observed deficiencies in existing educational game design models and a personal reflection on the transformative potential of gameplay. Traditional frameworks often struggle to translate abstract pedagogical goals into concrete, playable systems. VIOLETA emerged as a structured response to this gap, asserting that educational value can be systematically embedded into game systems through deliberate alignment of skills, emotions, and mechanics.

Unlike approaches that either oversimplify or overcomplicate game-based learning, VIOLETA embraces a middle path: it is rigorous yet intuitive, offering both design freedom and structural guidance. This balance is evident in the use of Triadic Integration (skills-emotions-mechanics), the Emotional Layering process, and STACK, the adaptive difficulty scaffolding. These tools transform educational game design from a domain of artistic guesswork into a repeatable, evaluative methodology.

Additionally, emotional anchoring as a formal design component proved particularly powerful. By designing around emotional arcs like "gradual control amidst chaos," the framework ensures that learning is not just intellectual but felt. This emotional realism enhances both player engagement and the internalization of targeted skills. In effect, VIOLETA suggests a new paradigm: games that do not simulate learning, but are learning.

5.2 How VIOLETA Could Be Improved

Despite its strengths, the VIOLETA framework is not without its challenges and limitations. One key issue is the cognitive burden placed on designers. The recursive, multi-layered design process—while conceptually sound—can be overwhelming, especially without digital tooling to support table construction, version tracking, and visualization of schema graphs. Without such support, there is a real risk that even well-intentioned educators might abandon the process due to complexity.

A second limitation lies in the subjectivity of emotion-mechanic mappings. While VIOLETA encourages designers to define emotional states intuitively, this openness may also lead to inconsistencies across games or teams. Structured libraries or validation rubrics could mitigate this issue by providing optional references that ensure emotional coherence.

Furthermore, the framework does not currently support collaborative design workflows. VIOLETA is optimized for single-author projects, and expanding it to accommodate distributed teams—narrative writers, gameplay engineers, learning scientists—would require new tooling and process models.

Finally, while the framework's claim that "more play = more mastery" is well-grounded in theory and structural design, it remains unvalidated empirically. Future work should focus on playtesting VIOLETA-based games across different learner demographics and domains to confirm its pedagogical effectiveness.

5.3 Potential Areas for Framework Expansion

- **Empirical Validation and Research:** Future studies should test whether extended gameplay within VIOLETA-designed systems results in measurable improvements in targeted skills. Experimental setups could include pre- and post-tests, player interviews, and behavioral logging.
- **Tool Development:** An integrated design environment could radically simplify the application of VIOLETA. Features could include automatic SIT and TIT generation, schema graph visualization, and gameplay simulation tools for cognitive load balancing.

- **AI-Assisted Game Design:** By training AI on a database of VIOLETA-designed games, designers could receive real-time suggestions for mechanics, emotional arcs, or skill integrations based on selected atomic units. This could make the framework accessible even to novice designers.

- **Support for Social Learning:** While VIOLETA currently emphasizes cognitive and emotional engagement, its support for relatedness (the third need in SDT) is limited. Expanding the framework to better support multiplayer and social learning contexts—e.g., cooperative skill-building, peer feedback loops—would increase its applicability.

- **SPARK Integration:** A potential future extension is the SPARK (System for Player Action, Reward, and Knowledge) module, which enhances player experience by embedding sensory feedback loops into the mechanics. While outside the scope of this thesis, SPARK represents a promising addition that could heighten emotional learning and player motivation.

5.4 Broader Implications

The implications of the VIOLETA framework extend beyond the design of a single game or prototype. VIOLETA proposes a new vocabulary and workflow for educational game design—one that is rooted in cognitive science, emotionally intelligent, and structurally reproducible. If widely adopted, VIOLETA could serve as a shared foundation for collaboration between educators, instructional designers, and game developers. At a societal level, VIOLETA contributes to a vision where learning is not separated from joy, challenge, or narrative. In a world increasingly shaped by automation and complexity, lifelong learning is not optional. Games—when designed properly—can become the scaffolds through which we acquire resilience, strategic thinking, and empathy. VIOLETA is one possible blueprint for building such games: games that don't just teach, but transform.

Chapter 6

Conclusion

6.1 Summary of Contributions

This thesis set out to address a fundamental challenge in educational game design: how to create games that are not only engaging but that also reliably foster real-world skill development. Traditional models often struggle to operationalize this goal, either remaining too abstract or offering overly rigid prescriptions. In response, the VIOLETA framework was developed.
VIOLETA introduces a structured, yet flexible, approach to designing educational games grounded in psychological theory, pedagogical logic, and creative design principles. Its key contributions include:

- **Triadic Integration** of Atomic Unit skills, emotional trajectories, and gameplay mechanics, ensuring that every design decision meaningfully contributes to learning outcomes.
- **Emotional Anchoring**, which transforms educational content from abstract objectives into emotionally resonant experiences, increasing engagement and deepening retention.
- **STACK (Structured Tiered Adaptive Core Knowledge)**, a tiered cognitive load management system that aligns game complexity with learner capability in real-time.
- **A step-by-step workflow**, from atomic unit selection to prototype development and integration, that turns theoretical principles into actionable design stages.
- The use of tools such as the **SIT, TIT, BMT, CLM,** and **SAM**, which bring transparency and evaluability to the design process.

These innovations collectively enable the core hypothesis of the framework: the more a player engages with the game, the more mastery they develop in the corresponding real-world skills.

6.2 Closing Thoughts on Games and Skill Development

VIOLETA is built on a belief that games can be more than entertainment—they can be engines of personal growth, systems of practice, and tools of transformation. Unlike passive consumption, gameplay is inherently active: players plan, experiment, adapt, and reflect. These are not just game behaviors; they are learning behaviors.

By structuring games to reward the correct expression of real-world skills, VIOLETA blurs the line between play and practice. It reframes the idea of "wasting time on games" by embedding mastery within the mechanics themselves. In doing so, it opens a path where learning becomes voluntary, joyful, and resilient—aligned with the way humans evolved to learn: through exploration, challenge, and emotional feedback.

Educational game design, then, is not merely about transmitting information. It is about creating experiences that change how people think, feel, and act. VIOLETA takes this seriously. It argues that we should no longer separate engagement from learning, nor play from purpose.

6.3 Future Outlook

While VIOLETA presents a theoretically sound and internally consistent methodology, several open challenges remain. Chief among them is empirical validation: the framework must now be tested in practice, across varied learner profiles, subject domains, and game formats. Playtesting, longitudinal studies, and cognitive assessments are needed to confirm whether its games truly deliver on the promise of "more play = more mastery."

Tool support will also be crucial. Manual application of VIOLETA is demanding. The creation of AI-assisted design platforms or interactive editors could lower the barrier to entry, allowing more educators and designers to experiment with the framework.

Moreover, collaborative design models must be explored. If VIOLETA is to scale beyond individual creators, it needs mechanisms for cross-role communication and distributed workflow integration—especially across pedagogy, design, and development.

Despite these challenges, the future of educational game design is promising. As automation reshapes the labor market and traditional learning methods struggle to keep pace, society needs tools that are flexible, scalable, and deeply human. Games—when built with intention—can meet that need. VIOLETA offers one such blueprint.

In closing, this thesis does not claim to have solved educational game design. Rather, it proposes a starting point: a structured invitation to rethink how we build games, how we teach, and ultimately, how we learn.

Bibliography

Ackerman, C. E. (2021). *Mihaly csikszentmihalyi: The father of flow* [Accessed: 2025-06-30]. https://positivepsychology.com/mihaly-csikszentmihalyi-father-of-flow/

Amory, A. (2007). Game object model version ii: A theoretical framework for educational game development. *Educational Technology Research and Development*, *55*(1), 51–77. https://doi.org/10.1007/s11423-006-9001-x

Annetta, L. A. (2010). The “i’s” have it: A framework for serious educational game design. *Review of General Psychology*, *14*(2), 105–112. https://doi.org/10.1037/a0018985

BoardGameGeek contributors. (2004). *Power grid* [Accessed: 2025-05-17]. https://boardgamegeek.com/boardgame/2651/power-grid

Bughin, J., Hazan, E., Lund, S., Dahlström, P., Wiesinger, A., & Subramaniam, A. (2018). Skill shift: Automation and the future of the workforce. https://www.mckinsey.com/featured-insights/future-of-work/skill-shift-automation-and-the-future-of-the-workforce

CAST. (2018). *Universal design for learning (udl) guidelines* [Accessed: 2025-04-27]. https://udlguidelines.cast.org/

Groos, K. (1898). *The play of animals* [Translated with the author’s cooperation by Elizabeth L. Baldwin]. D. Appleton; Company.

Hart, S., Halak, B., & Sassone, V. (2021). Motens: A pedagogical design model for serious cyber games. https://arxiv.org/abs/2110.11765

Jaccard, D., Suppan, L., Sanchez, E., Huguenin, A., & Laurent, M. (2021). The co.lab generic framework for collaborative design of serious games: Development study. *JMIR Serious Games*, *9*(3), e28674. https://doi.org/10.2196/28674

Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus; Giroux.

Nautiyal, V. V., Silverio, S. A., & Salvador, E. E. P. (2024). Let’s get onboard: A practical framework for designing and implementing educational board games in k-12 classrooms. *Frontiers in Education*, *9*, 1420515. https://doi.org/10.3389/feduc.2024.1420515

Przybylski, A. K., Rigby, C. S., & Ryan, R. M. (2010). A motivational model of video game engagement. *Review of General Psychology*, *14*(2), 154–166. https://doi.org/10.1037/a0019440

Roungas, B. (2016). A model-driven framework for educational game design. *International Journal of Serious Games*, *3*(3), 21–42. https://doi.org/10.17083/ijsg.v3i3.126

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, *55*(1), 68–78. https://doi.org/10.1037/0003-066X.55.1.68

Schüll, N. D. (2012). *Addiction by design: Machine gambling in las vegas*. Princeton University Press.

Silva, F. G. M. (2020). Practical methodology for the design of educational serious games. *Information*, *11*(1), 14. https://doi.org/10.3390/info11010014

Spieler, B., Schindler, C., Slany, W., Mashkina, O., Beltrán, M. E., Boulton, H., & Brown, D. (2017). Evaluation of game templates to support programming activities in schools. *Proceedings of the 11th European Conference on Games Based Learning (ECGBL)*.

Sweller, J., van Merrienboer, J. J. G., & Paas, F. G. W. C. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, *10*(3), 251–296. https://doi.org/10.1023/A:1022193728205

Xin, T. G. (2022). The framework of a game design (mda framework). *Preprint on ResearchGate*. https://www.researchgate.net/publication/360018773