

Формирование оценки

Этапы выполнения курсовой контрольной

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 репозитории 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 выполняться вручную и автоматизировано.

Вручную приложение будет собираться **стандартным** способом (Import project / Run) через

Android Studio Hedgehog | 2023.1.1 Patch 1

и эмулятор, указанный ниже.

Если вы используете какие-либо нестандартные подходы к разработке / фреймворки / языки и тд (отличные от штатных технологий, которые может без доп. настроек переварить среда разработки выше), то ваша обязанность предоставить dockerfile, в котором будут настроены все зависимости и можно будет штатно (см. выше) собрать и запустить проект.

Для автоматизированной проверки будет использован скрипт https://bitbucket.org/mark_zaslavskiy/adfmp/src/master/monkey.sh

Эмулятор, на котором будут проверяться работы

Экран

- hw.lcd.density 560
- hw.lcd.height 3120
- hw.lcd.width 1440

Прочее (Версия API в эмуляторе 30)

- image.androidVersion.api 30
- avd.ini.displayName Pixel 6 Pro API 30
- avd.ini.encoding UTF-8
- AvdId Pixel_6_Pro_API_30
- disk.dataPartition.size 2G
- fastboot.chosenSnapshotFile
- fastboot.forceChosenSnapshotBoot no
- fastboot.forceColdBoot no
- fastboot.forceFastBoot yes
- hw.accelerometer yes
- hw.arc false
- hw.audioInput yes
- hw.battery yes
- hw.camera.back virtualscene
- hw.camera.front emulated
- hw.cpu.ncore 2
- hw.device.hash2 MD5:a8abfd3536f3d35e4ba2041a7b99f40e
- hw.device.manufacturer Google
- hw.device.name pixel_6_pro
- hw.dPad no

- hw.gps yes
- hw.gpu.enabled yes
- hw.gpu.mode auto
- hw.initialOrientation Portrait
- hw.keyboard yes
- hw.mainKeys no
- hw.ramSize 1536
- hw.sdCard yes
- hw.sensors.orientation yes
- hw.sensors.proximity yes
- hw.trackBall no
- image.sysdir.1 system-images/android-30/google_apis/x86/
- PlayStore.enabled false
- runtime.network.latency none
- runtime.network.speed full
- showDeviceFrame yes
- skin.dynamic yes
- tag.display Google APIs
- tag.id google_apis
- vm.heapSize 384

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**, где также размещены сценарии использования приложения.

UI на заглушках

Результат:

- в репозитории установлен тег 0.5
- 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 и AS, обозначенных 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:

- в репозитории установлен тег 0.8
- 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 таблицей 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**
- вывод по итогам сравнения с аналогом (Кто удобнее)

Примечание: вывод о том, что интерфейс упрощать не надо (так как он лучше аналога / по другим причинам) использовать нельзя :) Проявите фантазию (от вас не требуется эти фиксы реализовывать, достаточно только изобразить и кратко описать)

Окончательная версия приложения

Результат:

- в репозитории установлен тег 1.0
- 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:

- тег unit
- 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 виде (
 1. Требования
http://se.moevm.info/doku.php/staff:courses:application_development_for_mobile_platforms:course_work . Если вы какие-то задания не сделали и вас устраивает текущая оценка, то вы можете пропустить соответствующие разделы (задания для которых вы не сделали) в записке.
 2. Выложена в репозиторий в doc(x)/odt + pdf (в каталог docs, формат названия report_ФАМИЛИИ.)
 3. Соответствует требованиям оформления ВУЗа.
 4. Есть непустой список литературы.
 5. Нет разделов без текста.
 6. Все таблицы, рисунки и схемы имеют подпись.
 7. В списке литературы указана ссылка на ваш открытый репозиторий в github/bitbucket.
 8. Бумажный вариант прошит или скреплен.

Создание интеграционных тестов для приложения

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

Результат:

- `тег integrationtests`
- 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

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 - 27.03.2024 23:59

- Жесткий дедлайн - 28.03.2024 14:00
- Срок окончательной проверки 29.03.2024 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 проигнорировано.

Оценка

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

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

- <70 — Неудовлетворительно
- >=70 && <80 — Удовлетворительно
- >=80 && <90 — Хорошо
- >=90 — Отлично

From:

<http://se.moevm.info/> - **se.moevm.info**

Permanent link:

http://se.moevm.info/doku.php/staff:courses:application_development_for_mobile_platforms:mark



Last update: **2024/04/10 10:18**