Общие требования и рекомендации по верстке

**1. Стиль программирования**

* Табуляция кода через двойной пробел. Можно использовать плагины в котором tab превращается в двойной пробел (так например делает редактор atom по умолчанию). Требование связано с единостью стиля чтобы компиляция не падала на ошибках вложенности
* Обязательно комментирование крупных смысловых блоков в целях читаемости
* Верстка может быть невалидной по валидатору W3C
* Осмысленное наименование классов по [BEM методологии](https://ru.bem.info/method/naming-convention/) (классический стиль block\_\_element--modifier). Обязательно загляните [сюда](http://nicothin.github.io/idiomatic-pre-CSS/#bem)

**2. Адаптивность**

* Под адаптивной версткой подразумевается отзывчивая (англ. - “responsive”) верстка, при которой содержимое игры подстраивается под фактические размеры экрана.
* При размерах более 320px не должно возникать горизонтальной прокрутки игры (за исключением отдельных блоков верстки, где подобная реализация предусмотрена дизайн-макетами).
* Допускается использование desktop first методов в подходе
* Ключевыми “брейкпойнтами” принято считать - 1920(hd+), 1440 (hd), 1200px (wide), 992px (desktop), 768px (tablet), 568px(mobile), 414px(mobile-s). Применяются для адекватного, пропорционального изменение отступов и размеров шрифтов. Либо, при наличии макетов мобильной версии, соответствии таковым
* Допускается использование других брейкпойнтов при необходимости с произвольным интервалом @media выражений

**3. Python**

**-** Любой описанный класс единовременно представляет из себя и объект.

**-** Функция множественного наследования.

**-** Поддержка виртуальных функций.

**-** Возможность легко управлять именами скрывать их особыми метками.

**-** Возможность жизнью объекта и распределение памяти.

**-** Управление работы операторов как символьных, так и логических.

**-** Возможность имитировать поле

**-** Управление полями – как прямой, так и частичный доступ.

**-** Контроль над самыми распространенными операциями. От глубокого до итерации по объекту.

**-** Возможно создавать триггеры и классы.

**4. Изображения**

* Все изображения должны быть оптимизированы через сервис tinypng в целях уменьшения суммарного веса картинок. Стандартные gulp оптимизаторы не проявили достаточной эффективности
* При использовании inline png в css, картинка должна быть оптимизирована До перевода в base64
* Обязательно использование srcset параметров в случае если макеты позволяют экспортировать изображения в большем размере для 2x экранов с высокой плотностью пикселей src=’img/name.png srcset=’img/name@2x.png 2x
* Непрозрачные картинки необходимо экспортировать в .jpg в целях экономии размера файла

**5. Шрифты**

* Шрифты, доступные через сервис WORD 2018
* В случае, если шрифты не находятся в бесплатном публичном доступе, использовать любой другой шрифт на свое усмотрение
* Используются колбеки типа шрифтов sans-serif, serif, monospace и т.д.
* Шрифты необходимо подключать в форматах eot, svg, ttf, woff, woff2. Используйте <https://transfonter.org>
* При подключении шрифтов одного семейства, перепишите font-weight в @font-face правиле чтобы в css можно было управлять жирностью и стилем через font-weight: 300 вместо font-family: ‘SuperFont-Light’, sans-serif

**6. Базовый функционал**

- Англоязычных версий игры не требуется

- Форма обратной связи для сбора заявок от пользователей игры.