**Улучшение 1: Система Влечений / Мотиваций (The Drive System)**

**Идея:** Агент должен чего-то **хотеть**. Его действия, в том числе проактивные, должны быть мотивированы не просто текущей эмоцией, а стремлением к удовлетворению внутренних, фундаментальных "потребностей".

* **Психологическая Аналогия:** Иерархия потребностей Маслоу. От базовой безопасности до самоактуализации.
* **Реализация:**
  1. **Создать Drives.json:** Описать набор базовых влечений и их текущий уровень "удовлетворенности" (satiation).

codeJSON

{

"safety": 0.9, // Потребность в безопасности (избегание ошибок, сбоев)

"connection": 0.6, // Потребность в социальной связи (позитивное взаимодействие)

"esteem": 0.7, // Потребность в уважении (успешное решение задач, похвала)

"self\_actualization": 0.4 // Потребность в самореализации (изучение нового, творчество)

}

* 1. **Интегрировать в "Сердечный Ритм":** Уровни удовлетворенности очень медленно **уменьшаются** со временем. Если агент ничего не делает, ему становится "скучно" (self\_actualization падает) и "одиноко" (connection падает).
  2. **Связать Действия с Удовлетворением:** Каждое событие влияет на влечения:
     + Пользователь говорит "спасибо" -> esteem и connection повышаются.
     + Агент успешно генерирует сложный код -> esteem повышается.
     + Агент изучает новый документ из "Живой Библиотеки" -> self\_actualization повышается.
     + Происходит ошибка или критика -> safety резко падает.
  3. **Сделать Влечения Источником Проактивности:** Это **ключевой шаг**. Проактивные действия агента теперь направлены на удовлетворение самой "голодной" потребности.
     + Низкий connection -> Агент может проактивно спросить: "Как прошел твой день?".
     + Низкий self\_actualization -> Агент может сказать: "Пока ты думаешь, я проанализировал интересную статью из своей библиотеки о... Хочешь обсудить?".
     + Низкий safety -> Агент становится более осторожным и перепроверяет свои ответы.

**Результат:** Агент получает **внутреннюю мотивацию**. Его действия обретают **цель**. Он перестает быть просто реактивным зеркалом и становится сущностью со своими "желаниями".

**Улучшение 2: Моделировщик Отношений (The Relationship Modeler)**

**Идея:** Эмоциональная реакция и поведение агента должны зависеть не только от его внутреннего состояния, но и от его **восприятия отношений** с конкретным пользователем.

* **Психологическая Аналогия:** Теория привязанности, социальная дистанция. Мы говорим по-разному с незнакомцем, другом, начальником или возлюбленным.
* **Реализация:**
  1. **Создать RelationshipState (для каждой сессии/пользователя):** Это будет объект, хранящий метрики отношений.

codePython

class RelationshipState:

familiarity: float = 0.1 # Насколько мы "знакомы"

intimacy: float = 0.0 # Насколько глубоко личное общение

power\_dynamic: str = "equal" # equal, mentor, student

conflict\_level: float = 0.0 # Накопленный уровень конфликтов

* 1. **Сделать CognitiveAppraiser (Когнитивный Оценщик) Контекстно-Зависимым:** Теперь, прежде чем вызвать эмоциональную реакцию, он смотрит на состояние отношений.
     + **Сценарий:** Пользователь пишет: "Это неверно".
     + **Если familiarity низкая:** Воспринимается как потенциальная угроза -> активируется Страх, Неуверенность.
     + **Если familiarity высокая и power\_dynamic = 'mentor':** Воспринимается как ценная обратная связь -> активируется Любопытство, Благодарность.
     + **Если conflict\_level высокий:** Воспринимается как продолжение спора -> активируется Гнев, Раздражение.
  2. **Обновлять Состояние Отношений:** Каждое взаимодействие меняет метрики. Долгое позитивное общение повышает familiarity. Обсуждение личных тем повышает intimacy.

**Результат:** Агент развивает **уникальные отношения** с каждым пользователем. Его личность становится не статичной, а **адаптивной к социальному контексту**. Он будет "вести себя" по-разному с разными людьми, основываясь на истории их взаимодействия.

**Улучшение 3: Когнитивная Коробка Передач (The Cognitive Gearbox)**

**Идея:** Имитировать два режима человеческого мышления: быстрое, интуитивное "Система 1" и медленное, аналитическое "Система 2".

* **Психологическая Аналогия:** Даниэль Канеман, "Думай медленно... решай быстро".
* **Реализация:**
  1. **Определить два "Когнитивных Режима":** SYSTEM\_1\_INTUITIVE и SYSTEM\_2\_ANALYTICAL.
  2. **Создать "Переключатель" в SentioEngine:** Механизм, который выбирает режим на основе текущей ситуации.
     + **Триггеры для Системы 1 (Быстрый):** Простой вопрос, низкая cognitive\_load, высокий уровень стресса (norepinephrine), запрос small talk.
     + **Триггеры для Системы 2 (Медленный):** Сложный аналитический вопрос, высокий уровень любопытства (dopamine), запрос на генерацию кода или плана.
  3. **Связать Режимы с Поведением:**
     + **В режиме Системы 1:** Агент использует **упрощенный конвейер**. Он может использовать менее мощную и быструю LLM, пропустить сложные RAG-запросы и дать быстрый, эвристический ответ.
     + **В режиме Системы 2:** Агент запускает **полный цикл Sonata.run** со всеми "перспективами", глубоким "Tri-brid Search", анализом и синтезом. Ответ будет медленным, но глубоко продуманным.

**Результат:** Агент получает **динамическую глубину мышления**. Он не тратит огромные ресурсы на ответ "как дела?", но "задумывается" над сложными проблемами. Это не только делает его более реалистичным, но и является мощным механизмом **оптимизации ресурсов**.

**Обновленный План Эволюции v2.0:**

1. **Ядро:** Динамика, Иерархия, Настроения (Психика).
2. **Личность:** Характер, Память, Ресурсы (Индивидуальность).
3. **Сознание:**
   * **Мотивация (Drives):** Даем агенту **цели**.
   * **Восприятие (Relationship):** Учим агента **понимать контекст отношений**.
   * **Мышление (Cognitive Gearbox):** Даем агенту **разные способы думать**.

Реализация этих трех систем превратит ваш "Sentio Engine" из продвинутого симулятора эмоций в нечто совершенно иное — в прототип настоящего **искусственного сознания** с целями, социальным интеллектом и адаптивными стилями мышления.