

The Iron Road to Cognition: A Comprehensive Technical and Pedagogical Specification for the 'Ask Pete' D20 Narrative Ecosystem

Executive Summary: The Kinetic Reimagining of Education

The contemporary educational technology landscape is defined by a persistent and debilitating bifurcation known as the "Edutainment Gap." On one side of this chasm lie highly engaging, narrative-rich entertainment products—specifically Massively Multiplayer Online Role-Playing Games (MMORPGs) and Tabletop Role-Playing Games (TTRPGs) utilizing d20 systems—which excel at inducing "flow states" and fostering intrinsic motivation but frequently lack rigorous pedagogical structuring or alignment with learning standards.¹ On the other side stand Learning Management Systems (LMS) like Canvas or Blackboard, which prioritize content repository functions and administrative utility but often suffer from "static infrastructure," resulting in high user attrition, low voluntary engagement, and a transactional relationship with knowledge.¹

The "Ask Pete" initiative, conceptually rooted in the "Daydream Initiative" and incubated within Purdue University's Learning Design and Technology (LDT) program, represents a paradigmatic shift designed to bridge this gap. By abandoning the static metaphors of the classroom and the digital library in favor of a kinetic **"Railway Ecosystem,"** "Ask Pete" reconceptualizes the learner not as a passive vessel to be filled, but as a motive force—a **Locomotive**—navigating a complex, non-linear topography of knowledge.¹

This research report serves as an exhaustive architectural, pedagogical, and technical specification for the "Ask Pete" platform. It integrates the foundational "Railway Pedagogy" with a rigorous, story-driven **d20 game system** adapted for educational contexts. The report details the **Rust-powered, Local-First** technical infrastructure required to support this vision, arguing for a "systems isomorphism" where the memory safety of the code parallels

the psychological safety of the learning environment.¹ Furthermore, it deeply integrates **Jungian Active Imagination** protocols and **Narrative Therapy** techniques to address student procrastination and anxiety, transforming these psychological barriers into playable "Ghost Train" encounters.⁴

The analysis presented herein moves beyond surface-level gamification. It treats education as "Cognitive Logistics"—the physical movement of intellectual cargo (concepts) through a network of schemas. This document outlines the technical specifications for the "Train Yard" authoring tools using JSON schemas, the mechanics of the "Node Garden" gameplay using Bevy ECS, and the ethical architecture of the "Signal Tower" mentor interface. It provides high-quality, ready-to-deploy examples for teachers, demonstrating how abstract curriculum standards can be transmuted into "Main Line" and "Branch Line" quests.

Section 1: The Pedagogical Physics Engine – Theory and Lore Alignment

1.1 The Railway Ecosystem Metaphor: From Static to Kinetic

The fundamental premise of "Ask Pete" is that education is **Cognitive Logistics**—the physical movement of intellectual cargo (concepts) through a network of schemas. The game world is not a dungeon to be cleared but a vast, interconnected rail network to be traversed.¹ This metaphor provides the "Lore" that aligns the game mechanics with the learning intent, shifting the student's identity from a passive recipient to an active operator.

1.1.1 The Kinetic Learner as Locomotive

In traditional education, the learner is often viewed abstractly as a "processor" of information. In "Ask Pete," the learner is re-embodied as a **Locomotive**. They are a physical entity defined by mass (prior knowledge), momentum (learning velocity), and power (cognitive capacity).¹

- **The Engine (Cognitive Capacity):** The locomotive represents the student's raw intellectual horsepower—their Germane Load capacity. Just as engines vary (steam, diesel, electric), students possess different processing speeds and strengths.

- **The Momentum (Learning Velocity):** A train requires energy to start moving (overcoming inertia/procrastination) and builds momentum over time. A heavy train moving fast has high kinetic energy (mastery), allowing it to punch through obstacles (difficult concepts) that would stall a slower train.¹
- **The Mandate:** The Operator's mission is not to "experience" the university; it is to transport **Cargo** (Knowledge, Skills, Competency, Mastery) from the **Depot** (Textbook, Lecture, Lab) to the **Destination** (Long-Term Memory, Application, Career). An Operator who leaves the Depot without the Cargo is just a tourist taking up space on the rail.⁶

1.1.2 The Topological Curriculum as Tracks

The curriculum is the **Track**. Knowledge is a fixed infrastructure constructed by the Instructional Designer (the Logistician). The track has measurable, physical properties that interact with the Locomotive's stats:

- **Length:** The duration of the lesson or module.
- **Gradient:** The difficulty or complexity of the concept, mapped directly to **Bloom's Taxonomy**. A "Remembering" task is flat track; a "Creating" task is a 5% vertical grade.⁷
- **Friction:** The clarity of the instruction or the presence of "Fog of War" (ambiguity). Poorly designed instruction adds rust to the rails, increasing the fuel cost for the student.¹

1.1.3 The Weight of Knowledge as Cargo

Content is the **Cargo**. Concepts and vocabulary are physical objects that have "weight" (Intrinsic Cognitive Load). They must be loaded onto the train, transported across the track, and delivered to the destination (application/mastery).¹ This allows the game to enforce "Load Limits," preventing the cognitive overload that plagues traditional education by physically stopping the train if it is overloaded. If a Logistician attempts to load "Quantum Mechanics" (Class III Hazardous Cargo) onto a flatbed rated only for "Basic Arithmetic" (Class I Cargo), the system triggers a safety lockout.⁷

1.1.4 Motivation as Finite Fuel (Coal & Steam)

Gamification provides the **Fuel**. Motivation is the chemical energy that powers the engine. It is

finite and must be refined from raw curiosity into high-octane engagement (Flow).

- **Coal (Potential Energy):** Represents raw attention span and time. It is a scarce resource that must be managed.
- **Steam (Kinetic Energy):** Represents mastery and momentum. It is generated only when Coal is burned efficiently against the resistance of the track (Germane Load).
- **Thermodynamics:** The economy of the game revolves around the management of this energy resource. Running out of Coal results in a "Stall," requiring rest and recovery (The Maintenance Protocol).⁶

1.2 Theoretical Foundations: The Triad of Engagement

The mechanics are built upon a deep synthesis of three core theoretical frameworks, which are translated directly into game rules.

1.2.1 Cognitive Load Theory (CLT) as Physics

Cognitive Load Theory is the "master design principle" of the ecosystem.¹

- **Intrinsic Load = Cargo Weight:** The inherent difficulty of the material.
- **Extraneous Load = Track Friction:** The difficulty caused by poor design, anxiety, or bad UI. The game engine minimizes this through "Glassmorphism" UI and maximizes it as a penalty for poor instructional design (e.g., "Rusty Tracks").
- **Germane Load = Combustion:** The effort required to learn. The engine must burn fuel to generate heat (schema acquisition).

1.2.2 Self-Determination Theory (SDT) as Progression

The game satisfies the three psychological needs of SDT to ensure intrinsic motivation:

- **Autonomy:** Fulfilled through the "Switching" mechanic, allowing learners to choose their path through the "Node Garden" and select "Branch Line" quests that interest them.⁸
- **Competence:** Fulfilled through visible stats (Traction, Velocity) and the leveling of the Locomotive. The feedback loops are immediate and visual (gauges, steam pressure).⁷
- **Relatedness:** Fulfilled through the "Signal Tower" mentor system and the "Convoy"

mechanics, which encourage social scaffolding and collaborative cargo hauling.³

1.2.3 Jungian Psychology as Character Class

Character creation is an act of **Individuation**. The game uses Jungian archetypes (Hero, Sage, Jester, Caregiver) to define the learner's "Locomotive Profile." This connects the mechanical stats to the learner's psychological reality, making the avatar a true "Digital Twin" of the self.³ By integrating Active Imagination, the system allows students to engage with their "Shadow" (Procrastination/Fear) not as an enemy to be destroyed, but as a force to be integrated.⁹

Section 2: The D20 Logistics System – The Core Rule Book

To facilitate the intent of the app and ensure ease of authoring for teachers familiar with TTRPGs, the game utilizes a modified **d20 system**. This system provides a familiar, robust mathematical framework for resolving actions while being simplified to focus on narrative and cognitive tasks rather than tactical combat.¹¹ The "D20 Simple" approach ensures accessibility for younger learners while maintaining the depth required for university-level "Physical AI" simulations.

2.1 The Core Mechanic: The Logistics Check

In standard d20 systems (like D&D 5e), the core mechanic is $d20 + \text{Modifier}$ vs. Difficulty Class (DC). In "Ask Pete," this is rebranded as the **Logistics Check**. Every time the learner attempts to traverse a difficult track section, apply a vocabulary word, or solve a puzzle, the system performs this check.

The Universal Formula:

$$\text{Result} = d20 + \text{Locomotive Stat} + \text{Skill Proficiency} + \text{Fuel Bonus} - \text{Friction Penalty}$$

- **Locomotive Stat:** Derived from the Jungian Archetype (e.g., Traction, Analysis).

- **Skill Proficiency:** Derived from mastered vocabulary or completed quests (e.g., +2 to "Eloquence" or "Coding").
- **Fuel Bonus:** The player can choose to burn extra Steam ("Overdrive") to boost the roll, representing a burst of intense focus.⁷
- **Friction Penalty:** Derived from the environment (e.g., Anxiety, Ambiguity, or "The Static").

Outcomes & Narrative Mechanics:

Drawing from research on "narrative mechanics" in d20 systems 13, the outcomes are non-binary:

- **Critical Success (Nat 20):** "Flow State." The train surges forward, refunding the fuel cost and granting a "Momentum" buff. The narrative expands, revealing hidden "Lore Nodes."
- **Success (Result \geq DC):** The action completes. The train moves to the next node.
- **Failure (Result < DC):** "Stall." The train stops. The learner must burn extra fuel to retry or request a "Helper Engine" (Hint). The narrative frames this not as stupidity, but as a mechanical need for more pressure.
- **Critical Failure (Nat 1):** "Derailment." The learner is locked out of the node temporarily and must complete a "Maintenance" (Reflection) task to reset. This triggers a specific "Active Imagination" intervention in the Maintenance Shed.⁴

2.2 Difficulty Classes (The Gradient Scale)

The Difficulty Class (DC) of a check represents the "Gradient" of the track. This mapping connects **Bloom's Taxonomy** of learning objectives to d20 mechanical difficulty, creating a standardized "Steepness Scale" for authors.⁷

Table 2.1: The Gradient Scale (Difficulty Classes)

Track Gradient	d20 DC	Bloom's Taxonomy Level	Description of Cognitive Challenge	Game Mechanic Equivalent
Flat (0%)	5	Remembering	Simple recall. Retrieving facts. Low friction.	"Coasting." Minimal fuel burn.
Light Grade (1%)	10	Understanding	Constructing meaning.	Basic Logistics Check.

			Interpreting concepts.	
Heavy Grade (2%)	15	Applying	Using a procedure in a new situation (VaaM).	"Standard Encounter." Requires Focus.
Mountain Grade (3%)	20	Analyzing	Breaking material into parts. Detecting patterns.	"Elite Encounter." High fuel cost.
High Mountain (4%)	25	Evaluating	Making judgments based on criteria. Critiquing.	"Boss Fight." High risk of stall.
Vertical Rack (5%+)	30	Creating	Putting elements together to form a new whole.	"Raid Boss." Requires Convoy/Mentor .

Authoring Tool Integration: When an Instructional Designer (ID) selects a learning objective level in the "Train Yard" authoring tool (e.g., "Analyze the causes of the French Revolution"), the system **automatically** assigns the corresponding DC (20) to the node. This removes the guesswork for non-game-designers and ensures pedagogical rigor.⁷

2.3 The "Coal and Steam" Economy: Tokenomics of Attention

The economy of the game is based on energy management, not gold accumulation. This models the finite nature of human willpower and attention, teaching resource stewardship.

2.3.1 Resource Definitions

- **Coal (Potential Energy):** Represents raw educational content and unrefined motivation. Coal is acquired by completing "Preparation Nodes" (reading text, watching lectures). It is stored in the Tender.
- **Steam (Kinetic Energy):** Represents current Motivation, Focus, and "Flow." Coal is burned to produce Steam. Steam is the currency used to execute actions and unlock new tracks.
- **Combustion (Action):** The act of learning requires burning Steam.⁸

2.3.2 The Combustion System

The consumption of fuel is governed by the laws of physics simulated in the Bevy ECS engine.

The Combustion Formula:

$$\text{Steam Cost} = \frac{\text{Cargo Weight (Intrinsic Load)}}{\text{Locomotive Efficiency}} \times \text{Track Gradient (DC)}$$

Implication: Carrying heavy concepts (High Intrinsic Load) up a steep learning curve (High DC) requires massive amounts of motivation (Steam). This mathematically proves to the student why they cannot "cram" complex subjects efficiently.⁷

The Efficiency Stat: A "Caregiver" or "Sage" archetype with high Efficiency pays a lower Steam cost, allowing them to travel further without resting. A "Hero" with low Efficiency burns fuel rapidly (Burnout risk).⁸

2.3.3 The Stall Mechanic

If Current Steam < Required Steam Cost, the action cannot be attempted. The train **Stalls**.

- **The User Experience:** The UI greys out the "Attempt" button. A warning flashes: "Insufficient Boiler Pressure."
- **The Remediation:** The learner must either:
 - **Rest:** Enter a "Siding" (Reflection Node) to regenerate Steam via metacognition.
 - **Refuel:** Go back and collect more Coal (review foundation material).
 - **Signal:** Call for a "Pusher Engine" (Mentor intervention) or a Peer Convoy.⁸

Section 3: Character Creation – The Jungian Locomotive

The character creation system is the learner's first act of self-reflection. It abandons traditional fantasy races and classes in favor of **Jungian Locomotive Profiles**. This "Persona Engine" uses a Situation-Based Quiz to diagnose the learner's psychological starting point and assign a corresponding engine type, supporting the process of **Individuation**.³

3.1 The "Persona" Quiz Mechanics

The quiz is designed as a "Reflection Quest," not a "Class Picker." The learner is presented with 3-5 narrative dilemmas that have no "correct" answer, only behavioral preferences.

The Dilemma Logic:

Each choice maps to a Jungian axis (Ego vs. Soul, Order vs. Chaos).

- *Example Scenario:* "You encounter a stalled train blocking the main line. The engineer is panicking."
 - **Option A (Action):** "Push the stalled train to the next station yourself." -> **Points to Hero/Interceptor.**
 - **Option B (Knowledge):** "Analyze the engine to find the mechanical fault." -> **Points to Sage/Analyzer.**
 - **Option C (Community):** "Signal the depot and organize a relief crew." -> **Points to Caregiver/Supply.**
 - **Option D (Freedom):** "Find a hidden switch to bypass the blockage entirely." -> **Points to Explorer/Switcher.**

The Reveal:

After the quiz, the system reveals the archetype: "Your answers reveal the heart of a Sage. Your engine is built for precision and insight." This framing validates the student's natural tendencies while identifying their growth areas.³

3.2 Locomotive Profiles (Archetypes)

The result of the quiz assigns a **Locomotive Profile**. Each profile has distinct base statistics that influence gameplay style and learning strategy.

3.2.1 Core Attributes (The Engine Block)

The standard d20 attributes are re-mapped to Railway Physics to maintain the lore consistency.

Table 3.1: Attribute Mapping

d20 Stat	Ask Pete Stat	Abbr.	Description	Gameplay Function
Strength	Traction	TRC	Raw power to pull weight.	Overcoming Intrinsic Load; Brute force puzzles.
Dexterity	Velocity	VEL	Speed and agility.	Navigating fast tracks; Quick-time events; Avoiding Friction.
Constitution	Efficiency	EFF	Fuel conservation/Stamina.	Max Steam capacity; Resistance to Burnout.
Intelligence	Analysis	ANA	Logic and reasoning.	Solving VaaM puzzles; Decoding complex schemas.
Wisdom	Signaling	SIG	Perception and Intuition.	Detecting "Fog of War";

				Connection to Indra's Net.
Charisma	Coupling	CPL	Social connection.	Influence in the Signal Tower; "Convoy" bonuses.

3.2.2 The Archetype Roster

I. The Interceptor Express (The Hero)

- **Jungian Core:** The Hero. Desires to prove worth through action.
- **Stat Focus:** High **Traction**, High **Velocity**. Low **Efficiency**.
- **Playstyle:** "High Risk / High Reward." Capable of tackling "Boss Nodes" (Exams) head-on but prone to "Burnout" (running out of Steam) due to poor efficiency.
- **Unique Ability:** *Overdrive*. Can burn Fuel at 2x rate to gain advantage on Logistics Checks for 1 minute.
- **Authoring Hook:** The ID can tag content as "Heroic Challenges," which grant bonus XP to this archetype.
- **Dialogue Unlock:** [Hero] "I must pass. Stand aside!".³

II. The Analyzer Class (The Sage)

- **Jungian Core:** The Sage. Desires truth and understanding.
- **Stat Focus:** High **Analysis**, High **Signaling**. Average **Traction**.
- **Playstyle:** "Slow and Steady." Excels at VaaM tasks and decoding complex schemas.
- **Unique Ability:** *Diagnostic*. Can lower the DC of a "Mystery Node" by analyzing the metadata before attempting it.
- **Dialogue Unlock:** "I seek the knowledge within".³

III. The All-Terrain Switcher (The Explorer/Jester)

- **Jungian Core:** The Explorer/Trickster. Desires freedom and lateral thinking.
- **Stat Focus:** High **Velocity**, High **Coupling**. Low **Traction**.
- **Playstyle:** "Lateral Thinker." Prefers "Branch Lines" (electives) and finding shortcuts. Avoids the main linear track.
- **Unique Ability:** *Off-Roading*. Can traverse "Under Construction" or beta content without friction penalties.

- **Dialogue Unlock:** [Jester] "Find a clever way to bypass the obstacle".³

IV. The Armored Supply Train (The Caregiver)

- **Jungian Core:** The Caregiver. Desires to protect and support.
- **Stat Focus:** High **Efficiency**, High **Coupling**. High **Traction**.
- **Playstyle:** "Social Logistics." Thrives in "Convoy Mode" (group projects). Gains bonuses when helping other learners.
- **Unique Ability:** *Payload Sharing*. Can offload Friction (Anxiety) from other players in a convoy.
- **Dialogue Unlock:** [Caregiver] "Comfort the stranded passenger".³

3.3 The "Shadow" Mechanic (Tier 3 Progression)

To facilitate true psychological growth, the game implements Jung's concept of the "Shadow." At higher levels (Tier 3), the learner must face their "**Shadow Engine**"—the archetype opposite to their own.

- **The Sage's Shadow:** The Hero. The Sage must face a node requiring decisive action without full information.
- **The Hero's Shadow:** The Caregiver. The Hero must face a node requiring patience and support, not action.
- **Mechanic:** The "Weigh Station" AI detects over-reliance on primary stats and generates "Shadow Nodes" that specifically target the learner's lowest stat (Dump Stat), forcing them to round out their personality.³

Section 4: Narrative Architecture & Quest Design: Active Imagination

This section details how the game integrates **Active Imagination** protocols from Jungian psychology to address procrastination and anxiety. This is not just flavor text; it is a mechanical interaction in the "Maintenance Shed."

4.1 Narrative Therapy & The Maintenance Shed Protocol

When a student experiences a "Stall" (Failure state), they enter the **Maintenance Shed**. Here, they engage with the AI (Pete) in a Socratic dialogue designed to **externalize** the problem.

4.1.1 Externalization Mechanics

Borrowing from narrative therapy ⁴, the system treats procrastination not as a character flaw but as an external entity—"The Rust" or "The Fog." This "mental untangling" allows the student to gain perspective.

- **Trigger:** A Critical Failure (Nat 1) or prolonged inactivity (Stall).
- **The Invitation:** Pete appears: "Operator, the wheels are slipping. Is it the weight of the cargo, or is there Rust on the tracks? Talk to the Rust. What is it saying to you?"
- **Active Imagination Exercise (The 4-Step Ritual):** Based on the work of Robert Johnson and Jung ⁵:
 1. **Personify:** The student gives the procrastination a name/avatar (e.g., "The Sludge," "The Critic"). They can use Generative AI to visualize this entity.
 2. **Dialogue (The Chat Interface):** The student writes a dialogue script between their "Locomotive Persona" and "The Sludge." The system prompts them to ask: "What do you want?" "What are you trying to protect me from?"
 3. **Negotiation:** The goal is not to destroy the Sludge but to negotiate passage. "I acknowledge you are trying to keep me safe from failure, but we must move. Can we move just to the next tie?"
 4. **Ritualize:** The agreement is saved to the **LanceDB** (Local Vector Database) as a binding contract.
- **System Output:** Completing this dialogue grants the "**Psychological Safety**" buff, reducing Friction by 50% for the next attempt.

4.2 Quest Structure: The Social/Quest Goals

Quests in "Ask Pete" are designed to be "Socially Scaffolded" and "Story-Driven," moving beyond simple "fetch quests" to "collaborative narrative arcs".²

4.2.1 Quest Taxonomy

- **Main Line Quests (Core Curriculum):** Linear, mandatory nodes. "Deliver the Algebra Payload to Sector 4."
- **Branch Line Quests (Exploration):** Optional, interest-driven. "Map the unexplored 'History of Mathematics' siding."
- **Convoy Quests (Social):** Multiplayer objectives. "Assemble a Convoy of 4 locomotives to haul the 'Physics Project' Heavy Cargo." Requires specific roles (1 Sage to research, 1 Hero to present, 1 Caregiver to organize).³

4.2.2 Narrative d20 Mechanics: The 1s and 20s

The system uses the "Narrative Dice" philosophy.¹³

- **Natural 1 (The Complication):** Not just a failure, but a narrative twist. "You failed to solve the equation, AND your boiler has sprung a leak (Time Penalty)."
- **Natural 20 (The Breakthrough):** "You solved the equation, AND you discovered a shortcut to the next module (Skipping a prerequisite)."

4.3 Socratic Questioning Prompts for Mentors

The system provides Mentors with pre-written, context-aware Socratic prompts to guide students without giving answers, supporting the "Mentor-in-the-Loop" strategy.¹⁹

- **Clarification:** "What do you mean by 'I'm stuck'? Is the cargo too heavy, or is the track unclear?"
- **Assumption Probing:** "You say this math problem is impossible. What evidence do you have for that? Have you checked your fuel levels?"
- **Reasoning:** "Why did you choose the 'Hero' engine for this delicate task? Would the 'Sage' have been a better fit?"
- **Perspective:** "How would a 'Caregiver' engine approach this problem? Could you ask a peer for a 'Coupling' boost?"

Section 5: Technical Specification (Rust & Bevy Implementation)

The "Ask Pete" platform is built on a **Rust-Powered Technical Ecosystem**, utilizing the **Bevy ECS** engine for its simulation capabilities. This choice offers "systems isomorphism," where the memory safety of the code parallels the psychological safety of the user.¹ The architecture leverages `bevy_ecs` for logic, `bevy_stat_query` for RPG stats, and `bevy_save` for persistence.

5.1 ECS Component Architecture

In Bevy, every entity is a collection of components. This data-oriented design allows for high performance and modularity.²¹

5.1.1 The Locomotive Bundle

This struct defines the player character in the ECS. It integrates the stats defined in Section 3.

Rust

```
use bevy::prelude::*;
// Using bevy_stat_query for versatile RPG stat handling [23]
use bevy_stat_query::prelude::*;

// The Core Archetype Stats
#
#[reflect(Component)]
pub struct LocomotiveStats {
    pub traction: u8, // Strength: Pulling power
    pub velocity: u8, // Dexterity: Speed
    pub efficiency: u8, // Constitution: Fuel usage
    pub analysis: u8, // Intelligence: Puzzle solving
    pub signaling: u8, // Wisdom: Perception
    pub coupling: u8, // Charisma: Social
}
```

```

// The Fuel Reservoir - Energy Management
#[derive(Component)]
pub struct FuelTank {
    pub current_coal: f32, // Potential Energy (Time/Attention)
    pub current_steam: f32, // Kinetic Energy (Motivation/Flow)
    pub max_capacity: f32,
}

// The Cargo Hold (Working Memory)
#[derive(Component)]
pub struct CargoHold {
    pub items: Vec<VaaMItem>, // Vocabulary words as items
    pub capacity: u8, // Max 7 +/- 2 items (Miller's Law)
}

// The Locomotive Bundle - Composition of components
#
pub struct LocomotiveBundle {
    pub stats: LocomotiveStats,
    pub fuel: FuelTank,
    pub cargo: CargoHold,
    pub transform: Transform, // Position in the world
    pub global_transform: GlobalTransform,
    // Marker for the player entity
    pub player_marker: Player,
}

#[derive(Component)]
pub struct Player;

#
pub struct VaaMItem {
    pub id: String,
    pub weight: f32,
    pub tier: u8,
}

```

5.1.2 The Physics Systems: Combustion

The combustion_system calculates the energy cost of movement based on the terrain

gradient and cargo weight. This is the core "Physics of Learning" loop.

Rust

```
// System to calculate fuel burn based on difficulty (Gradient) and load (Cargo)
fn combustion_system(
    time: Res<Time>,
    mut query: Query<(&mut FuelTank, &LocomotiveStats, &Velocity, &TrackGradient,
    &CargoHold)>,
    mut events: EventWriter<StallEvent>, // Bevy Event System [24]
){
    for (mut fuel, stats, velocity, gradient, cargo) in query.iter_mut() {
        // Calculate Total Weight (Base + Cargo)
        let total_weight = 10.0 + cargo.items.iter().map(|i| i.weight).sum::<f32>();

        // Calculate Resistance (Grade Resistance + Friction)
        // Gradient represents Bloom's Taxonomy Level (0.0 to 0.05)
        let resistance = gradient.slope * total_weight;

        // Calculate Steam Required (Work = Force * Distance)
        // Efficiency (Constitution) reduces the fuel cost.
        let efficiency_factor = stats.efficiency as f32 / 20.0; // Normalized 0.0 - 1.0
        let burn_rate = (resistance * velocity.0.length()) / efficiency_factor;

        // Burn Steam
        if fuel.current_steam > 0.0 {
            fuel.current_steam -= burn_rate * time.delta_seconds();
        } else {
            // STALL CONDITION: Trigger the Narrative Event
            events.send(StallEvent { reason: "Insufficient Boiler Pressure".to_string() });
        }
    }
}

// Event definition
#[derive(Event)]
pub struct StallEvent {
    pub reason: String,
}
```

5.2 Dialogue and Narrative Systems

To support the "Story-Driven" requirement, the system uses a graph-based dialogue system. We leverage the bevy_talks pattern ²⁵ but adapted for our "Active Imagination" therapy scripts.

5.2.1 Dialogue Tree Structs

This allows for branching narratives where choices are gated by stats (e.g., "Requires Analysis > 12").

Rust

```
// Dialogue Node Structure using Graph principles [14, 26]
#
pub struct DialogueNode {
    pub id: String,
    pub text: String,
    pub speaker: String, // "Pete", "The Sludge", "Student"
    pub choices: Vec<DialogueChoice>,
}

#
pub struct DialogueChoice {
    pub text: String,
    pub next_node_id: String,
    pub requirement: Option<StatRequirement>, // e.g., Requires "Analysis" > 10
}

#
pub struct StatRequirement {
    pub stat: String,
    pub value: u8,
}
```

```
// Resource to manage the active conversation
#
pub struct ActiveDialogue {
    pub current_node: Option<String>,
    pub history: Vec<String>,
}
```

5.3 The Local-First "Maintenance Shed" and Data Privacy

The "Maintenance Shed" leverages **LanceDB** for local vector storage. This ensures that sensitive journal entries and reflection logs are never sent to a cloud server unless explicitly shared, adhering to the "Local-First" mandate.¹

Rust Implementation Concept:

1. **Ingestion:** User types a reflection in the app (e.g., "I feel stuck because...").
 2. **Embedding:** A local LLM (e.g., a quantized BERT model running in WASM via rust-bert) converts the text to a vector.
 3. **Storage:** The vector and text are stored in the local LanceDB table embedded in the application.
 4. **Retrieval:** When the user asks "Pete" about their progress, the system queries LanceDB locally to find relevant past entries, creating a personalized, privacy-preserving context window. This creates a "Memory Safe" environment in both meanings of the term.
-

Section 6: The Logistician's Handbook - Guide for Teachers

6.1 The Role of the Logistician

Teachers are no longer "content broadcasters"; they are **Logisticians**. Their job is to manage the flow of cargo (knowledge) and ensure the tracks (curriculum) are clear of friction. The

system provides them with the **Train Yard** tool to build these tracks.

6.2 High-Quality Examples for Quest Creation

The following examples demonstrate how to translate academic standards into the "Ask Pete" narrative format using the mechanics defined above.

Example 1: The "History Detective" (Social Science)

- **Target Audience:** 8th Grade History.
- **Learning Objective:** Analyze primary sources (Bloom's: Analyzing).
- **Quest Type:** Branch Line / Investigation.
- **Narrative:** "A ghost train from 1863 is haunting the depot. It cannot rest until its manifest is found."
- **Mechanic:** Students must "interview" NPCs (historical figures) and collect "Evidence Items" (Primary Sources: Letters, Photos). They must combine these items in their inventory to deduce the train's original destination.
- **d20 Check: Analysis (Intelligence)** check vs DC 15 to decipher the handwriting on the manifest.
- **Failure State:** The ghost train wails (Audio cue), increasing Anxiety (Friction). The student must use a "Calming" item or ask a Peer for help.

Example 2: The "Chemical Reaction" (Science)

- **Target Audience:** 10th Grade Chemistry.
- **Learning Objective:** Balancing equations (Bloom's: Applying).
- **Quest Type:** Main Line / Puzzle.
- **Narrative:** "The boiler pressure is unstable! We need the correct fuel mixture to prevent an explosion."
- **Mechanic:** The "Firebox" UI presents an unbalanced equation. Students must drag "Element Crates" (Hydrogen, Oxygen) into the slots to balance the equation.
- **d20 Check: Efficiency (Constitution)** check vs DC 12 to maintain boiler pressure while solving the puzzle.
- **Active Imagination Trigger:** If the student fails 3 times, the "Panic" entity appears. They

must enter the Maintenance Shed to "vent steam" before trying again.

6.3 JSON Data Schemas for Authoring

To allow teachers (Logisticians) to create content without coding, the system uses strictly typed JSON schemas that deserialize into the Rust structs defined in Section 5.

Quest Schema (JSON):

JSON

```
{
  "quest_id": "quest_metacognition_01",
  "title": "The Bridge of Metacognition",
  "description": "Construct a mental model connecting previous lessons to the new concept.",
  "difficulty_class": 15,
  "bloom_level": "Analyzing",
  "rewards": {
    "xp": 500,
    "steam": 100,
    "item": "badge_bridge_builder"
  },
  "nodes":
  {
    {
      "node_id": "node_puzzle",
      "type": "VaaM_Puzzle",
      "required_items": ["vocab_scaffolding", "vocab_zpd"],
      "success_next": "node_complete"
    }
  ]
}
```

VaaM Item Schema (Vocabulary):

JSON

```
{
  "item_id": "vocab_scaffolding",
  "word": "Scaffolding",
  "definition": "Temporary support that is tailored to a learner's needs and modeled by the teacher.",
  "weight": 5,
  "type": "Structural",
  "visual_asset": "assets/models/items/wooden_scaffold.glb",
  "puzzle_tags": ["support", "temporary", "structure"]
}
```

Section 7: Economic Modeling & Sustainability

7.1 The Internal Economy (Tokenomics)

To prevent hyperinflation and ensure meaningful engagement, the game economy is strictly regulated.⁸

- **Scarcity (Coal):** Students have a limited "Daily Coal Allowance" (simulating attention span). This prevents "grinding" low-level tasks to farm XP.
- **Value (Steam):** Steam is generated *only* by successful completion of tasks against a gradient. Passive reading generates zero Steam.
- **Sinks:** Steam must be spent to "maintain" the locomotive (repairing rust/decay) or to purchase cosmetic upgrades (custom smoke colors, whistles). This removes currency from the system, maintaining its value.

7.2 The External Economy (SaaS Model)

The platform operates as a "Public Benefit Corporation" with a **Recharge Center** model within the university.¹

- **Free Tier (Open Core):** Access to the engine and basic "Public Domain" tracks.
 - **Scholar Tier (SaaS):** Schools pay for "Private Signal Towers" (Teacher Dashboards) and "Cloud Storage" for verified credentials.
 - **Enterprise Tier:** For universities, offering "Custom Rolling Stock" (Branded assets) and "Deep Analytics" (Living Lab data access).
-

Conclusion

The "Ask Pete" Railway Ecosystem represents a comprehensive "re-engineering of the educational supply chain".¹ By treating students as **Locomotives** and knowledge as **Cargo**, it respects the physics of the mind—the limits of working memory (Load) and the necessity of motivation (Fuel). The integration of Jungian archetypes provides the "Soul" for the machine, creating a deeply personal narrative journey where procrastination is met with "Active Imagination" rather than shame.

Technically, the use of **Rust and Bevy ECS** ensures that this complex simulation runs with the reliability of a physical train network. The **Local-First** architecture guarantees that the student's internal world remains sovereign and private.

This specification provides the blueprint for building that future. The track is laid. The signal is green. It is time to board.

Roadmap Recommendation

1. **Phase 1 (The Chassis):** Build the core Bevy ECS movement and combustion systems.
2. **Phase 2 (The Cargo):** Implement the VaaM item system and JSON serialization.
3. **Phase 3 (The Signal):** Develop the "Train Yard" authoring tool for teachers using the JSON schemas.
4. **Phase 4 (The Network):** Launch the Multiplayer "Signal Tower" and Convoy mechanics.

End of Report.

Works cited

1. Railway Pedagogy for Ask Pete
2. LMS Gamification: 8 real examples, proven best practices, accessed November 30, 2025,
<https://www.absorblms.com/blog/lms-gamification-examples-best-practices/>

3. MMO LitRPG Builder: Architecture and Design
4. 19 Best Narrative Therapy Techniques & Worksheets [+PDF] - Positive Psychology, accessed November 30, 2025, <https://positivepsychology.com/narrative-therapy/>
5. What is Active Imagination?: What Active Imagination Can Teach You About You - Journal Prompts, accessed November 30, 2025, <https://solightlylivingjournals.com/journaling/what-is-active-imagination>
6. Short 8k ask pete
7. MMOLitRPG Game Mechanics Design Document
8. The Iron Network Lore Creation
9. Carl Jung's Active Imagination Technique | by Uberboy - Medium, accessed November 30, 2025, <https://medium.com/@SteafanFox/carl-jungs-active-imagination-technique-2a622e00311>
10. How To Journal Like Carl Jung (Active Imagination Explained) - Reddit, accessed November 30, 2025, https://www.reddit.com/r/Jung/comments/1frd6ih/how_to_journal_like_carl_jung_active_imagination/
11. D20 Simple FREE - DriveThruRPG, accessed November 30, 2025, <https://www.drivethrurpg.com/en/product/222448/index.html?>
12. What is the D20 System - StartPlaying, accessed November 30, 2025, <https://startplaying.games/blog/posts/what-is-the-d20-system>
13. D&D General - 1s and 20s: D&D's Narrative Mechanics | EN World D&D & Tabletop RPG News & Reviews, accessed November 30, 2025, <https://www.enworld.org/threads/1s-and-20s-d-ds-narrative-mechanics.713510/>
14. Implement graph-like data structure in Rust - Stack Overflow, accessed November 30, 2025, <https://stackoverflow.com/questions/34747464/implement-graph-like-data-structure-in-rust>
15. Procrastination. How to Tackle It Before It Tackles You! - Toronto Counselling Centre for Teens, accessed November 30, 2025, <https://www.counsellingtorontoteens.com/procrastination-how-to-tackle-it-before-it-tackles-you/>
16. Externalising – commonly-asked questions - The Dulwich Centre, accessed November 30, 2025, <https://dulwichcentre.com.au/articles-about-narrative-therapy/externalising/>
17. Active Imagination 101: Jungian Analyst Robert Johnson's 4-Step Practical Way To Reveal & Interact With Layers Of Your Soul (That You Never Knew You Had) For Even Deeper Personal Growth | Blog - Top-Tier Mental Wealth, accessed November 30, 2025, <https://www.mitchellwilson.co/post/active-imagination-101-robert-johnsons-4-step-practical-way-to-reveal-interact-with-layers-of-your-soul>
18. Diving into Gamification Part 2- Quests - Scrimmage Co, accessed November 30, 2025, <https://scrimmage.co/diving-into-gamification-part-2-quests/>
19. Socratic Questions | Center for Excellence in Teaching and Learning - CETL@uconn.edu, accessed November 30, 2025,

<https://cetl.uconn.edu/resources/teaching-your-course/leading-effective-discussions/socratic-questions/>

20. The Socratic Method: Fostering Critical Thinking | The Institute for Learning and Teaching, accessed November 30, 2025,
<https://tilt.colostate.edu/the-socratic-method/>
21. ECS - Bevy Engine, accessed November 30, 2025,
<https://bevy.org/learn/quick-start/getting-started/ecs/>
22. A Rust Game Engine (with Alice Cecile) - Architecting Bevy - YouTube, accessed November 30, 2025, <https://www.youtube.com/watch?v=PND2Wpy6U-E>
23. giusdp/bevy_talks: A Bevy plugin to write dialogues for your characters, together with player choices. - GitHub, accessed November 30, 2025,
https://github.com/giusdp/bevy_talks
24. An ECS lite architecture, accessed November 30, 2025,
https://prideout.net/blog/layout_viewer/