

Для презентации.

О движке.

Движок игры полностью реализован при помощи такого сверхсовременного средства разработки, как интерпретируемый язык программирования Питон третьей версии.

Сий инструмент был выбран по нескольким причинам:

- Высокоуровневость и автоматическое позволяет писать код достаточно абстрактным, не отвлекаясь на мелочи и полностью сосредоточившись на самом процессе разработки.
- Интерпретируемость позволяет запускать код на любой системе, для которой существует интерпретатор языка, что обеспечивает крайне высокую переносимость (Windows XP и новее, большая часть дистрибутивов Linux, MacOS – вот далеко не полный список ОС, поддерживающих язык Питон).
- Также внутри языка невозможно не соблюдать структуру кода, что однозначно повышает его читаемость.
- Ну и самое главное – требование заказчика, которое, как известно, свято.

Если говорить о коде движка, то он соответствует следующим требованиям:

- Крайне высокая производительность. К примеру, игровая карта генерируется всего за $O(n^2)$, где n – количество звёзд на ней. Так как среднее количество звёзд не превышает двух десятков, время генерации не ощущается человеком. Это же касается и всех остальных внутриигровых вычислений.
- Использует высокотехнологичные структуры и алгоритмы, некоторые из которых уникальны и заслуживают отдельного упоминания:
 - К примеру, в коде был искусно реализована так называемая СНМ (Система Непересекающихся Множеств), которая используется для пресловутой быстрой генерации. Она позволяет узнать, объединены ли любые два элемента, и объединить любые два объекта. Кроме того, для максимальной производительности она структура была улучшена через эвристику по размеру.
 - Стоит также упомянуть, что основная карта хранится в максимально приспособленной для этого структуре данных – в графе. Он гарантированно является деревом, что обеспечивает больший внутриигровой накал: так как между двумя системами всегда существует только маршрут, то за каждую систему будут идти ожесточённые бои.
 - Для одной генерации карты используется авторская модификация алгоритма Краскала, которая обеспечивает построение дерева из графа путём перемешивания всех доступных рёбер и последовательным их добавлением при условии, что две вершины графа ещё не соединены, что проверяется через вышепоименованную СНМ.
 - Для грамотной гарантии, что атака корабля не будет производиться, необходимо было научиться считать расстояние на гексагонах. Для этого был написан алгоритм BFS (поиск в ширину) на графе, который позволяет получить множество доступных для перемещения в них вершин графа.
- Так называемая «чистота» кода или его понятность рядовому программисту и его избавленность от громоздких или неоднозначных конструкций, по оценке написанного профессионалами ПО pylint3, везде превышает 80% (всего 10 баллов, в кодах движка ≥ 8).
- Универсальность, которая обеспечена предусмотрительностью автора движка, кто добавил в код исключений с подробным описанием каждой ошибки, которую могут допустить пользователи движка.
- Отсутствие громоздкой и постоянно обновляющейся документации (для разработчиков!), которая в ходе работы постепенно была вытеснена так называемой доктриной «самодокументирующегося кода», что предусматривает адекватное количество комментариев, содержательные имена переменных и отсутствие сложных нюансов в использовании движка.
 - Отдельно стоит сказать, что изначально система документации основывалась на html-документах, которые несли крайне подробное описание кода. Но позже практика совместной работы показала, что в них нет необходимости, ибо качество кода действительно на высоте, что позволило сократить затрачиваемое на написание документации время.

Об интерфейсе.

Продукт нашей команды имеет несколько замечательных достоинств, о которых можно долго рассказывать. Одним из них является наличие целых двух систем UI:

- GUI позволяет даже тем, кто с современными ЭВМ на «вы», играть в «SpaceEmpire» без каких-либо трудностей. Он обладает крайним быстродействием, оформлен в ретро-стиле и интуитивно понятен.
- Но для тех, кто не может позволить себе такую роскошь, как запуск нетребовательного интерфейса на своей машине, была создана и аналогия ГУЯ: терминальный интерфейс. Он ещё больше погрузит игроков в детство, в далёкие восьмидесятые; он развивает фантазию и запускается на абсолютной любой машине, но в плане функциональности ничем не уступает графическому. Стоит ещё добавить, что он написан в форме интуитивно удобного класса с крайне удобной системой переадресации между различными игровыми меню.