 Название работы: Битвы ботов

Заказчик: Антон

Руководитель: Антон

Консультант: Антон

Описание:

Платформа для проведения соревнований ботов по математическим играм.

Должна предоставлять [API](https://ru.wikipedia.org/wiki/API) для реализации различных игр (TODO требования к API и играм).

Должна предоставлять независимый от игры функционал, связанный с проведением соревнований:

* Добавление/удаление ботов в турнир
* Установка ограничений на ход (TODO какие)
* Подсчет и оформление результатов (TODO какие)
* Et Cetera

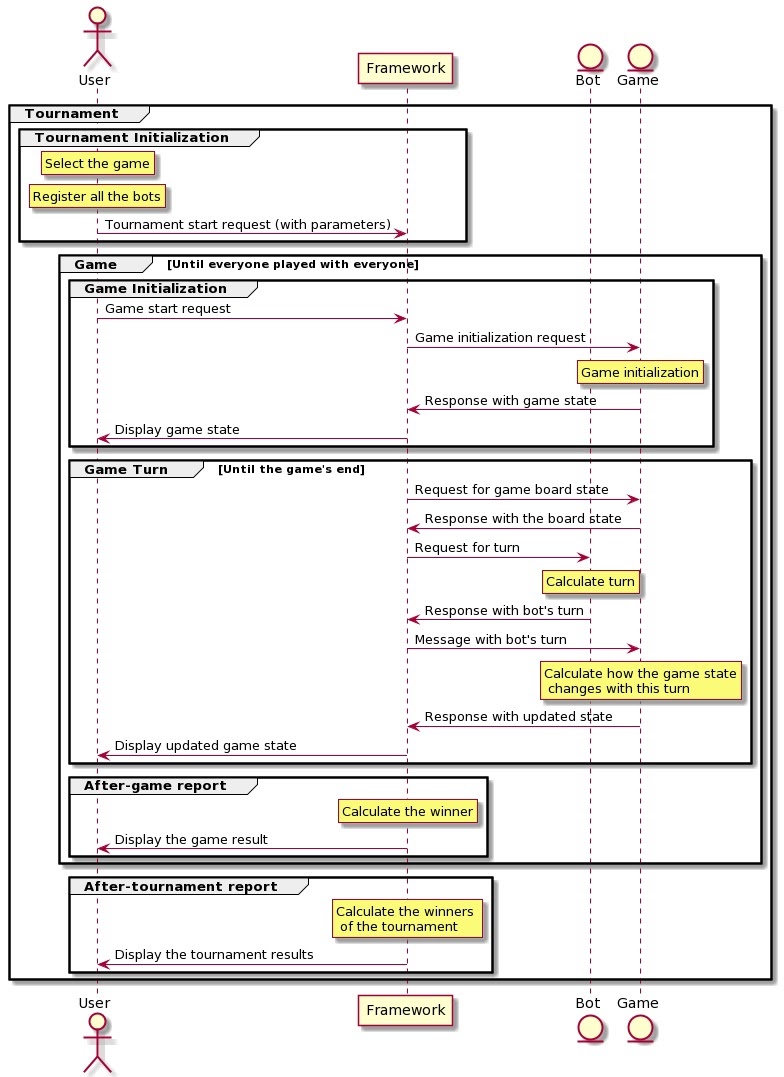
На спринт:

* Относительный импорт Game (только игр) - сделать функцию, которая возвращает список классов игр из модулей в папке ./game
  + Сделать импорт класса независимо от его имени и положения в коде
* Вставить диаграмму активности в ТЗ
  + Описать требования к функционалу которые из этого следуют
* Подумать над пользовательским интерфейсом
* Подумать, что нужно для турнира
  + Написать proof-of-concept класса Tournament
* Начать рефакторинг кода

**API:**

Игры - реализация класса (TODO - описать класс)

Боты - исполняемые файлы, общение с которыми происходит через стандартный ввод-вывод.



**Режимы работы программы**

**Турнир**

1. Выбор игры.
2. Регистрация ботов в турнир.
3. Игра
   * 1. инициализация игры между одной из пар ботов.
     2. процесс игры
     3. результат игры
4. Результаты турнира

Режим работы 1 - Выбор игры

1. Все классы игр в папке game импортируются в программу, и выводятся в виде списка **(уточнить)** пользователю на выбор.
2. После выбора игры начинается следующий режим

Режим работы 2 - Регистрация ботов в турнир.

1. Пользователь по очереди вводит полные пути до файлов ботов, тем самым добавляя их в турнир.
2. Когда нужные боты были импортированы пользователь нажимает кнопку, новый режим работы начинается

Режим работы 3 - Игра

1. Из списка ботов формируется турнирная сетка **(уточнить).**

Инициализация игры

1. Выбирается пара ботов и отправляется запрос на инициализацию игры.
2. game возвращает начальное состояние игры и рисунок поля.

Процесс игры

1. Платформа запрашивает состояние игрового поля, игра возвращает его.
2. Платформа отправляет боту, ход которого сейчас идет, состояние игрового поля. Бот возвращает ход, который он хочет сделать.
3. Платформа отправляет ход game, она меняет параметры поля и состояние игры.
4. Отрисовывает новое состояние поля.
5. Повторяется до тех пор, пока не произошла ничья или победа одного из побед

Конец игры

1. Платформа выводит пользователю результат игры. Сохраняет партию в турнирную сетку.

Режим работы 4 - результаты турнира

1. Выводится результаты турнира

**Интерфейс**

(тут нужны **\_рисунки\_** или скриншоты) Возможно обойтись “техническим описанием интерфейса”.

**Данные**

(Входные: могут быть описаны выше в режимах работы программы или в интерфейсе)

Параметры схемы

(Выходные: могут быть описаны выше в режимах работы программы или в интерфейсе)

Схема на экране

Файл картинки на жёстком диске

**Хранение данных**:

Настройки программы хранятся в .ini файле, лежащем в папке с программой.

(Возможны варианты:

БД для хранения данных

Данные хранятся на сервере …

Введённые пользователем данные хранятся в текстовых файлах в указанной папке на жёстком диске)

**Технические особенности**

ОС: Windows.

Тип программы: однопользовательское desktop-приложение.

(Запускается, как программа на компьютере)

Работает оффлайн.

Системные требования (ограничения): нужна такая-то библиотека. лицензионная версия Word.

Нужно 232Гб памяти, и 101 млрд лет, чтобы отсортировать мой массив

(Возможны варианты:

Web-страничка (запускается в браузере)

Многопользовательское клиент-серверное Web-приложение. Запускается на двух (нескольких) компьютерах одновременно. На одном - программа-сервер, на другом - программа-клиент. Требуется возможность подключения к локальной сети.

Онлайн-приложение. Для работы требуется доступ в инет.)