

Групповой проект - контрольная работа по программированию

Вступление

В качестве контрольной работы за третий модуль вам предлагается объединиться в команды и сделать небольшой консольный проект. Ниже будут описаны какие-то формальные требования, но надо понимать, что:

1. Они могут меняться
2. Так как такое мероприятие проводится первый год, большая часть требований будет сформулирована неформально. Если есть необходимость официально закрепить какие-то из них, нужно сказать мне об этом.

Проект подразумевает самостоятельное обучение некоторым техническим и социальным навыкам, например, работе с консолью linux, формированию рабочей команды, git и так далее. Разумеется, приветствуются консультации с преподавателем (в том числе групповые, очные), инициированные учащимися.

После представления проекта могут быть выставлены три оценки (**константирующая**, **формирующая**, **творческая**), причем одному участнику будут выставлены как минимум две оценки, вторая может зависеть от роли в проекте. Оценка состоит из (групповая) + (личная). Групповая оценка одинакова для всех участников группы, личная зависит от вклада в проект каждого участника.

Стартовая оценка по каждому из типов равна 1.

Да, получение частичных баллов возможно, округление будет известно позже (с очень большой вероятностью будет округление вниз).

Этапы и дедлайны

1. [14.02 - 17.02] Контрольная точка (далее КТ) 1
Формирование группы, распределение ролей в ней, формулирование базовой идеи проекта
2. [21.02 - 24.02] КТ2
Презентация четко сформулированной идеи проекта, итогового состава и заготовки кода
3. [07.03 - 10.03] КТ3
Первый показ кода
4. [24.03/29.03] Итоговая точка
Последняя возможность презентации проекта

Формальные требования

Выполнение формальных требований будет оцениваться в баллах к какой-либо оценке. Цветом будет обозначаться тип оценки, к которой добавляется балл. Если буква заглавная, то балл идет к групповой оценке, иначе - к личной.

- **Уникальность идеи [+0.5Б, +1Б]**

Балл засчитывается, если идея проекта не повторяет существующие известные продукты или является прямым развитием данных

- **Сложность и объем кода [+1Б, +1Б]**

Субъективная оценка, будут смотреться использованные алгоритмы, логика проекта

- **Архитектура [+0.5Б, +0.5Б, +1Б]**

Чтобы получить балл за этот пункт, необходимо в явном виде представить внутреннюю архитектуру разработанного проекта и доказать её отказоустойчивость

- **Баги [+0.5Б, +1Б]**

Ваш проект должен пройти стресс-тестирование на баги и не упасть во время презентации

- **Презентация [+0.5Б, +1Б]**

Во время презентации каждый участник проекта **без страха** высказывается (да, уверенность во время выступления тоже будет учитываться) и находится в курсе происходящего. Также учитывается внешний вид самой презентации

- **Командообразование [+1Б]**

Каждый участник знает свою роль, отвечает только за свою часть проекта

- **UI [+1Б]**

У проекта существует выделяющийся дизайн

- **UX [+1Б]**

У проекта интуитивно понятный интерфейс

- **Внешний вид кода [+1Б]**

Соблюден code-style, есть документация/комментарии

- **Соблюдение дедлайнов [+1Б, +1Б, +1Б]**

Все этапы выполнены и представлены вовремя

Пример работы над проектом

Команда 1: "Arkanoid 2.0"

Участники:

- Разработчик 1 (лидер и разработчик)
- Разработчик 2 (дизайнер)
- Разработчик 3 (второй разработчик)

Идея проекта: Создание консольной версии игры "Arkanoid", где при столкновении с кирпичиком шарик делится на два меньших, что добавляет элемент стратегии и сложности.

Этапы работы:

1. **КТ1 (14.02 - 17.02):** Команда собралась и разделилась на роли: два разработчика и дизайнер. Сформулирована базовая идея проекта (консольная версия "Arkanoid"), начата работа над концепцией игры.
2. **КТ2 (21.02 - 24.02):** В процессе работы команда поняла, что лидер не справляется с ролью руководителя, хоть и является хорошим разработчиком. Дизайнер предложил переработать идею, добавив новый элемент: шарик будет делиться на два при столкновении с кирпичиком. Команда перераспределила роли, и каждый участник взял на себя четкие задачи.
3. **КТ3 (07.03 - 10.03):** К этому времени команда подготовила шаблон проекта с дизайном, но без логики игры. Были четко сформулированы ожидания от каждого участника, а также установлено рабочее время для совместной работы.
4. **Итоговая точка (24.03):** Проект был полностью готов, хотя реализованная идея отличалась от первоначальной. Игра работала без багов и была готова к презентации. Каждый участник уверенно представил свою часть работы, и команда получила высокую оценку за уникальность идеи и качество исполнения.

Формальные требования:

- Дизайнер + руководитель (5Б, 5Б)

- Разработчик 1 (5Б, 5Б)
- Разработчик 2 (4Б, 4Б)

Команда 2: "Проект без названия"

Участники:

- Лидер (отвечает за идеи)
- Разработчик 1
- Разработчик 2

Идея проекта: Изначально команда планировала создать проект, но в итоге осталась без четкой идеи.

Этапы работы:

1. **КТ1 (14.02 - 17.02):** Лидер команды "украл" идею у GPT, но не записал её. В команде был представлен список участников, где лидер гордо обозначил себя как "лидера".
2. **КТ2 (21.02 - 24.02):** Команда забыла идею, и от проекта осталась только название. Лидер не смог ответить на вопросы разработчиков и отложил их, ссылаясь на более важные задачи.
3. **КТ3 (07.03 - 10.03):** К этому времени участники забыли о проекте. Разработчики, осознав, что не могут работать без четкой идеи, ушли в другой проект, оставив лидера одного.
4. **Итоговая точка (24.03):** Лидер попытался оправдаться, говоря, что проект будет, но не сразу. Однако команда не смогла представить ничего конкретного, и итоговая презентация провалилась.

Формальные требования:

- Уникальность идеи: +0Б
- Сложность и объем кода: +0Б
- Архитектура: +0Б
- Баги: +0Б
- Презентация: +0Б
- Командообразование: +0Б
- UI: +0Б
- UX: +0Б
- Внешний вид кода: +0Б
- Соблюдение дедлайнов: +0Б

Итоговая оценка: 0Б

Примеры кода

Можно взять в качестве примера какие-то известные открытые библиотеки или консольные игры:

- [ninvaders](#)
- [Консольный Ray Tracing](#)
- [Gif-visualize](#)