Список проектов

1. Помощник фотографа

Приложение облегчает создание красивых фотографий путем отображения сеток поверх видоискателя, то есть перед тем, как делать кадр, фотограф сможет совместить объекты в кадре с линиями сетки (центром кадра, центральным квадратом/прямоугольником, спиралью Фибоначчи и тд).

Функции: Рисование реального уровня горизонта и вертикали поверх кадра. Отображение сеток поверх кадра по схемам:

- http://fototips.ru/teoriya/zolotoe-sechenie-i-proporcii-fibonachchi/
- http://fototips.ru/praktika/9-sovetov-po-kompozicii-ot-stiva-makkarri/
- http://fototips.ru/praktika/kak-rabotat-s-kompozicionnymi-liniyami/

Реализация под IOS https://github.com/moevm/adfmp18-photo-assistant

Необходимо реализовать аналог на Android+Kotlin.

Ожидаемый результат: Android приложение на Kotlin.

2. Кастомный клон игры Каркассон

Реализуйте клон игры Каркассон Правила: https://gaga.ru/gaga/files/pdf/rules/ru/1.pdf Описание: https://gaga.ru/game/carcassonne/

Что необходимо изменить: - плитки шестиугольные - картинки для игры http://pousse.rapiere.free.fr/tome/ Мультиплеер - только hotseat

Ожидаемый результат: Android приложение на Kotlin.

3. Озвучивание активности на гитхабе

Цель: Необходимо реализовать аналог приложения http://listen.hatnote.com/ для актуальной статистики Github в реальном времени.

Задачи: - получение актуальной статистики из github в реальном времени - воспроизведение / синтез звуков - работа в фоновом режиме

Ожидаемый результат: веб-приложение на языке Python3, которое на бакенде получает данные о статистике Github, а на фронтенде - визуализирует и озвучивает её.

4. Автоматизация работы преподавателя в курсах по программированию и информатике

Требуется реализовать веб-приложение, позволяющее:

- Авторизоваться через Stepik
- Проверить факт прохождения курсов на Stepik у студентов
- Графики среднего прогресса группы (количество решенных всеми задач/количество всех заданий курса)*количество человек в группе.
- Графики среднего прогресса группы по каждой лабораторной работе.
- Статистика по прохождению студентами курса. Статистика включает: дату первого удачного решения на степик, дату первого неудачного решения на степик, даты первого решения для каждой лабораторной работы.

Список студентов указывается через ссылку на гугл-таблицу в конфиг файле, список курсов - через конфиг файл.

Ожидаемый результат: веб-приложение на языке Python3, которое позволяет посмотреть статистику по студентам и курсам.

5. Информационная система кафедры: учёт студентов

Требуется расширить функционал компонентов информационной системы кафедры, которые обеспечивают:

- Управление студенческими данными
- Импорт/экспорт данных
- Рейтинги по предметам, курсовые, практики.
- Генерация отчетов.

Ожидаемый результат: веб-приложение на языке Python3/Flask + MongoDb.

6. Информационная система кафедры: научная работа

Требуется портировать на Flask и расширить функционал компонентов информационной системы кафедры, которые обеспечивают возможность ввода/обработки/хранения/предоставления информации о:

- публикациях/научных трудах (группировка по годам/людям)
- участии (достижениях) в конференциях/конкурсах/НИОКР и т.д.
- численности аспирантов, докторантов, соискателей и стажеров
- Также приложение должно обеспечивать разные уровни доступа к информации для различных пользователей.

Ожидаемый результат: веб-приложение на языке Python3/Flask + MongoDb.

7. Информационная система кафедры: индивидуальный план преподавателя

Требуется портировать на Flask и расширить функционал компонентов информационной системы кафедры, которые обеспечивают управление индивидуальными планами

преподавателей:

- Планирование учебной нагрузки
- Переподготовка и повышение квалификации
- Участие в мероприятиях, конференциях,
- Руководство НИР
- Генерация отчетов
- … и другие параметры индивидуального плана

Ожидаемый результат: веб-приложение на языке Python3/Flask + MongoDb.

8. Автоматический вырезатель пауз из видео

Очень часто при записи онлайн-курсов авторы видео делают много пауз для того, чтобы собраться с мыслями/обдумать следующее слово. Задача - найти и удалить их.

Цель - разработать инструмент, автоматизирующий монтаж видео для онлайн-курсов.

Требования: Linux, Python

Задачи и Технические требования - поддерживаемые форматы - TS, MKV - разметить паузы (по звуковой дорожке) длинной более N мс в видео, дать монтажный план - визуализировать вырезаемое - смонтировать видео без пауз

Ожидаемый результат: консольное / Gui приложение для Linux на Python3.

9. Расширение Chrome - вспомогательные инструменты по Stepik

Цель: создать вспомогательные инструменты для администраторов курсов на Stepik.

Задачи: - создание расширения для Google chrome - внедрение своего кода в HTML Stepik - работа со stepik API

Фичи: - показывать в комментарии ссылку на последнее неудачное решение автора, его прогресс по курсу, даты последних решений - график, статистика решений

Требования: HTML, JS, CSS

Ожидаемый результат: расширение для Google Chrome, внедряющее дополнительную информацию на страницу.

10. Генератор контрольных на Stepik + составитель черновиков учебных программ по Stepik

Требуется реализовать веб-приложение, позволяющее:

- Авторизоваться через Stepik
- Сгенерировать контрольную работу по заданной теме
- Составлять черновик учебной программы по степам с описанием

- Обновлять курс на степик

Ожидаемый результат: веб-приложение на языке Python3.

11. Scratch

Цель: сделать форк https://scratch.mit.edu/projects/editor/?tip_bar=home с возможностью использования в MOOC обучении.

Задачи: - разобраться, где живут исходники (https://github.com/LLK/) - обернуть код в Docker-контейнер - добавить интерфейсы для создания задач и создать несколько задач - добавить интерфейсы для оценивания пользователей - добавить интерфейсы для подключения к Stepik через Grader*

Требования: NodeJS, Linux

Ожидаемый результат: NodeJS приложение с интерфейсами для создания задач.

12. Online курс-тренажер для задач на языке Си

Цель: разработка задач для курса на платформе Stepik “Тренажер для языка Си”.

Варианты задач:

- Задачи на синтаксис
- Задачи на простейшие алгоритмы
- Задачи на массивы, линейные списки
- Задачи на сборку программ
- …

Ответы к задачам должны быть запушены в отдельный приватный репозиторий, степы - в публичный.

Ожидаемый результат: набор автоматически проверяемых заданий, интегрированных в Stepik.

13. Простой планировщик туристической активности

Приложение, которое составляет подробный план отпуска (с возможностью оффлайн-работы) в незнакомом городе/стране с учетом интересов пользователя, бюджета на отпуск, местоположения отеля, параметров начала/конца отпуска, желаемого темпа, климатических условий.

Ожидаемый результат: веб-сервис на языке Python3.5/Flask + MongoDb с функциями: регистрации, планирования, корректировки и сохранения планов отпуска.

14. Бакэнд для курса по информатике

Цель: разработка задач для курса на платформе Stepik “Информатика”.

Варианты задач:

- Интерактивные задачи, в которых визуализируется положение некоторого робота на прямоугольной сетке + препятствия + объекты. Робот должен обойти препятствия или собрать какие-то объекты в сетке. Вариант решения может быть на Си и Python. Робот может шагать в любом направлении на одну клетку, видеть объекты с дальностью 1 клетка.
- Задача для Linux для моделирования работы машины Тьюринга. Решение может быть на Python.
- Задача для Linux для моделирования работы указателей реального режима. Решение может быть на Python.
- …

Ответы к задачам должны быть запушены в отдельный приватный репозиторий, степы - в публичный.

Ожидаемый результат: набор автоматически проверяемых заданий, интегрированных в Stepik.

From:
https://se.moevm.info/ - **МОЭВМ Вики [se.moevm.info]**

Permanent link:
https://se.moevm.info/doku.php/courses:mse:2018:project_list

Last update: