Формирование оценки

Этапы выполнения курсовой контроля

Идея

В курсе балльно-рейтинговая система оценивания. Баллы (0-100) складываются из:

- (индивидуально) Полное прохождение онлайн-курса (0-40 баллов пропорционально степени прохождения, нужный балл в курсе указан в Таблице успехов)
- (группа) Выполнение курсовой (0-60 баллов)

О курсовой:

- Подготовка курсовой работы разбита на отдельные блоки (этапы).
- Работа ведется в github репозитории проекта, доступ в который вам дает преподаватель.
- Вы работаете в репоизтории полностью самостоятельно.
- Результаты этапов сдаются в ветке **main**, сдавать их в виде Pull Request не нужно.
- Этап считается сданным, когда в таблице с текущим контролем он получает соответствующую отметку. Внимательно следите, чтобы эти отметки появлялись - **от этого зависит объем вашей работы**.
- За каждый сданный этап команда получает баллы. Баллы указаны в Таблице успехов.
- У каждого этапа есть срок, когда его необходимо сдать. Этот срок указан в заголовке в Таблице успехов.
    - Если этап сдан без опоздания, то команда получает все баллы за этап.
    - Если этап сдан с опозданием от необходимого срока, то команда получает половину (50%) от баллов за этап.
- Фраза **в виде вики-страницы**, означает, что задание сдается в виде вики-страницы в репозитории проекта. Иные варианты сдачи (аттач в письме, файл в репо, устно, в виде песни или танца ….) - не принимаются.

!! Внизу указано больше этапов, чем вам нужно выполнить. Не делайте лишней работы. !!

Общие советы по курсовой работе

Смысл данного курса - **разработать приложение с перспективы потенциального пользователя**. Это означает, что первичны вопросы (и оцениваются именно они):

- Удобства и понятности пользовательского интерфейса / сценария,
- Универсальности и стабильности работы приложения.

Следующие вопросы вторичны:

- Бакенды
- Модели машинного обучения,
- Базы данных.

Проверка стабильности работы приложений

Проверка приложений на стабильность работы будет выполнятся вручную и автоматизировано. Для автоматизированной проверки будет использован скрипт https://bitbucket.org/mark_zaslavskiy/adfmp/src/master/monkey.sh

Эмулятор, на котором будут проверятся работы

Экран

- hw.lcd.width: 480
- hw.lcd.height: 800
- hw.lcd.density: 240

Прочее

- CPU/ABI: Google APIs Intel Atom (x86)
- Target: google_apis [Google APIs] (API level 30)
- SD Card: 512M
- fastboot.chosenSnapshotFile:
- runtime.network.speed: full
- hw.accelerometer: yes
- hw.device.name: Nexus S
- hw.lcd.width: 480
- hw.initialOrientation: Portrait
- image.androidVersion.api: 30
- tag.id: google_apis
- hw.mainKeys: yes
- hw.camera.front: emulated
- hw.gpu.mode: auto
- hw.ramSize: 343
- PlayStore.enabled: false
- fastboot.forceColdBoot: no
- hw.cpu.ncore: 2
- hw.keyboard: yes
- hw.sensors.proximity: yes
- hw.dPad: no
- hw.lcd.height: 800
- vm.heapSize: 48
- skin.dynamic: yes
- hw.device.manufacturer: Google
- hw.gps: yes
- hw.audioInput: yes
- image.sysdir.1: system-images/android-30/google_apis/x86/
- showDeviceFrame: yes
- hw.camera.back: virtualscene
- hw.lcd.density: 240
- hw.arc: false

- hw.device.hash2: MD5:cacc00877ea1aeafae04c1b3addc16f7
- fastboot.forceChosenSnapshotBoot: no
- fastboot.forceFastBoot: yes
- hw.trackBall: no
- hw.battery: yes
- hw.sdCard: yes
- tag.display: Google APIs
- runtime.network.latency: none
- disk.dataPartition.size: 800M
- hw.sensors.orientation: yes
- avd.ini.encoding: UTF-8
- hw.gpu.enabled: yes

Список этапов

Макет и сценарий использования

Руководство

- Продемонстрирована работа Android Studio на вашем компьютере (в формате микроскринкаста работы helloworld).
  - Скринкаст выложен в репо и / или на него дана ссылка
  - Код приложения helloworld выложен в репозиторий (просто для проверки что есть доступ)
- Сформулированы реализуемые сценарии использования в виде **вики-страницы**.
- Нарисован макет пользовательского интерфейса в виде графа (по аналогии с прошлым семестром) и **выложенн в репозиторий**, макет показан на **вики-странице** со сценариями.

UI на заглушках

Результат:

- код приложения выложен в репозитории,
- есть .gitignore для Android Studio, в котором в числе прочего полностью добавлен каталог .idea (каталога .idea не должно быть в репо также как и промежуточных артефактов сборки),
- приложение собирается и запускается на эмуляторе, обозначенном выше,
- переходы работают, но данные отображаются только те, что захардкожены в элементах UI, приложение не падает с exception в ответ на любые действия пользователя.
- пакет приложения называется согласно теме курсов (использование названия по умолчанию или не информативного названия будет ошибкой).
- в приложении есть экран About, где указаны авторы.

Частично работоспособный UI

Результат:

- выполнены требования “UI на заглушках”,
- код приложения выложен в репозитории,
- если для работы приложения нужны secrets, ключи или иные чувствительные данные, то авторы должны их предоставить в письме
- если для сборки требуются нетривиальные действия (что-то кроме Run), то авторы должны подготовить в README.md инструкцию (в целом, желательно избегать дополнительных шагов)
- если в приложении есть механизм регистрации, то авторы должны добавить в приложение тестовый аккаунт и указать его данные в README.md
- приложение собирается и запускается на эмуляторе, обозначенном выше,
- UI позволяет вводить пользовательские данные,
- реализовано не менее одного сценария использования,
- в приложении есть или реальные, или демо данные. В последнем случае, данных должно быть достаточно для демонстрации реализованных сценариев использования.

Оценка сложности пользовательского интерфейса вашего приложения

Руководство по измерению последовательности действий и оценки сложности UI

Результат: **вики-страница**

- с таблицой подсчета количества действий (суммарным),
- количеством действий по каждому виду взаимодействия (кликов/вводов текста/ нажатий на апп.кнопки и пр.), иллюстрирующие подсчет скриншоты.
- выводом о том, как можно упростить последовательность.
- макетом интерфейса, реализующим предыдущий пункт.
- аналогичным подсчетом количества действий **для ближайшего аналога**

Окончательная версия приложения

Результат:

- выполнены требования от “Частично работоспособный UI” http://se.moevm.info/doku.php/staff:courses:application_development_for_mobile_platforms:mark#%D1%87%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%BE_%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BE%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_ui
- код приложения выложен в репозитории, его можно скачать, собрать и запустить. При этом выполняются все сценарии использования, приложение работает стабильно.
- У приложения есть иконка, корректное название (согласно теме).
- В приложении есть или реальные, или демо данные. В последнем случае, данных должно быть достаточно для демонстрации всех сценариев использования.

Создание юнит-тестов для приложения

Результат:

- в репозитории выложены файлы юнит-тестов (не менее 3х TestCase) для основных классов, которые можно запустить стандартным способом через Android studio,
- в репозитории настроен автоматический запуск юнит-тестов по коммитам через Github actions.

Если вам кажется, что для вашего приложения юнит-тесты не сделать - это означает, что либо оно еще слишком сырое (в нем только заглушки), либо что вы не отделили бизнес-логику от интерфейсов.

Пояснительная записка

1. Пояснительная записка в электронном и бумажном виде (требования http://se.moevm.info/doku.php/staff:courses:application_development_for_mobile_platforms:course_work)
    1. Выложена в репозиторий в doc(x)/odt + pdf
    2. Соответствует требованиям оформления ВУЗа.
    3. Есть непустой список литературы.
    4. Нет разделов без текста.
    5. Все таблицы, рисунки и схемы имеют подпись.
    6. В списке литературы указана ссылка на ваш открытый репозиторий в github/bitbucket.
    7. Бумажный вариант прошит или скреплен.

Создание интеграционных тестов для приложения

Интеграционные тесты == espresso-тесты (каюсь, термин не совсем удачный)

Результат:

- в репозитории выложены файлы интеграционных тестов для основных сценариев использования, которые можно запустить стандартным образом (как Android InstrumentedTest), либо скрипт для запуска (например, если это тесты для игры);
- тесты стабильно выполняются при нескольких запусках подряд на эмуляторе, обозначенном выше;
- тесты проверяют работу приложения преимущественно через его UI;

Если в вашем приложении сложно писать интеграционные тесты, возможно, у вас есть проблемы с UI (он плохо показывает состояния приложения) и/или с архитектурой.

Материалы для публикации

Подготовьте материалы, необходимые для публикации приложения в Play Market.

- краткое описание, 80 символов;
- полное описание, не более 4000 символов;
- иконка(512 x 512 32-bit PNG (with alpha));
- Feature Graphic 1024 w x 500 h, JPG or 24-bit PNG (no alpha);
- три скриншота.

Материалы необходимо выложить в репозитории в каталог **play_market_publication/**

Формирование оценки ЛЭТИ

Допуск (зачет)

Необходимые условия (ЛЭТИ): Чтобы претендовать на оценку выше “Не аттестован”,

- в репозитории проекта должно быть не менее (10 / 15 / 20 - Удовл. / Хор / Отл) коммитов в абсолютном выражении в **КОД ПРОЕКТА**, созданных участником.
  - Комииты в README, wiki, загрузка картинок к коду проекта не относятся:(
  - Аргументы из серии “у меня не было доступа и за меня пушил коллега по команде” / “я неправильно настроил гит / ссш” не принимаются :(
- участник должен полностью пройти онлайн-курс;
- у группы должно быть сдано задание “Пояснительная записка”.

Дедлайны проверок

Все время - Московское.

- Мягкий дедлайн - 28.03.2023 23:59
- Жесткий дедлайн - 30.03.2023 14:00
- Срок окончательной проверки 31.03.2023 14:00

Как работают дедлайны:

- Все присланное до **мягкого дедлайна** будет проверено до наступления **жесткого дедлайна**
- Все присланное до **жесткого дедлайна** будет проверено до **срока окончательной проверки**
- Все присланное после **жесткого дедлайна** будет проигнонировано.

Оценка

Оценка выставляется только при получении зачета (см. выше).

Как баллы трансформируются в оценки (5-балльная система):

- <60 — Неудовлетворительно
- >=60 && <70 — Удовлетворительно
- >=70 && <80 — Хорошо
- >=80 — Отлично

Формирование оценки ВШЭ / ИТМО

Дедлайны проверок

Все время - Московское.

- Мягкий дедлайн - 28.03.2023 23:59
- Жесткий дедлайн - 30.03.2023 14:00
- Срок окончательной проверки 31.03.2023 14:00

Как работают дедлайны:

- Все присланное до **мягкого дедлайна** будет проверено до наступления **жесткого дедлайна**
- Все присланное до **жесткого дедлайна** будет проверено до **срока окончательной проверки**
- Все присланное после **жесткого дедлайна** будет проигнонировано.

Оценка

Как баллы трансформируются в классические оценки (5-балльная система):

- <60 — Неудовлетворительно
- >=60 && <75 — Удовлетворительно
- >=75 && <90 — Хорошо
- >=90 — Отлично

Как баллы трансформируются в оценку по 10-балльной системе для ВШЭ - пропорционально требованиям ВШЭ:

- Если у вас оценка из диапазона Удовлетворительно, то в баллы для ВШЭ она считается пропорциоально диапазону >=4 && <6
- Если у вас оценка из диапазона Хорошо, то в баллы для ВШЭ она считается пропорциоально диапазону >=6 && <8
- Если у вас оценка из диапазона Отлично, то в баллы для ВШЭ она считается пропорциоально диапазону >=8 && <10

From:
http://se.moevm.info/ - **se.moevm.info**

Permanent link:
http://se.moevm.info/doku.php/staff:courses:application_development_for_mobile_platforms:mark?rev=1675322095

Last update: **2023/02/02 07:14**