

SMART WRITE



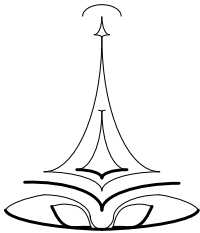
OPERATION MANUAL

Table of Contents

Programming directly benefits Creative Writing

Object oriented organization

Front panel user control



Programming directly benefits Creative Writing

Just as an artful painting represents the fruit of the subconscious intellect of the creator, taking form on the canvas through the rational part of his or her mind, a literary piece is also a combination of both intuition and rationale.

In some specific genre more than within the others, the attention of the author needs to split between the awareness of his or her subconscious impulses, and the logical consistency of the entire plot.

The program we deliver helps reducing the intellectual strain on the brain. This opens up wider space for the hidden insight of the subconscious part of mind, and promotes unveiling its limitless powers looming within the undiscovered depths, to emerge on the surface of awareness where these can be easily accessed by the artist. An art is a game of intertwining spontaneity and control.

Part of the last aspect is how should a writer decide how many people having what kinds of characters must enter a situation to achieve the desired outcome?

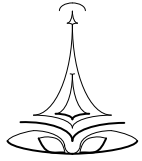
Smart Write can give you additional control on the events told by forecasting the possible result of various scenarios. That ensures the author can select the best characters and circumstances to convey his thoughts and feelings to us.



Sir Arthur Conan Doyle (1859 –1930), the inimitable visionary of the detective genre, the creator of legendary characters such as Sherlock Holmes and Prof. Moriarty. A crucial role in his memorable tales is played by logical perfection. One of his admirers was Albert Einstein himself.



One of the original Strand magazine covers presenting the Adventures of Sherlock Holmes, a piece of art where blending of reason and spontaneous intuition was displayed in most singular fashion.



Object oriented organization

Class participant

private Properties:

- ID
- partner ID
- character
- seat
- partner mating score



... N participants

Seat Randomizer

This program's first version
implementation is set for **four**
participants !

Class booth

private Properties:

- booth #
- participant ID
left seat
- participant ID
right seat

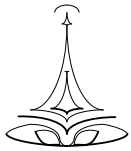


... roundup(N/2) booths

Class *booth* includes a *public* method
determining the "mating" between the pair of
participants in a booth

Based on their characters
and a rule-set





Front panel user control

The user sets the initial relationships between the four participants

The auto-correlation **blue** fields are for representing the characters of each participant

The final result = end – initial mating scores (the more difference, the more drama)

A monitor for displaying the properties of each participant's object inside the program, both initial values and after "mating" (for testing purposes)

If a “mating” between a pair of participants occurs, the updated mating score is displayed in **red** at the crossing index (takes into account the initial relationship)

SmartWrite

Enter Initial Relationships Between the Four Participants (A through D) :

	A	B	C	D
A	5 - tends to get personal fast	4 - strongly mated	4 - strongly mated	2 - strongly mated
B		2 - tends to get friendly	2 - mated	1 - semi-mated
C			0 - very introvert	0 - not mated
D				4 - tends to get personal

Click me after entering all input !

Initial mating score : 14

Drama Level : 3. Please select your input again !

Test Monitor

A properties list for each participant :

Participant A :
 cID = 1
 PartnerID = 0
 Character = 5
 Seat = 0
 PartnerMateScore = 0

Participant B :
 cID = 2
 PartnerID = 0
 Character = 2
 Seat = 0
 PartnerMateScore = 0

Participant C :
 cID = 3
 PartnerID = 0
 Character = 0
 Seat = 0
 PartnerMateScore = 0

Participant D :
 cID = 4
 PartnerID = 0
 Character = 4

