

Система продажи книг Soul of Fallen Worlds

К сожалению, оригинальная продажа книг Fallout 3 не работает со всем разнообразием книг Fallout New Vegas... И тут на помощь прихожу я, Trust.

Старая система продажи книг имела свои недостатки и сильно отличалась от оригинала, тем не менее, идеи старой системы легли в основу нового фикса, в частности, способность «Книголюб» позволяет находить ценные книги и продавать их по большей цене.

Новая система конфигурируется через файл настроек, и по умолчанию, предлагает формулы цены и процент нахождения ценных книг основанные на навыках и параметре Удача.

В этом документе собраны все нюансы скриптов системы, и опубликован код с подробными комментариями. Фикс вошёл в состав SFW – Приятной игры!!!

Авторы системы: Trust (я) и мой друг Вис691, он дал множество ценных советов, без которых реализовать эту систему бы не получилось.

```

scn aaaSFwbtcBookPerkScript ; Имя скрипта, квестовый скрипт способности "Книголюб"

int BookPerk ;Хранилище переменной BookPerk, для функции. Если больше 99, то способность загружена
int DoOnceBookPerk ;Хранилище переменной DoOnceBookPerk, нужно для одноразового срабатывания проверки и сообщения

int LuckAffects ; Переменная влияния удачи на способность нахождения ценных книг. Настраивается в конфиге
; Цикл игрового режима
Begin GameMode
    ; Если игра была загружена, то читаем значение LuckAffects из конфига (0 или 1)
    If GetGameLoaded || GetGameRestarted
        LuckAffects = GetINIFloat "BooksToCapsOptions:bLuckAffects" "BooksToCaps.ini"
    EndIf
    ; Запускаем функцию способности с аргументом LuckAffects
    Call aaaSFwbtcBookPerkScrFunc LuckAffects
End

; Цикл режима меню. 1009 - диалог с персонажем.
Begin MenuMode 1009

    ; Если игра была загружена, то читаем значение LuckAffects из конфига (0 или 1)
    If GetGameLoaded || GetGameRestarted
        LuckAffects = GetINIFloat "BooksToCapsOptions:bLuckAffects" "BooksToCaps.ini"
    EndIf

    ; Запускаем функцию способности с аргументом LuckAffects
    Call aaaSFwbtcBookPerkScrFunc LuckAffects
End

```

#####

scn aaaSFwbtcBookPerkScrFunc ; **Имя скрипта-функции. Функция способности.**
; **Я отделил этот скрипт в функцию, чтобы избежать дублирования кода в основном скрипте**

; Объявление переменных. int это целое число, float это дробное число, ref это идентификатор

int LuckAffects

int RecycleArrSize

int LuckValue

int tmpEachBookCount
ref rBookSFwbtcPERK

float Treshold1000
float Treshold725
float Treshold450

int CountForExchange
float BookRandom

int SFWBookGeneric1000Count
int SFWBookGeneric725Count
int SFWBookGeneric450Count
int PreWarBookCount

int CountDivided
int DivOffset
float RandomMin
float RandomMax
int i
int RandomBooksCount

; Начало функции. От аргумента LuckAffects зависит, влияет ли удача на нахождение ценных книг. Если 1, то ДА
Begin **Function** { LuckAffects }

; Главный блок. Срабатывает, если у игрока есть книги на обмен
If Player.GetItemCount aaaSFWBooksFormList

RecycleArrSize = ListGetCount aaaSFwbtcBooksFormList ; **Размер массива Отбраковки**

; Считывание удачи. Если удача больше десяти, ограничиваем её десяткой
LuckValue = Player.GetAV Luck
If LuckValue > 10
LuckValue = 10
EndIf


```

; Если у игрока есть хоть одна ценная книга
If SFWBookGeneric1000Count || SFWBookGeneric725Count || SFWBookGeneric450Count
    ShowMessage aaaSFWbtcBooksVALUABLEmsg ; Показать сообщение "Вы нашли ценные книги"

    If SFWBookGeneric1000Count ; Обменять счётчик на очень ценные книги
        Player.AddItem SFWBookGeneric1000 SFWBookGeneric1000Count 1
        ;SFWBookGeneric1000Count = 0 ; ТИП "ФУНКЦИЯ" НЕ СОХРАНЯЕТ ПЕРЕМЕННЫЕ
    EndIf

    If SFWBookGeneric725Count ; Обменять счётчик на ценные книги
        Player.AddItem SFWBookGeneric725 SFWBookGeneric725Count 1
        ;SFWBookGeneric725Count = 0 ; ТИП "ФУНКЦИЯ" НЕ СОХРАНЯЕТ ПЕРЕМЕННЫЕ
    EndIf

    If SFWBookGeneric450Count ; Обменять счётчик на довольно ценные книги
        Player.AddItem SFWBookGeneric450 SFWBookGeneric450Count 1
        ;SFWBookGeneric450Count = 0 ; ТИП "ФУНКЦИЯ" НЕ СОХРАНЯЕТ ПЕРЕМЕННЫЕ
    EndIf

EndIf

If PreWarBookCount ; Если счётчик обычных книг больше нуля

    ; Здесь мы делим число книг отбраковки на количество стопок, и ниже получаем остаток от деления
    CountDivided = PreWarBookCount / RecycleArrSize ; Разделённое количество книг на стопки
    DivOffset = PreWarBookCount % RecycleArrSize ; Остаток от деления
    ;i = 0 ; ТИП "ФУНКЦИЯ" НЕ СОХРАНЯЕТ ПЕРЕМЕННЫЕ

    If DivOffset ; Если есть остаток от деления, то прибавляем его в случайную стопку книг
        ; (в диапазоне от номера 0, до границы массива)
        Player.AddItem (ListGetNthForm aaaSFWbtcBooksFormList (GetRandomInRange 0 RecycleArrSize)) DivOffset 1
    EndIf

    If CountDivided ; если количество книг в стопке больше нуля

        ; Вычисляем минимум и максимум случайного числа
        ; Минимум: это 1 разделить на квадратный корень из количества, умножить на количество и отнять 1
        ; Максимум: это 2 разделить на квадратный корень из количества, умножить на количество и отнять 1

        RandomMin = 1 / (fsqrt CountDivided) * CountDivided - 1
        RandomMax = 2 / (fsqrt CountDivided) * CountDivided - 1

        ; Если количество стопок книг в массиве нечётное, то с помощью остатка
        ; от деления на два мы это определяем, и записываем в первую ячейку обычное количество
        ; и счётчик массива и цикла ставим на вторую ячейку.
        ; Это позволяет избежать ошибки выхода за границы массива, так-как следующий цикл обрабатывает ячейки попарно

        If RecycleArrSize % 2
            i = 1
            Player.AddItem (ListGetNthForm aaaSFWbtcBooksFormList 0) CountDivided 1
        EndIf
    EndIf
EndIf

```

```
;RandomBooksCount = 0 ; ТИП "ФУНКЦИЯ" НЕ СОХРАНЯЕТ ПЕРЕМЕННЫЕ
```

```
While i < RecycleArrSize ; Цикл. Пока счётчик меньше количества стопок в массиве
```

```
    If CountDivided != 1 ; Если мы получили квадратный корень из единицы, то пропускаем этот блок  
        RandomBooksCount = Rand RandomMin, RandomMax ; Вычисление случайного числа.
```

```
    EndIf
```

```
    ; Псевдослучайный алгоритм. В стопках должно лежать разное случайное число книг
```

```
    ; Этого мы добиваемся тем, что временное случайное число отнимаем у первого парного элемента
```

```
    ; И прибавляем ко второму парному элементу. Псевдослучайность.
```

```
    Player.AddItem (ListGetNthForm aaaSFwbtcBooksFormList i) (CountDivided - RandomBooksCount) 1
```

```
    Player.AddItem (ListGetNthForm aaaSFwbtcBooksFormList (i + 1)) (CountDivided + RandomBooksCount) 1
```

```
    i += 2 ; Прибавляем счётчик цикла на следующую итерацию. Для обработки парных элементов.
```

```
Loop
```

```
EndIf
```

```
;PreWarBookCount = 0 ; ТИП "ФУНКЦИЯ" НЕ СОХРАНЯЕТ ПЕРЕМЕННЫЕ
```

```
EndIf
```

```
End
```

```
#####
```

```
scn aaaSFwbtcBooksToCaps ; Название скрипта-функции. Эта функция продажи работает в диалогах Писца Йерлинг для продажи обычных книг.
```

```
; Объявление переменных
```

```
int HowMuchToSale
```

```
ref BookListSFwbtc
```

```
ref rBookSFwbtc
```

```
int Count
```

```
int BookPrice
```

```
int iRewardXP
```

```
Begin Function { HowMuchToSale } ; Функция принимает аргумент HowMuchToSale (количество для продажи)
```

```
    ; Побочный эффект такой реализации состоит в том, что если аргументом функции подать 2 миллиарда, то продадутся все книги, что я и использую.
```

```
    ; Читаем цену книги из файла настроек. Если цена меньше одного, то рассчитываем её согласно навыкам
```

```
    BookPrice = GetINIFloat "BooksToCapsOptions:iFixedPrice" "BooksToCaps.ini"
```

```
    If BookPrice < 1
```

```
        BookPrice = (Player.GetAV Science)/5+(Player.GetAV Repair)/5+(Player.GetAV Medicine)/5+(Player.GetAV Barter)/5+(Player.GetAV Survival)/5
```

```
    EndIf
```

```

; Читаем количество очков опыта из файла. Если оно меньше одного, то устанавливаем его в значение 10
iRewardXP = GetINIFloat "BooksToCapsOptions:iRewardXP" "BooksToCaps.ini"
If iRewardXP < 1
    iRewardXP = 10
EndIf

; Выбираем какие книги продать, отбраковку или обычные. Если способность 100, то отбраковку, если нет, то обычные.
If aaaSFwbtcBookPerkQuest.BookPerk > 99
    BookListSFwbtc = aaaSFwbtcBooksFormList
Else
    BookListSFwbtc = aaaSFWBooksFormList
EndIf

; Читаем ID каждой книги из списка в цикле
ForEach rBookSFwbtc <- BookListSFwbtc
    Count = Player.GetItemCount rBookSFwbtc ; Количество книг в стопке

    ; Если количество книг 0, то прерываем текущую итерацию цикла и начинаем следующую
    If Count < 1
        Continue
    EndIf

    ; Если количество книг в стопке больше чем осталось продать, то приравниваем количество к продаже
    If Count > HowMuchToSale
        Count = HowMuchToSale
    EndIf

    Player.RemoveItem rBookSFwbtc Count ; Удаляем количество книг текущей итерации
    Player.AddItem f (Count * BookPrice) ; Прибавляем деньги (Цена умноженная на количество)
    RewardXP (Count * iRewardXP) ; Прибавляем очки опыта. (Опыт за книгу умноженный на количество)

    aaaSFwbtcBookPerkQuest.BookPerk += Count ; Делаем переменную способности равной количеству проданных книг

    HowMuchToSale -= Count ; Отнимаем от количества для продажи число проданных книг в текущей итерации
    ; Если количество для продажи равно нулю, то прерываем цикл
    If HowMuchToSale < 1
        Break
    EndIf
Loop

; Здесь считаем состояние способности, и когда она равна 100, то запускаем её
If aaaSFwbtcBookPerkQuest.DoOnceBookPerk == 0 && aaaSFwbtcBookPerkQuest.BookPerk > 19
    aaaSFwbtcBookPerkQuest.DoOnceBookPerk = 1
    ShowMessage aaaSFWBookPerkSpoiler ; Показать сообщение-спойлер
EndIf
If aaaSFwbtcBookPerkQuest.DoOnceBookPerk == 1 && aaaSFwbtcBookPerkQuest.BookPerk > 99
    aaaSFwbtcBookPerkQuest.DoOnceBookPerk = 2
    Player.AddPerk aaaSFWBookPerk ; Добавить способность
    StartQuest aaaSFwbtcBookPerkQuest ; Запустить квест способности
    ShowMessage aaaSFWBookPerkMsg ; Показать сообщение "Вы заработали способность"
EndIf
End

```

#####

scn aaaSFwbtcBooksToCapsVALUABLE ; Эта функция работает в диалоге Писца Йерлинг для продажи всех Ценных Книг

```
int Count
float ValuablePriceMult

Begin Function {}

    ; Читаем множитель цены для ценных книг из файла
    ; Параметр может быть дробным
    ValuablePriceMult = GetINIFloat "BooksToCapsOptions:fValuablePriceMult" "BooksToCaps.ini"

    ; Если переменная в файле ноль или меньше, то сбрасываем её до 0.1
    If ValuablePriceMult <= 0
        ValuablePriceMult = 0.1
    EndIf

    ; Далее Ceil это округление до большего целого числа

    Count = Player.GetItemCount SFWBookGeneric450 ; Получаем количество довольно ценных книг

    If Count > 0
        Player.RemoveItem SFWBookGeneric450 Count ; Удаляем их
        Player.AddItem f (Ceil (Count * 45 * ValuablePriceMult)) ; Добавляем деньги за все довольно ценные книги
    EndIf

    Count = Player.GetItemCount SFWBookGeneric725 ; Получаем количество ценных книг

    If Count > 0
        Player.RemoveItem SFWBookGeneric725 Count ; Удаляем их
        Player.AddItem f (Ceil (Count * 72.5 * ValuablePriceMult)) ; Добавляем деньги за все ценные книги
    EndIf

    Count = Player.GetItemCount SFWBookGeneric1000 ; Получаем количество очень ценных книг

    If Count > 0
        Player.RemoveItem SFWBookGeneric1000 Count ; Удаляем их
        Player.AddItem f (Ceil (Count * 100 * ValuablePriceMult)) ; Добавляем деньги за все очень ценные книги
    EndIf
End
```


; Конфигурационный файл книг BooksToCaps.ini

[BooksToCapsOptions]

; Влияет ли удача на нахождение ценных книг

; Если 0 – то нет, если 1 – то да

bLuckAffects = 1

; Фиксированная цена в крышках (целое число)

; Если 0 – то цена будет рассчитываться согласно навыкам

iFixedPrice = 0

; Количество Очков Опыта за каждую проданную книгу

iRewardXP = 10

; Множитель цены ценных книг.

; Может принимать дробные значения

fValuablePriceMult = 10.0

Кроме-того, в диалогах Йерлинг предусмотрены все условия так, чтобы ценные книги нельзя было упустить.

В диалоге считывается состояние перка, и если он больше 99 то список обычных книг подменяется на список отбраковки, в то время как скрипт способности может срабатывать во время диалога, и подменить обычные книги на ценные и отбраковку