

Тестовое задание на программиста - Metagameplay

Цель:

Необходимо реализовать консольное приложение, которое представляет собой упрощенную модель экономики игры.

Требования:

Приложение состоит из 2 компонентов: Game Server и Game Client.

Описание имущества (items: корабли, снаряжение и прочее) можно сделать в наиболее простом виде: конфиг в ru, xml файле и пр. Информация по каждому предмету: name, price.

Все операции с имуществом и балансом аккаунта выполняются на server'е. Client может формировать только соответствующие запросы (желательно, с валидацией и кэшированием).

Game Server:

- Является master-data информации об аккаунтах: nickname, credits, items.
- Хранит все аккаунты в некотором хранилище (file, sqlite db, mysql, etc).
- Является master-data всех используемых в игре items. Они передаются на client при логине.
- Обрабатывает все запросы client'a на login, logout, покупку/продажу items.
- Процедуру логина клиента достаточно проводить только по nickname.

Заработок credits: после каждого login'a однократно на аккаунт начисляется произвольное количество credits в диапазоне значений, задаваемом в конфиге сервера.

Пользовательский сценарий:

Игрок запускает client. Client находится в state 'login' и предлагает игроку указать аккаунт для 'game session'. Выбор происходит по введенному игроком nickname. Client посылает запрос на server, который либо создает новый, либо выбирает существующий из storage и возвращает на клиент данные по аккаунту: nickname, items, credits. Client переходит в state 'game session'.

Игровая сессия предоставляет следующие возможности игроку:

- Просмотр баланса аккаунта: credits;
- Полный список всевозможного имущества (items);
- Просмотр списка имеющихся на аккаунте items;
- Игрок может купить любой item, который отсутствует на аккаунте и продать любой имеющийся у него item;

Игрок может закрыть игровую сессию и перейти в state login.

При очередном логине игрок увидит все купленные ранее items и credits.

Сетевое взаимодействие:

Любой удобный протокол, как прикладной, так и TCP/UDP сокет.

Язык Программирования:

Python (не ниже 2.7). Не рекомендуется использовать фреймворк Django, Flask. В остальном ограничений на библиотеки нет. Если это не стандартная библиотека Python, просьба приложить ее к исходникам. Приложение должно запускаться под Win.

Что оценивается:

Полнота и качество реализации, архитектура, качество и оформление кода.