

MyBaby: Путь к Супер-Игре

🎯 Цель: От веб-прототипа к полноценной игре

Этап 1: Улучшение веб-версии (Next.js)

Game Engine подход:

- **Phaser.js** - полноценный 2D игровой движок для веба
- **PixiJS** - высокопроизводительная графика и анимации
- **Matter.js** - физический движок для реалистичных взаимодействий

Архитектура игры:

```
GameEngine/  
├── entities/  
│   ├── Baby.ts      # Главный персонаж  
│   ├── Item.ts      # Базовый класс предметов  
│   └── Environment.ts # Игровое окружение  
├── systems/  
│   ├── PhysicsSystem.ts # Физика столкновений  
│   ├── AudioSystem.ts  # Звуковые эффекты  
│   └── AnimationSystem.ts # Анимации  
└── scenes/  
    ├── PlayScene.ts   # Основная игра  
    ├── MenuScene.ts   # Главное меню  
    └── SettingsScene.ts # Настройки
```

Этап 2: Продвинутые возможности

Система эмоций и ИИ малыша:

- Конечные автоматы для поведения
- Система потребностей (голод, сон, веселье, здоровье)
- Адаптивный ИИ - малыш запоминает предпочтения игрока

Расширенный геймплей:

- Мини-игры (кормление, одевание, купание)
- Система прогресса и достижений
- Коллекция предметов и костюмов
- День/ночь цикл с разными активностями

Этап 3: Переход на Flutter (мобильная версия)

Технический стек:

- **Flutter + Flame Engine** - кроссплатформенная разработка игр
- **Shared State Management** - синхронизация между платформами
- **Local Storage** - сохранение прогресса локально

Архитектура Flutter версии:

```
flutter_mybaby/  
├── lib/  
│   ├── game/  
│   │   ├── components/ # Игровые компоненты  
│   │   ├── systems/    # Игровые системы  
│   │   └── worlds/      # Игровые миры  
│   ├── data/  
│   │   ├── models/      # Модели данных  
│   │   └── repositories/ # Управление данными  
│   └── ui/  
│       ├── screens/     # Экраны приложения  
│       └── widgets/      # UI компоненты
```

Immediate Next Steps

1. Установка Game Engine (Web)

```
bash  
  
pnpm add phaser matter-js howler  
pnpm add -D @types/matter-js
```

2. Создание Game Scene

```
typescript
```

```
// src/game/scenes/PlayScene.ts
import Phaser from 'phaser';

export class PlayScene extends Phaser.Scene {
  private baby!: Phaser.GameObjects.Sprite;
  private items: Phaser.GameObjects.Group;

  constructor() {
    super({ key: 'PlayScene' });
  }

  preload() {
    this.load.image('baby-happy', '/mybaby/babysmile.svg');
    this.load.image('baby-sad', '/mybaby/baby.png');
    this.load.image('bottle', '/mybaby/bottle.png');
    this.load.image('teddy', '/mybaby/teddy.png');
  }

  create() {
    this.baby = this.add.sprite(400, 300, 'baby-sad');
    this.baby.setInteractive();
    this.input.setDraggable(this.baby);

    this.items = this.add.group();
    this.createItems();
    this.setupDragAndDrop();
  }
}
```

3. Система состояний малыша

typescript

```
// src/game/entities/BabyState.ts
```

```
export class BabyState {  
  private happiness = 50;  
  private hunger = 50;  
  private energy = 50;  
  private health = 100;  
  
  feedBaby(nutrition: number) {  
    this.hunger = Math.min(100, this.hunger + nutrition);  
    this.happiness += 10;  
    this.emitStateChange();  
  }  
  
  playWithToy(funValue: number) {  
    this.happiness = Math.min(100, this.happiness + funValue);  
    this.energy -= 5;  
    this.emitStateChange();  
  }  
}
```

Продвинутые фичи

Физический движок:

- Реалистичное падение предметов
- Столкновения и отскоки
- Гравитация и инерция

Звуковая система:

- Реактивные звуки на действия
- Фоновая музыка для настроения
- Пространственное аудио

Визуальные эффекты:

- Particle системы (звездочки, сердечки)
- Плавные переходы между состояниями
- Параллакс-скроллинг фонов

AI-поведение:

- Малыш реагирует на игнорирование
- Адаптивные потребности

- Персонализированные предпочтения

Flutter Migration Strategy

Поэтапный переход:

1. **API-слой** - создать REST API для игровых данных
2. **Shared Logic** - выделить бизнес-логику в отдельные пакеты
3. **UI Recreation** - воссоздать интерфейс во Flutter
4. **Game Engine** - интегрировать Flame engine
5. **Cross-platform** - синхронизация между веб и мобильной версией

Преимущества Flutter версии:

- Нативная производительность
- Доступ к камере/микрофону
- Push-уведомления
- Офлайн режим
- App Store распространение

Roadmap Timeline

Неделя 1-2: Phaser.js интеграция, базовая физика **Неделя 3-4:** Расширенная система состояний, звуки **Неделя 5-6:** Мини-игры и достижения **Неделя 7-8:** Flutter прототип **Неделя 9-12:** Полноценная мобильная версия

Monetization Strategy

- Freemium модель
- Premium костюмы и игрушки
- Дополнительные комнаты/миры
- Семейная подписка с синхронизацией