PunyInform

An Inform library for writing small and fast text adventures.

# Properties

A property can be used to store a 16-bit value, or an array of values (up to 32 values in z5, but only 4 values in z3).

If a property is declared as additive, the values for an object are concatenated with the values of its class, if any, and put into an array.

A property can either be common or individual. Common properties are a little faster to access and use a little less memory than individual properties. A game can use a maximum of 62 common properties. The value of a common property can always be read, but it can only be written if it has been included in the object declaration. If you don’t include it, there is no memory allocated to store a value. If you read the value of such a property, you just get the default value (typically 0).

A common property is created by declaring it with

Property *propertyname*;

To access a property, you write object.*propertyname*, like this:

Dog.description = “The dog looks sleepy.”;

## Limitations for z3

If you want to compile a game to z3 format, this is what you need to keep in mind:

* A game can use no more than 32 attributes and 31 common properties
* When using message passing (like “MyBox.AddWeight(5)” ), no more than one argument may be passed. In regular Inform, message passing doesn’t work at all in z3.
* Dynamic object allocation can not be used.

These are the properties defined by the library:

<TBD>