

# Senior Unity Developer

## Тестовое задание / Multicast Games

### Задача

Создать прототип мобильной игры-головоломки, в которой игрок должен составлять слова из кластеров букв, являющихся последовательностью букв в том порядке, в котором они находятся в заданных словах.

Прототип должен включать следующие элементы:

#### 1. Главное меню:

- Кнопка "Play", которая запускает последний не пройденный уровень. Если пройдены все уровни то снова запускается первый;
- Счетчик, который отображает количество пройденных уровней.

#### 2. Экран игры:

Игровое поле, на котором столбцом расположены четыре незаполненных слова (пустые ячейки под буквы) длиной 6 букв каждое.

В нижней части экрана находится прокручиваемая по горизонтали панель с кластерами букв (по 2-4 буквы в каждом), из которых игроку предстоит составлять загаданные в уровне слова, выкладывая их на игровое поле. Необязательно кластер должен являться слогом слова. Примеры кластеров: (КЛ) (АС) (ТЕР).

Кластеры букв должны быть объединены рамкой для наглядности. Она остается видимой при выкладывании на поле, чтобы игрок мог убрать или переместить неправильно выставленный кластер.

При выставлении нового кластера на игровое поле должна происходить автоматическая валидация игрового поля на правильность выложенных игроком слов. Условием прохождения уровня считается тот факт, что игрок выложил все кластеры на поле и собрал из них все загаданные слова. Положение слов в столбце на поле значения не имеет.

#### 3. Экран победы:

- Отображение всех разгаданных слова столбцом в том же порядке, в каком их разгадал игрок
- Кнопка "Main Menu", которая переводит игрока в Главное меню
- Кнопка "Next Level", которая переводит игрока на следующий уровень

## Требования

- **Версия Unity:** 6000.x.x
- **Технологии.** Необходимо самостоятельно выбрать технологии подходящие под поставленную задачу. Разрешено использовать сторонние библиотеки которые ускорят разработку и упростят поддержку игры.
- **Активы.** Допускается использовать бесплатные активы из Unity Asset Store либо можно встроенные примитивы.
- **Финальная работа:** требуется предоставить ссылку на открытый репозиторий GitHub или любой его игры аналог, apk файл (android) и краткое описание архитектуры проекта.
- **Хранение конфигурации уровней.** Конфигурацию уровней необходимо хранить в json файлах. Структуру .json файла уровня необходимо спроектировать самостоятельно.
- **Сохранение игрового прогресса.** При перезапуске игры должен сохраняться прогресс игрока.

## Будет оцениваться

- **Расширяемость архитектуры,** а именно сложность доработки проекта под новые требования. Полный список оцениваемых требований:

- **Расширяемость системы загрузки уровней.** Допустим, потребуется заменить загрузку уровней из json-файла на асинхронную загрузку с сервера;
- **Расширяемость геймплея.** Допустим, потребуется сделать игровой режим со словами длиной 8 букв;
- **Заменяемость геймплея.** Допустим, потребуется заменить геймплей, но при этом оставить остальную логику (систему загрузки уровней, сохранения прогресса, интерфейс);

Какое количество кода (не связанного с поставленной задачей) потребуется переписать для решения данных задач? *Примечание:* задачи из списка выше реализовывать **не** нужно.

- **Обоснованность выбора сторонних библиотек.** Упрощают и ускоряют ли библиотеки разработку или вносят избыточную сложность для данного проекта.
- **Структура кода.** Баланс простоты и производительности. Количество аллокаций памяти. Разделение конфигурации игры, игровой логики и логики интерфейса. Уместное соблюдение принципов SOLID и KISS
- **Аккуратность выполнения работы.** Соблюдение единого Code Style, аккуратность именования и расположения активов.