**Техническая документация для игровых приложений**

**1. Игра "Змейка" (Python/Pygame)**

**1.1 Введение**

2D-игра с управлением змейкой, собирающей еду для роста. Реализована на Pygame с использованием объектно-ориентированного подхода.

**1.2 Требования**

* Python 3.7+
* Pygame 2.0+
* ОС: Windows/Linux/macOS

**1.3 Установка**

bash

Copy

Download

pip install pygame

python snake.py

**1.4 Архитектура**

python

Copy

Download

# Блок-схема основных компонентов

[Главный цикл] → [Обработка ввода] → [Логика движения] → [Проверка коллизий]

↳ [Отрисовка GUI] ← [Генерация еды]

**1.5 Ключевые функции**

* game\_loop(): Основной игровой цикл (60 FPS)
* draw\_snake(): Визуализация змейки
* Коллизии:

python

Copy

Download

if head\_x < 0 or head\_x > WIDTH: # Проверка границ

if any(segment == snake\_head for segment in snake\_list[:-1]): # Самопересечение

**1.6 Ограничения**

* Нет системы очков
* Фиксированная скорость
* Отсутствие меню паузы

**1.7 Расширения**

* Добавить уровни сложности
* Реализовать звуковые эффекты
* Ввести систему бонусов

**2. Онлайн-шахматы (Python/Pygame/Sockets)**

**2.1 Введение**

Клиент-серверное приложение для игры в шахматы по сети. Использует TCP-сокеты для коммуникации.

**2.2 Требования**

* Сервер: Python 3.7+
* Клиенты: Pygame 2.0+
* Сеть: Локальная или VPN

**2.3 Установка**

bash

Copy

Download

# Сервер

python chess\_server.py

# Клиенты (2 экземпляра)

python chess\_client.py

**2.4 Протокол связи**

| **Команда** | **Описание** | **Пример данных** |
| --- | --- | --- |
| GET\_BOARD | Запрос текущего состояния | - |
| MOVE | Отправка хода | ((1,2), (3,4)) |
| BOARD\_UPDATED | Уведомление об изменении | - |

**2.5 Алгоритм валидации хода**

python

Copy

Download

def validate\_move(from\_pos, to\_pos):

piece = board[from\_pos[0]][from\_pos[1]]

if piece == 'wP': # Логика для пешки

return check\_pawn\_move(from\_pos, to\_pos)

# ... аналогично для других фигур

**2.6 Ограничения**

* Нет рокировки и взятия на проходе
* Отсутствует проверка шаха
* Лимит: 2 игрока

**2.7 Безопасность**

* Использовать SSL/TLS для шифрования
* Внедрить аутентификацию
* Добавить логгирование ходов

**3. 2D-платформер (Python/Pygame)**

**3.1 Введение**

Песочница с базовой физикой и управлением персонажем. Поддержка прыжков и коллизий.

**3.2 Системные требования**

* CPU: 1.5 GHz+
* RAM: 512 MB+
* Видео: Интегрированная GPU

**3.3 Физическая модель**

python

Copy

Download

# Формулы движения

vertical\_momentum += gravity \* dt

position.y += velocity \* dt + 0.5 \* gravity \* dt²

**3.4 Конфигурация управления**

| **Клавиша** | **Действие** | **Параметры** |
| --- | --- | --- |
| A/← | Движение влево | Ускорение: 5 px/frame |
| D/→ | Движение вправо | Трение: 0.8 |
| Пробел | Прыжок | Высота: -12 px/s |

**3.5 Коллизии**

python

Copy

Download

# Псевдокод обработки столкновений

for platform in platforms:

if player\_rect.colliderect(platform):

handle\_collision(platform)