# Тестовое задание на позицию

# "Программист Unity3d/C#"

## Основное задание

Необходимо создать игру “Flappy Bird” (см. <https://www.youtube.com/watch?v=fQoJZuBwrkU> ) в Unity3d с использованием Di-контейнера Zenject. Задачу необходимо декомпозировать и декомпозицию предоставить в отчете.

Игра должна работать на PC/Android, сохранять свое состояние между сессиями, в игре должно быть три экрана:

**Главное меню**

1. Старт - Начать игру
2. Очки - Отображается окно с максимально набранным скором и кнопкой “Закрыть”
3. Настройки - Отображается окно настроек, в котором есть кнопка включения/отключения звука и кнопка “Закрыть”

Игра должна сохранять свое состояние между сессиями.

**Игровой процесс**

На экране игрового процесса (после нажатия кнопки “Старт”) - игрок должен видеть “Очки” и на экране должна быть кнопка “Пауза”, при нажатии на которую - экран затемняется, и есть две кнопки - “Продолжить” и “Выход в меню”.

Геймплей должен быть как на ролике выше.

**Поражение**

После поражения игроку отображается окно проигрыша, на котором есть

1. Текущие набранные очки
2. Максимально набранные очки (рекорд)
3. Кнопка “Выход в меню”

Можно использовать свободно доступные арт-ассеты, можно не использовать их совсем и делать все разноцветными квадратными спрайтами.

## Дополнительные задания

Дополнительные задания не являются обязательными к выполнению и расположены не по порядку, можно делать любое их количество.

* Загрузка игровых ассетов (главный герой/препятствия) по сети из AssetBundles
* Загрузка конфигурации игры (сложность/скорость/расстояния/...) по сети, конфигурации должны быть в виде читаемом и понятном для условного “гейм-дизайнера”, который будет игру конфигурировать
* Реализуйте модульное и интеграционное тестирование
* Реализуйте любую классную технологическую “фичу” в свободной форме (без использования готовых ассетов), примеры:
  1. Интересный шейдер, который меняет/анимирует картинку в зависимости от игрового состояния, ввода игрока
  2. Возможность после проигрыша отмотать время назад, и продолжить геймплей
  3. Локальный мультиплеер
  4. Любой другой вариант

## Предоставление результатов и оценка

Результат тестового задания должен быть доступен на любой code-hosting платформе (Github/BitBucket/..) и в виде 2х билдов для Android/PC. Вместе с результатами тестового задания необходимо предоставить отчет, который будет содержать:

* Декомпозиция задачи
* Время, затраченное на выполнение, с детализацией
* Чем вы руководствовались при выборе архитектуры для игры?
* Как вы построили работу с UI, почему именно так?
* Какими паттернами объектно ориентированного дизайна вы воспользовались при разработке проекта?
* Соблюдаются ли в вашем коде, принципы SOLID, если да, какие из них?
* Какие изъяны/технический долг есть в этом проекте с точки зрения разработки?
* Если бы это был коммерческий проект с гораздо большим бюджетом времени на разработку, что бы вы сделали по другому?
* Что вызвало у вас наибольшие сложности при разработке этого задания?
* Если есть что-то важное, что вы хотели бы сказать про это задание, можете написать в свободной форме.

Важные критерии оценки выполнения тестового задания

* Время выполнения
* Качество кодовой базы (безопасное программирование, обработка ошибок, связность, сложность поддержки, ...)
* Архитектура проекта
* Организация и оформление кодовой базы
* Отчет