

Системный и бизнес анализ в GameDev-стартапе

Всеволод Ляпин



Обо мне



Всеволод Ляпин

IDDQD

- ▶ IT менеджер из разработки ПО
Основной опыт в FinTech
Так же - EdTech, MarTech,
GameDev
- ▶ Участие в проектах и командах, используя различные методологии
- ▶ Разносторонний опыт внедрения, построения, изменения процессов командной работы в разработке ПО

Цели

- ▶ Понимание специфики Геймдева
- ▶ Поиск аналогий работы в GameDev из FinTech
- ▶ Как построить работу в GameDev стартапе
- ▶ Как привычные методики применить в новой области



Особенности GameDev стартапа

Стартап - это как запуск корабля без носа на орбиту



Особенности GameDev стартапа

- ▶ Ламповая атмосфера
- ▶ Сжатые сроки
- ▶ Ограниченные ресурсы
- ▶ Не проработанные требования
- ▶ Высокие ожидания
- ▶ Накопление собственного опыта



Из FinTech в GameDev

Команда и роли

- ▶ Анализ и постановка ТЗ
- ▶ Визуал
- ▶ Написание кода

Процессы

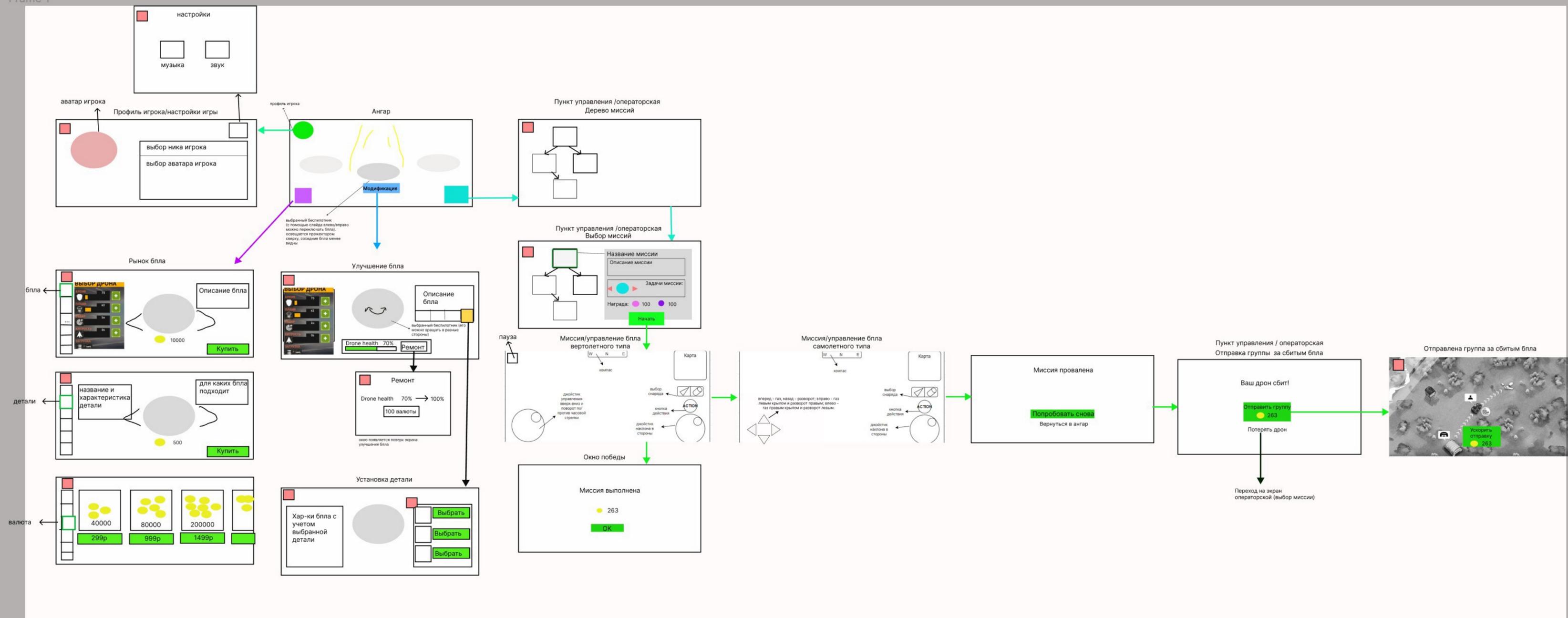
- ▶ Методологии работы
- ▶ Командная атмосфера
- ▶ Инструменты



Бизнес анализ в геймдизайне

Прототипирование логики геймплея

Frame 1



Бизнес анализ в геймдизайне

Прототипирование логики



Бизнес анализ в геймдизайне

Сравнительный анализ конкурентных решений

The collage includes several key elements:

- Invite Friends Screens:** Multiple screenshots showing 'ПРИГЛАШАЙ ДРУЗЕЙ' (Invite Friends) screens with various reward structures like 'кейс с изумрудами' (gem chest) and 'БОНУС ЗА ВХОД' (bonus for login).
- Gameplay and Rewards:** Screenshots of game environments and reward pop-ups, such as 'Награда за принятия приглашение!' (Reward for accepting invitation!) and 'Измуруды' (Emeralds).
- Storefronts:** Detailed views of in-game stores ('Магазин') selling items like 'Бэги жемчужин' (Gem bags), 'Сундуки жемчужин' (Gem chests), and 'Вагоны жемчужин' (Gem wagons) for different currencies (Rubles, Gems).
- Resource Alerts:** Screenshots of error messages like 'Не хватает ресурсов' (Not enough resources) and 'Не хватает кристаллов' (Not enough crystals).
- UI Elements:** Examples of reward icons, including 'Сундук - арт' (Chest - art) and 'Получено картинки + цифры' (Received pictures + numbers).
- Player Progress:** A screenshot of a player's profile with a 'BRAWL PASS' and 'GEM GRAB' badge.

Чем больше игр сыграют твои друзья, тем больше твоя награда!

добавь друзей в игру, и вы все получите награду

Получай финальную награду, когда все твои друзья сыграют максимальное количество игр!

Получено:

Иконки валют + визу их количество
Все валюты - 3 типа жетонов + золото + изумруды

Сундук - арт

Получено картинки + цифры

Ok

Бизнес анализ в геймдизайне

CustDev

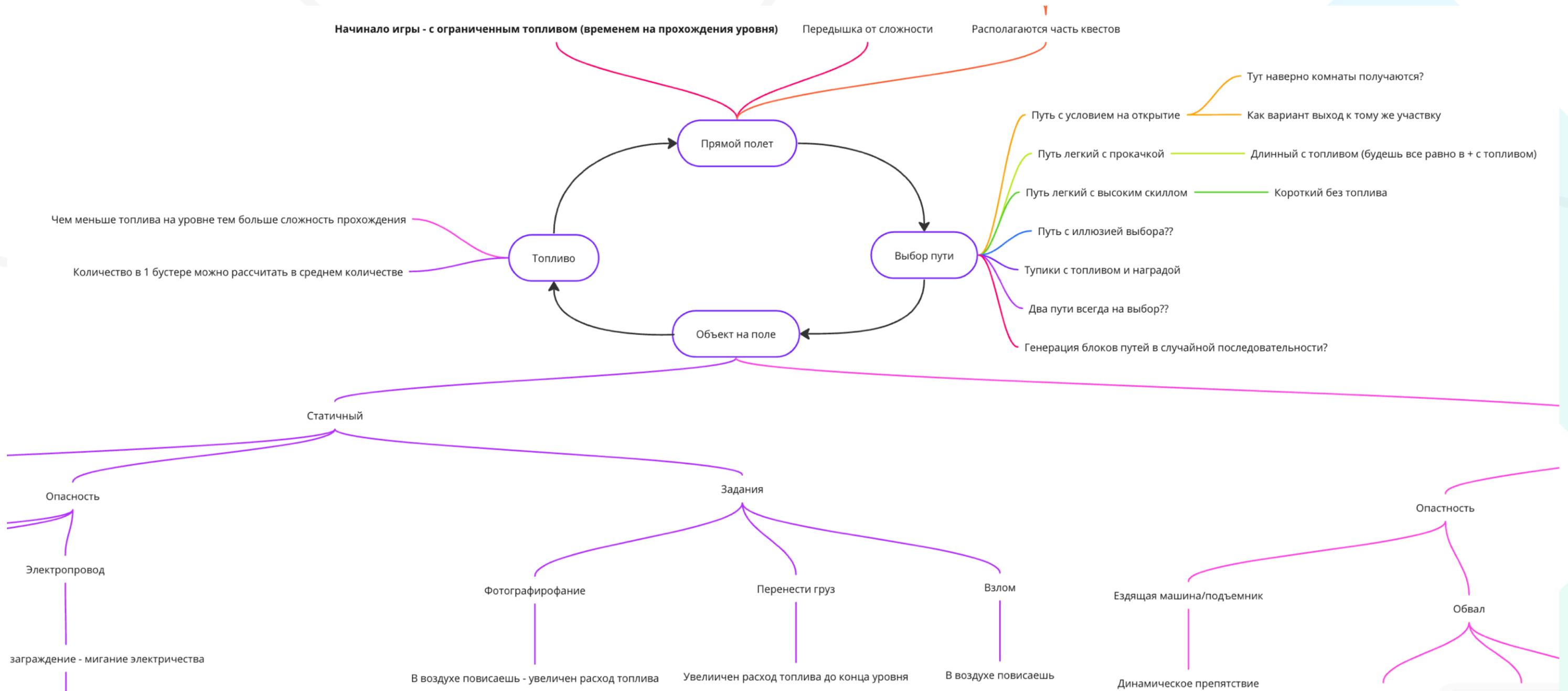


Разработка игровых проектов на заказ:

- Получение «болей» клиента
- Проектирование нескольких вариантов
- Тестирование на клиенте и сбор обратной связи
- Выбор итогового варианта для разработки

Бизнес анализ в геймдизайне

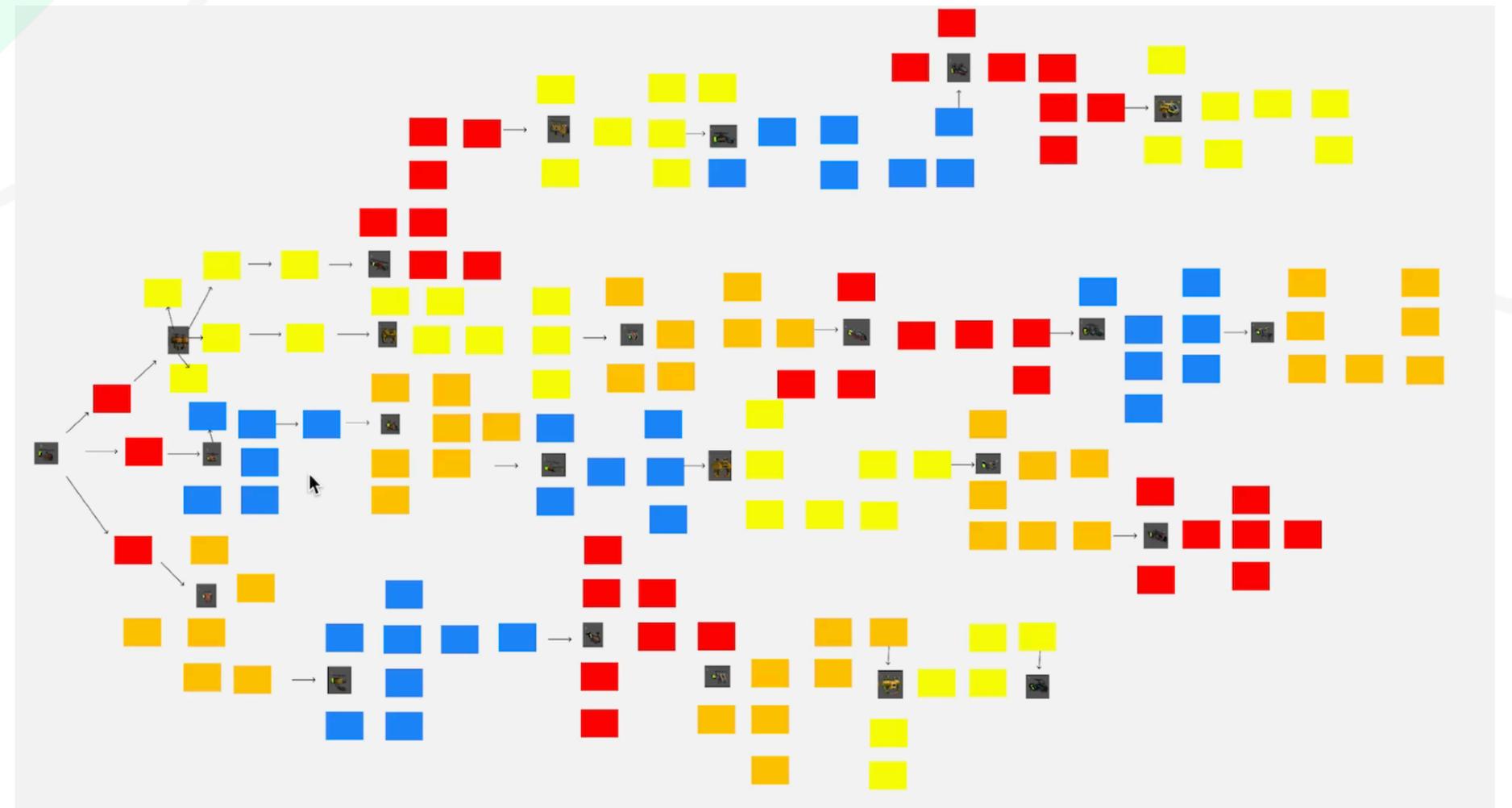
Майна-карты



Системный анализ в геймдизайне

Описание игровых юнитов, объектов и правил взаимодействия как модель «Сущность-связь»

Увеличение награды за грузовые миссии на	15%
Уменьшение расхода топлива на	1
	-
Увеличение скорости взлома на	0,2
Увеличение награды за миссии со взломом на	15%
Повышение максимального ХП на	1
Увеличение скорости перемещения на	10%
Уменьшение времени разгона на	0,3
Увеличение емкости бака с топливом на	20
Увеличение скорости перемещения на	10%
Увеличение скорости перемещения на	15%
Повышение максимального ХП на	1
Уменьшение расхода топлива на	2

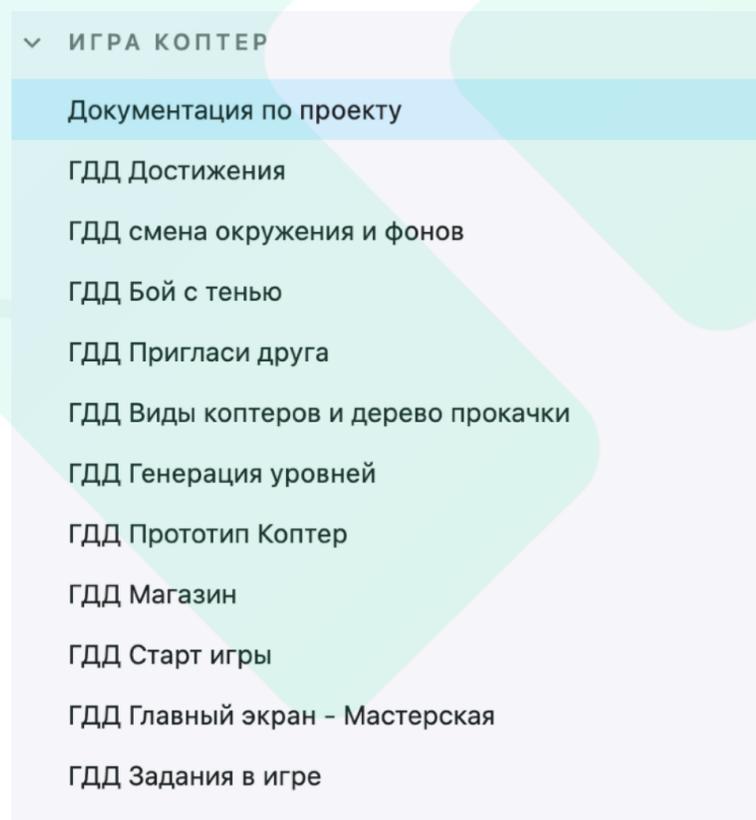


E2-E4!

Саня, это нарды, а не шахматы!

Системный анализ в геймдизайне

Структурирование системной документации по игре



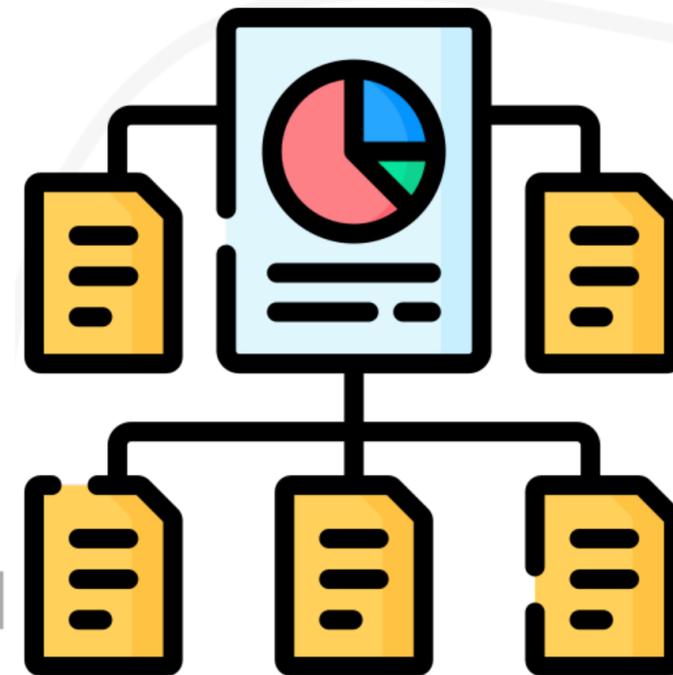
★ Документация по проекту

Прототип

1. Концепт
2. ГДД
3. Фигма арт
4. Фигма схемы
5. Мир схемы
6. Мейл документы
7. Баланс
8. Конфиг - Конфигурации dev_test

Информация по игре:

1. **Название** - Drone Master



Системный анализ в геймдизайне

Системное описание задач разработчикам

2. Матчмейкинг

1. Осуществляется на основе рейтинга игрока
2. Подбираем игроку человек в приоритетном порядке
3. Игрок-соперник выберется если его рейтинг находится в промежутке = $([\text{очки игрока}] - 3 * [\text{Рейтинг -}] ; [\text{очки игрока}] + 3 * [\text{Рейтинг +}])$
4. Поиск игрока осуществляется [20-30] секунд
5. Если другой человек не найден даем соперника ИИ
6. За победу/поражение над ИИ

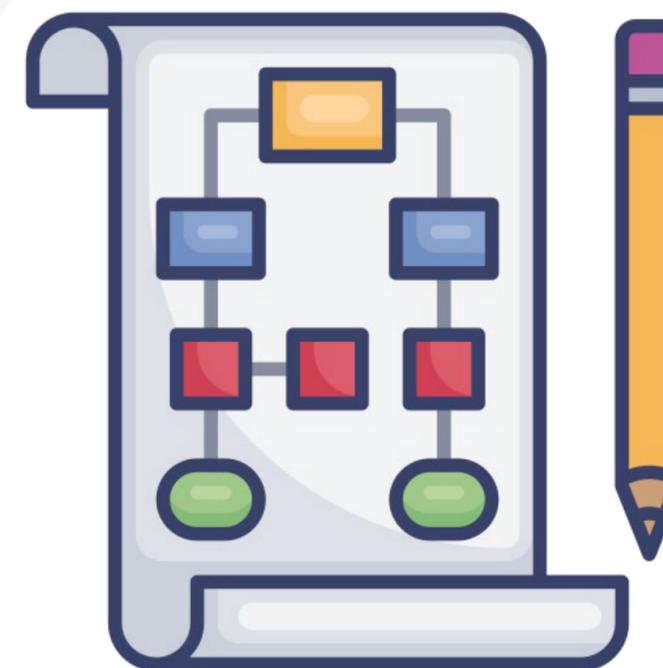
3. ИИ соперник

1. Игрок играет с ИИ на скрытом рейтинге
2. Скрытый рейтинг ИИ
 1. Игрок начинает со сложности [0,25]
 2. За каждую победу над ИИ увеличиваем сложность на [0,05]
 3. За поражение понижаем сложность на [0,025]

А можно
еще раз?

Я пока тебе
рассказывал,
уже сам все
понял!

- Описание объектов
- Правил и условий взаимодействия
- Структуры объектов

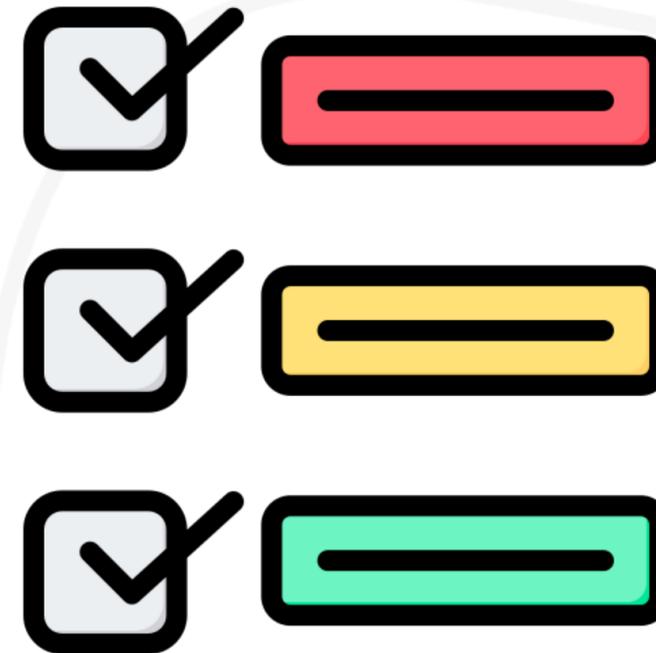


Системный анализ в геймдизайне

Системная приемка задачи

В анимации слайдов - состояния кнопок отрисованы, нужно сделать только само смахивание. Схема смахивания [здесь](#).

- UI выбора достижений - добавить чек бар
- Окно персонажа + текст (макет) - s3://gamedev-art-bucket/Project_tochki/Tutor_pers/
- Главный экран - два значка, без верхних плашек
- Таппер
- Анимация перехода для сцен обучения
- Анимация нажатия кнопок
 - Аватарки
 - i
- Окно предупреждения что не ходит - по середине экрана
 - Заголовок "Внимание!"
 - Текст - "Если вы не будете ходить - вам засчитают поражение!"
- Экраны для выбора анимация кнопок - слайды
- Убрать Организация на экранах



Системная архитектура в играх

Реализация механизмов, сокращающих работу других специалистов

- ▶ Конфигурирование баланса
- ▶ Оптимизация внедряемых артов

Шаг 1: Уменьшение размера текстур

Пример вызова:

```
TARGET_WIDTH=1024 ORIENTATION=ALBUM node resize-images.js
```

Обнаружен язык C

Описание параметров:

TARGET_WIDTH - ширина экрана, под которую собирается билд. Текстуры уменьшаются исходя из того, что исходные текстуры созданы под 4K

ORIENTATION - ориентация игры, либо альбомная (ALBUM), либо портретная (PORTRAIT)

P.S. Иногда возникают ситуации, когда ресайз изображения невозможен в силу особенностей отображения в игре (например когда изображение используется как повторяющаяся текстура и уменьшение размера приведет в увеличению частоты повторений). Изображений, в пути до которых будет встречаться фраза "ignore_resize", не будут ресайзиться. Эта фраза может встречаться как в названии файла, так и в названии папки, в которой все изображения должны игнорировать ресайз.

```
"tier1": {
  "levelGenerationRules": {
    "levelBlocksStepsCount": 3,
    "safeZoneSize": 0.1,
    "obstacleQuantity": {
      "start": {
        "static": 5,
        "damaging": 1
      },
      "inner": {
        "static": 4,
        "damaging": 2
      },
      "finish": {
        "static": 4,
        "damaging": 3
      }
    }
  },
  "copterSafeZone": 0.1
},
"copterConfig": {
  "common": {
    "maxMovementSpeed": 10000,
    "accelerationTime": 3,
    "fuelConsumptionInMotion": 10,
    "fuelConsumptionInIdle": 20,
    "batteryEasy": 120,
    "batteryMedium": 100,
    "batteryHard": 80,
    "maxFuelQuantity": 200,
    "maxHealth": 3
  }
},
```

Системная архитектура в играх

Переиспользуемые компоненты и модули

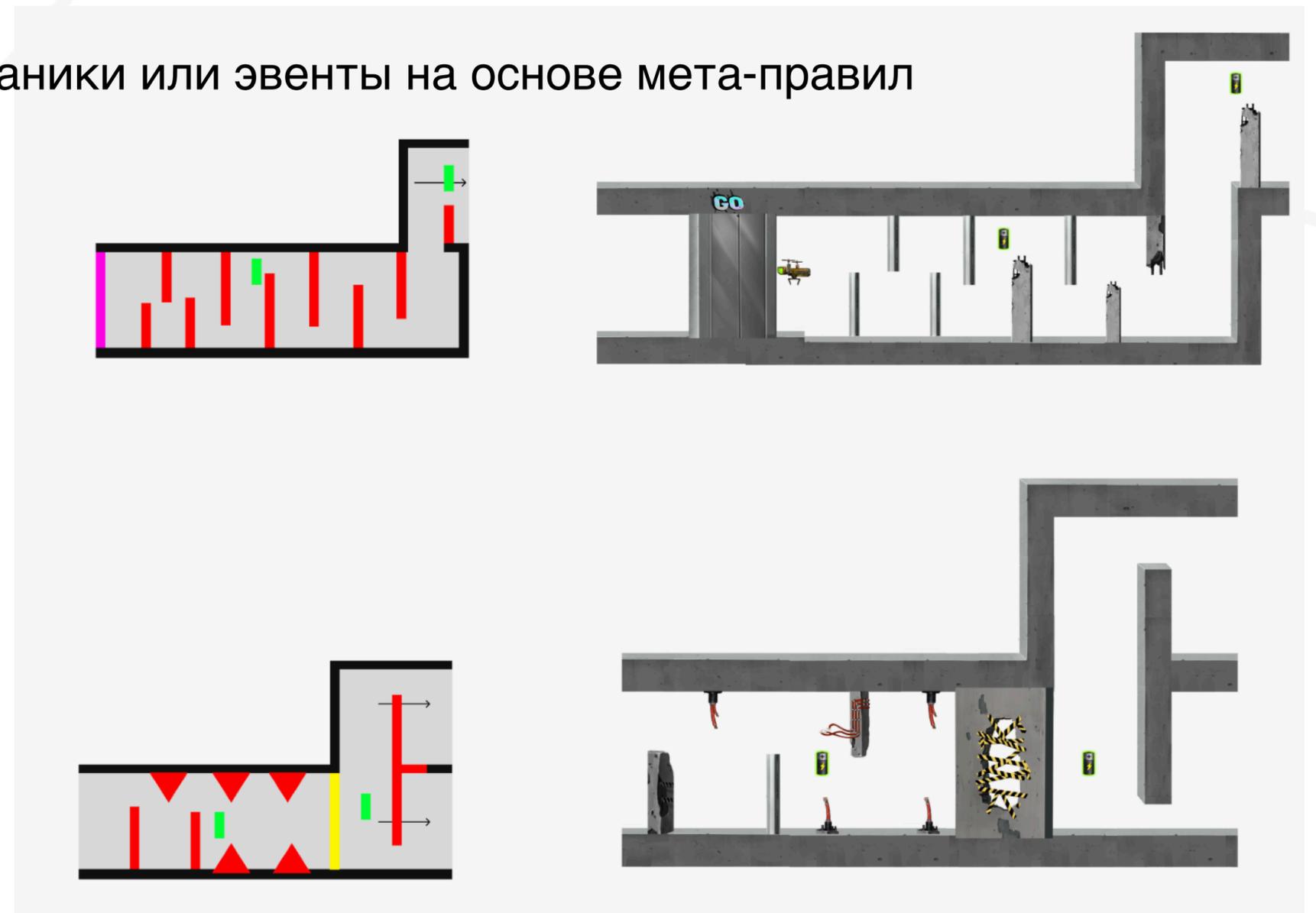
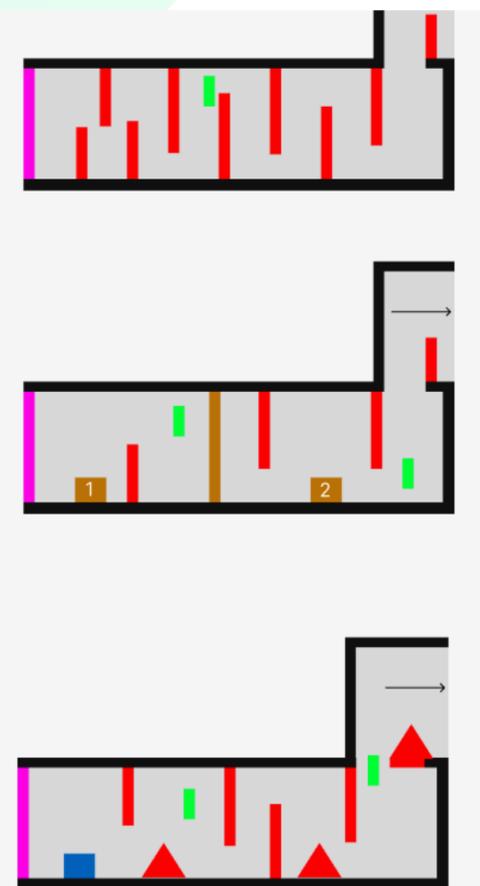
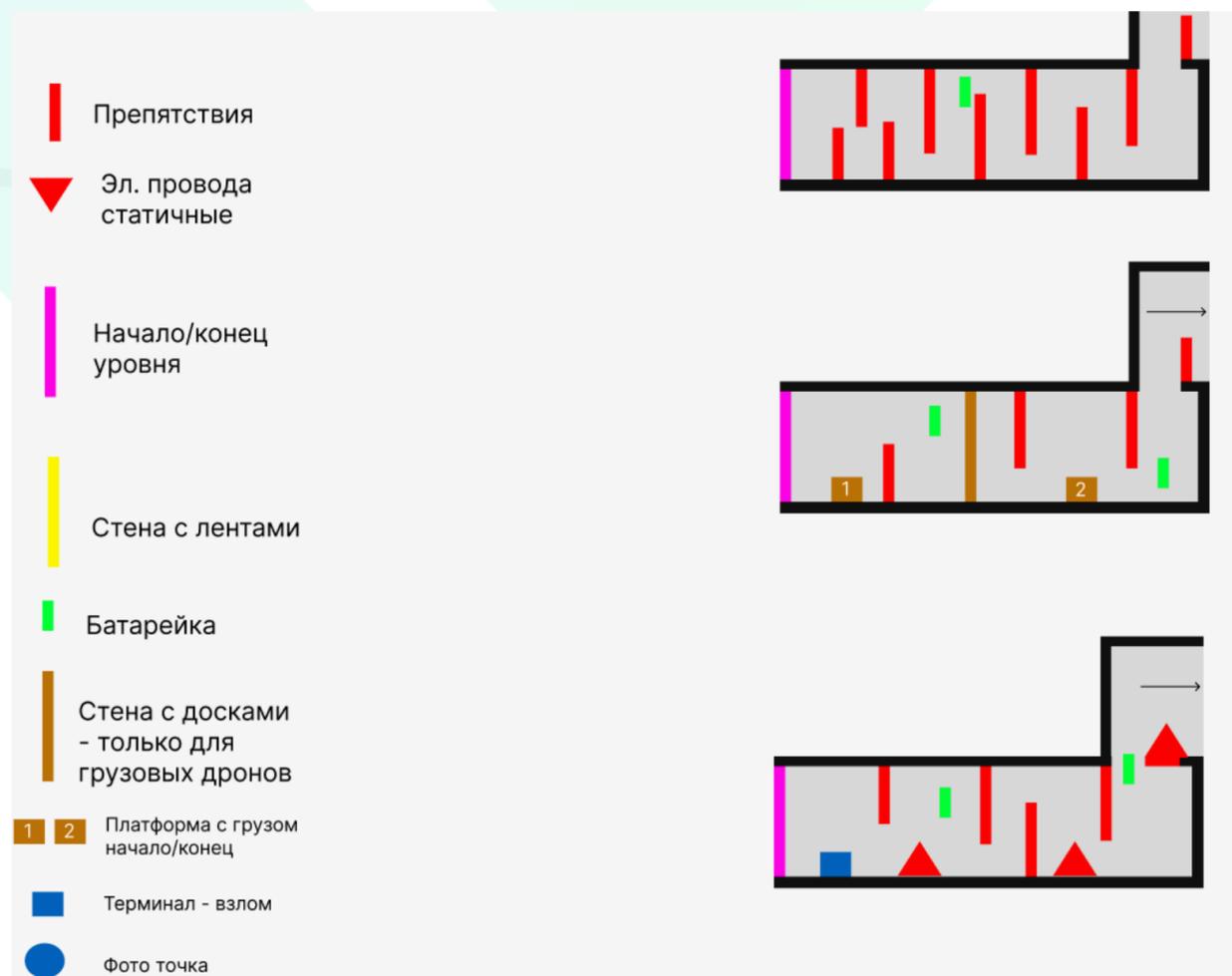
- Дробим игровые механики на модули и собираем из них игры



Системная архитектура в играх

Автоматизация игровых механик

- ▶ Программная генерация уровней
- ▶ Конструкторы, создающие игровые механики или эвенты на основе мета-правил



Геймдев стартап и методологии

Циклы разработки фичей

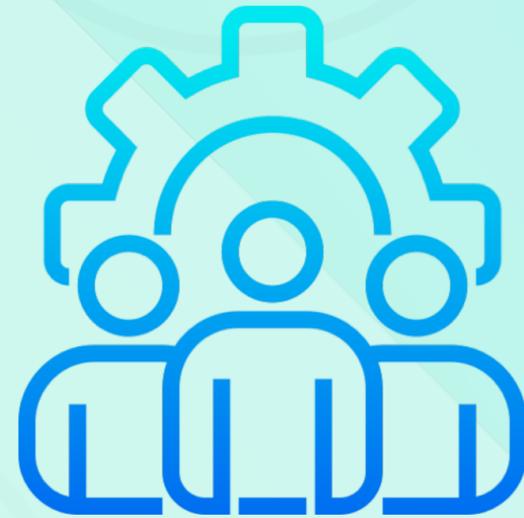


Геймдев стартап и методологии

Ожидания от методологий



Высокая эффективность
работы команды



Самодостаточная и
автономная работа
Масштабирование
команд



Структурирование и накопление
информации по проекту



Четкие, понятные и
прозрачные процессы работы

Методологии разработки

Getting Real & Crystal Clear



Методология для создания стартапа

Когда **ничего нет**,
но результат **нужен быстро**
и при активном росте команды и проекта

Методологии разработки

Задачи

- ▶ **Пренебрегайте деталями в начале**
— делаем до состояния работающей игры
- ▶ **Используйте правило 3F** (fix time, fix budget, flex score) — когда нельзя сделать 3 составляющие идеально, избавьтесь от одного.



Методологии разработки

Задачи

- ▶ **Интерфейс до программирования** — Работа над артами и геймдизайном велась до появления в команде разработчиков
- ▶ **Создавайте ПО для общих решений** — создаем внутренние системы по управлению играми



- Главная
- Статистика
- Игроки



НАЗАД

список конфигов для проекта DroneMaster

ДОБАВИТЬ КОНФИГ

balanceConfig

GenerateLevels

Методологии разработки

Инструменты

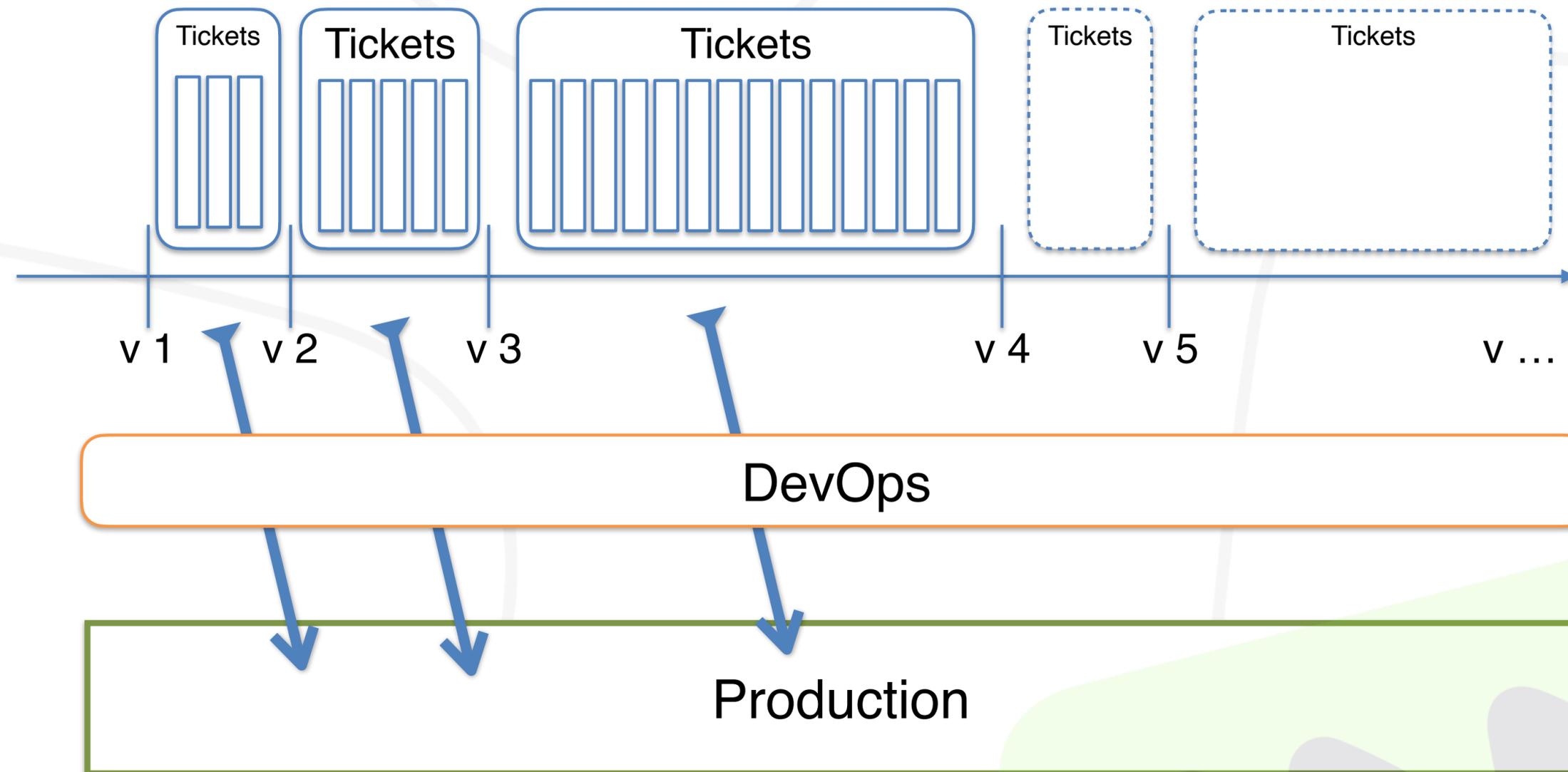
- ▶ **Инструменты, дающие счастье** - сами выбираем инструменты, которые используем



Методологии разработки

Инструменты

► Быстрая доставка продукта



Методологии разработки

Инструменты

▶ Нет функциональным спецификациям

☆ Механизм внедрения зависимостей

Внедрение зависимостей позволяет решить несколько задач:

1. Гарантировать работу паттерна Singleton
2. Менять реализации сервисов в зависимости от окружения, переменных, сцены и так далее
3. При необходимости подменять классы.

Для объявления зависимости в пакете с инструментами присутствует функция `provide()`, у нее есть несколько вариантов вызовов, но независимо от сигнатуры в ней присутствуют два глобальных параметра, это `token` и `scope`.

Параметр `token` это своего рода идентификатор зависимости, ключ, по которому зависимость можно будет найти в слове. Этот параметр может быть строкой, числом, классом и так далее. Не может быть интерфейсом, так как во время выполнения кода в javascript не существует понятия интерфейса.

Параметр `scope` определяет область видимости зависимости. Он помогает определить где зависимость должна существовать. Значение этого параметра по умолчанию равно `"game"`, это означает, что внедренная зависимость будет глобальна и будет доступна из любой сцены.

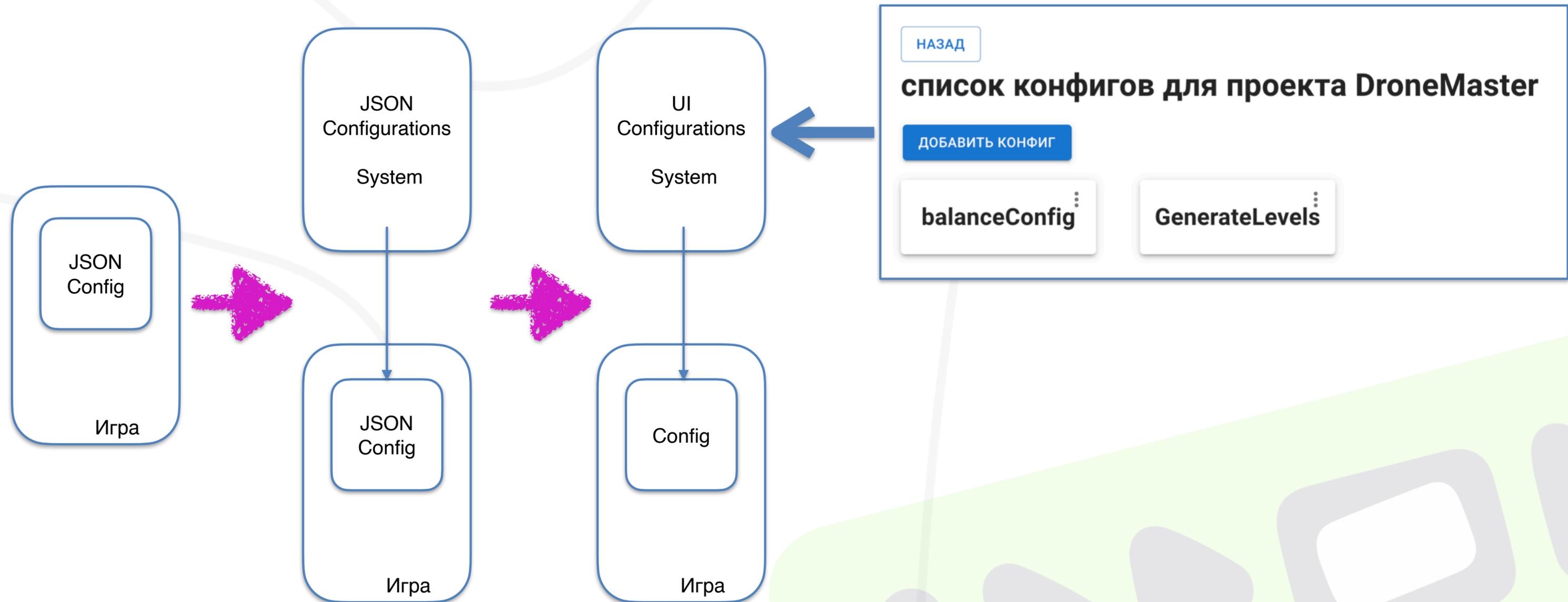
Описание проблемы

Предложение решения с разьяснениями

Методологии разработки

Инструменты

► Инкрементальная смена архитектуры



Методологии разработки

Команда

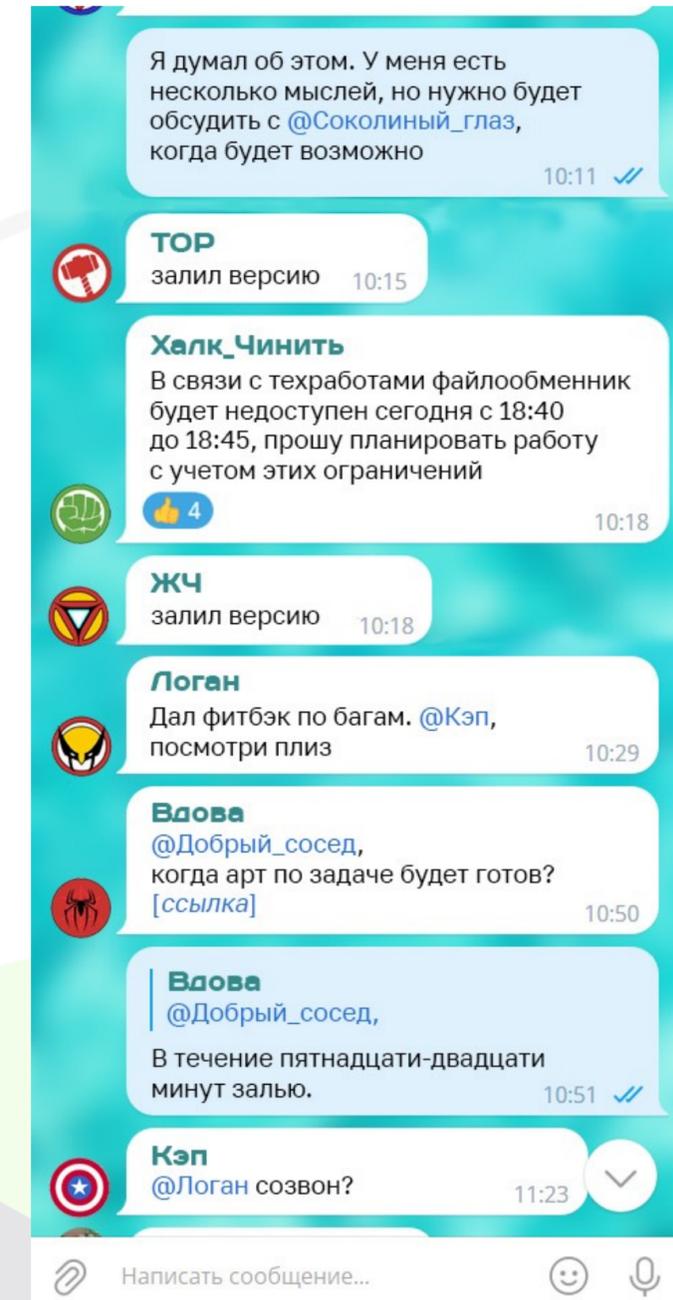
- ▶ **Персонал.** Нанимайте меньше и позже — начали с художников, закончили разработчиком
- ▶ **Оставайтесь маленькими** — небольшая команда — это максимум к сплоченности и эффективности
- ▶ **«Три мушкетера»** — в команде достаточно 3 роли - у нас это **геймдизайнер, художник и разработчик**



Методологии разработки

Коммуникации

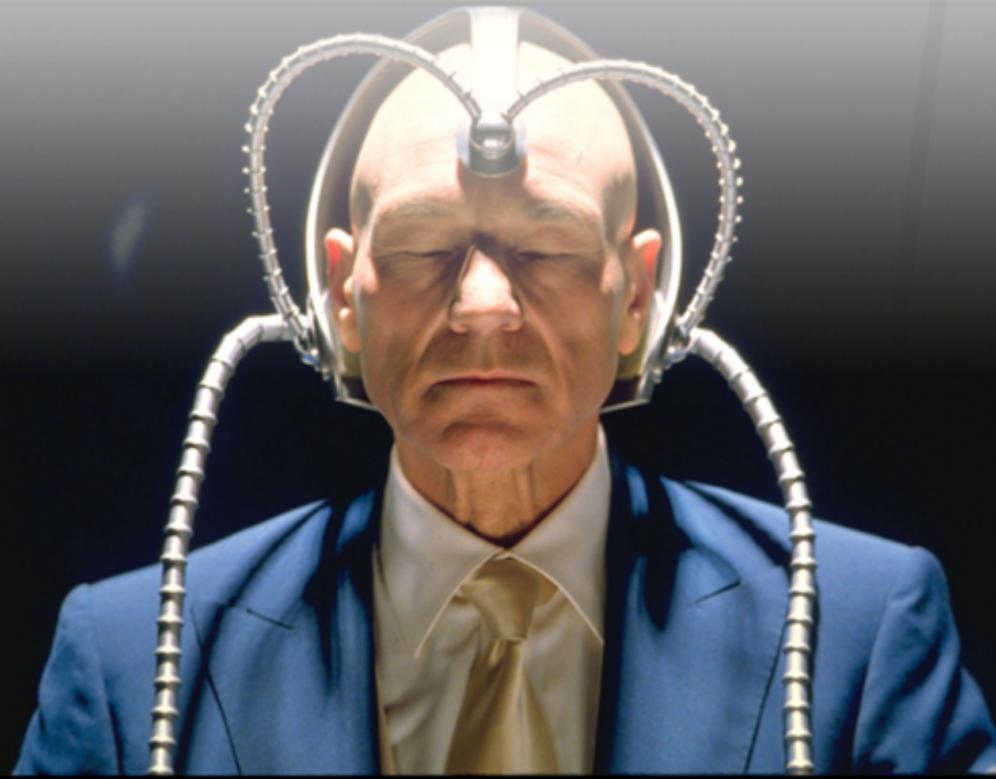
- ▶ **Хорошая коммуникация** - доступная всем и прозрачная коммуникация в общих чатах
- ▶ **Информационные дисплеи** - простой доступ к общей информации и ходу работ по проекту
- ▶ **Исследование 360** - командное рассмотрение идеи по разным направлениям



Методологии разработки

Философия

- ▶ **Легкое название** — **IDDQD** — что может быть проще?
- ▶ **Не тратьте время на проблемы, которых еще нет**
- ▶ **Финансируйте себя сами** — у нас есть инвестор
- ▶ **Делайте ПО для себя** — мы делаем игры, от которых сами фанатеем



Выводы

- ▶ Даже для нулевой точки есть методология
- ▶ Методологии можно миксовать и это нормально
- ▶ Опыт из одной области можно смело применять в другой
- ▶ В геймдизайн из аналитики





Благодарю за внимание
Ваши вопросы?



@zetorg