



Technical Design Document

1§ Тупое ядро:

Stupid_Core

Суть тупого ядра...Это программа, которая запускается, выводит лого, и позволяет сделать тебе целый один выбор. "Следующий день". Нажимаешь кнопку, наступает следующий день, все просто. На экране отображается, сколько дней прошло.

2§ Предметы:

Items

Предметы в игре бывают самые разные. Каждый нужно будет создать и описать.

Что общего у всех предметов - это id, некое значение которое они изменяют, это может быть здоровье (аптечка), голод (еда), оружие(урон). Вот пока и остановимся на этих трех составляющих - это id, name, modification. Естественно класс должен иметь конструктор. Далее, все предметы должны быть расфасованы по динамическим массивам. То есть массив - `vector <int,item> medical;` и так далее. (eat, drink, buff, weapon) Так же, нужно сделать функцию, которая будет выводить все массивы со всеми items. `void DebugItems();`

В будущем классы будут модифицироваться. Но на этом этапе нам важно сделать рабочие прототипы.

- На данном этапе делается простой прототип, это несколько массивов, с предметами взятыми от балды, список которых будет выводить `DebugItems();`

3§ Графика_items:

Да-да, это больше из раздела "Дизайн-библия", но техническая часть тут не маловажна. Все рисунки делаются основываясь на английской локализации. Так как в этой локализации у Аски есть хоть какие то символы. Мы не можем использовать цвета, и сложные уловки, так как консоль на разных компьютерах работает по разному.

Стандартный английский работает на всех одинаково. Тут мы сталкиваемся с проблемой из раздела "Story-bible". Как писать на русском. Либо мы замудрим с локализациями, но скорее всего сделаем свой корректор текста, и Русский будет работать через английскую локаль.

Рисунки устроены в виде двумерных массивов забитыми чаровскими переменными. К примеру `char s = 177;` - это серый квадратик. `char print[3][3]{{s,s,s},{0,s,0},{s,s,s}};` - так у нас получается английская буква "I" или римская 1.

К каждому items полагается свой набор карточек с рисунком. Это полноценная картинка и ее уменьшенная копия для инвентаря.

- После создания первого набора Items, необходимо подготовить рисунки.

4§ Добро и зло:

Это реализуется очень просто. Берется одна переменная, и в зависимости от действий ей либо прибавляется значение, либо вычитается. Когда требуется проверка, мы просто смотрим, достаточно эта переменная велика либо низка.

К примеру мы делаем доброе дела, нам прибавилось +1. А в итоге Если (Доброта >= 10) { хорошая заставка } иначе { ты сдох }

- Создать прототип, где можно будет сделать выбор между добром и злом. Скажем выбор будешь делать 5 итераций. По завершению программа будет выдавать, какой ты персонаж - добрый, злой или нейтральный.

5§ Метод выбор:

Выбор в игре осуществляется стрелками, это не сложно, и я активно использую эту систему, в корне которой лежит обычное `_getch()`; Игрок будет выбирать одно из действий, где будут запускаться разного рода функции.

- Создать прототип с несколькими вариантами выбора, где результатом будет выводиться сообщение о сделанном выборе.

6§ Приключение:

Рандомно выбирается одно из приключений. Затем выводится текст в рамочке, и в зависимости от приключения - это может быть влияние на персонажа, либо выбор какого либо действия влекущего за собой последствия. В общем, в зависимости от story-bible составляются приключения, добавляются в массив приключений, а затем рандомайзер выбирает нужное.

В идеале разделить приключения на подтипы, разбить каждое на сектора, и позволить их частям перемешиваться. Это грамотно подобранные тексты, и сбалансированные влияния*.

- Написать прототип, который будет генерировать истории. Затем выбирать один из массивов, и историю из этого массива. И выводить ее на экран. Если в истории есть

выбор, то предоставить его игроку, затем вывести на экран результат выбора. Это прототип, по этому истории самые простые.

*Влияние - это, последствия приключения, которые вносят изменения в статус персонажа. Уменьшают здоровье, или увеличивают деньги.

7§ Инвентарь :

Это таблица, в которой размещаются мини-карточки предметов. Выбор осуществляется стрелочками, затем после нажатия определенной кнопки(Enter) открывается подменю, что именно сделать с этим предметом. (Экипировать, использовать, продать.)

- Прототип с таблицей, и элементарными значками, но с возможностью выбирать вещь в инвентаре, использовать ее и продавать.

8§ Аферист :

С технической точки зрения, это рандомайзер, который выдает один из предметов. Кроме оружия.

- Прототип: Это генератор кейсов в кс-го. Взять несколько массивов с item's и вывести один из предметов на экран

9§ Статус :

Монитор отображающий характеристики персонажа и его экипировку. А также при получении уровня, здесь можно прокачать навык.

- Прототип выводит абстрактные характеристики персонажа, и дает возможность прокачать одну из них.

10§ Карта :

Функция которая при определенных условиях позволит отправится на новую локацию. Можно сделать не карту, а например Эвакуация, или еще что. Но суть в том, чтобы по достижению определенного прогресса, игрок попадал в новую сюжетную арку и локацию.

- Написать прототип, который выводит описание текущей локации. Здесь будет 2 действия: 1) Спасать людей, 2) переезд.
- Когда ты спасешь определенное количество людей, тебе будет доступен переезд, не раньше. Затем описание новой локации и так далее.

11§ Лагерь (Beta) :

Лагерь это таблица, в которой со временем будут появляться специальные item-ы улучшающие статусы персонажа. За выполнение определенных квестов, там будут открываться эти улучшения. Например спас девушку, в лагере появилась медсестра, которая увеличивает эффективность аптек.

- написать прототип с простой таблицей, куда при нажатии определенных клавиш будут добавляться объекты, увеличивающие статус. Простой интерфейс статуса вывести внизу.

12§ Персонаж:

Один из самых важных аспектов. Это единичный экземпляр класса- структуры, который имеет множество статусов: Здоровье, Голод, жажда, эвакуированные люди, опыт, экипировку, злобу и доброту, деньги, и всякие прочие штуки.

- Создать прототип персонажа, со всеми характеристиками и вывести на экран.

13§ Карьера:

Это некий объект, хранящий в себе информацию о сделанных выборах и продвижении сюжета в целом.

Это пока абстрактная штукенция, непонятно как это будет работать, пока story-bible не написана.