Interactive Fiction Document (IFD) Format Specification and Information

V4

Created by: Dr. Ray

Edited by: William Reed and Stephen Killough

New for Version 3

In this version of the IFD file, there have been a few additions allowing for more interactive gameplay. This version will allow for the random placement of areas based on new area tags, which currently include <basement>, <basementR>. <ground>, <groundR>, <upper>, <upperR> and <stairs>. These area types are only able to connect to other areas with a matching area type, or the <stairs> type to link floors. The types that end with ‘R’ are used for the areas meant to be randomly generated when the player moves, rather than being preset at the start of the game. Also included in this edition are new item types. These include <omen> and <interact> items. The <omen> item type is used for items that are to be collected by the player in order to exit the house, winning the game. The <interact> item type is used to replace the <use> and <consume> items in the previous version. <interact> items can affect one of the player’s attributes, either positively or negatively. A new <snag> function has also been implemented so that the player can pick up and use the item if they choose. Inside of rooms there are now two kinds of events that can happen, dubbed room effects and room roll effects. Room effects are unavoidable and will happen upon entry, room roll effects are dependent upon your stats in a specific trait and can have three different outcomes.

**Supported Tags**

\*\*New Area Tags and Their Corresponding Tags\*\*

* <basement>: defines a new basement area
* <ground>: defines a new ground level area
* <upper>: defines a new upstairs area
* <stairs>: defines stairs, connecting different floors
* <rmDmg>: defines a new automatic room damage event
* <rmEft>: defines a new automatic room effect event
* <act>: determines the number of times that a <rmDmg> event or <rmEft> event can affect the player when the room in question is entered
* <rmMess>: similar to the description of an area, this message is printed when a <rmDmg> event occurs in a room, describing what happened to the player
* <Dmg>: contains multiple clauses: 1) integer value of either 1(for positive effect/gaining stats), or 0(for negative effect/damage), 2) a/m/s/k – one of these characters is picked to determine which stat is effected, and 3) an integer determining the amount of damage done.
* <#d>: the randomly generated number corresponding to the randomly simulated “dice roll.” this value will determine how the player is affected by the <rmEft>
* <hRoll>: rollAmt,good-1 or Bad-0,a/m/s/k,amt – contains multiple clauses: 1) rollAmt – the minimum dice roll needed to choose which Roll is used against the player, 2) integer value of 1 or 0, determining positive or negative effect on stats, 3) a/m/s/k – one of these characters is chosen to determine which stat is effected by the specific roll, 4)amt – the amount of damage done or stats given to the player
* <mRoll>: same as <hRoll>
* <lRoll>: same as <hRoll>
* <hMsg>: message printed after the roll type is determined based on the roll amount
* <mMsg>: same as <hMsg>
* <lMsg>: same as <hMsg>

\*\*New Item Tags\*\*

* <interact>: designed to replace consume/use items
* <effect>: since it is meant to replace consume/use items, this tag determines which attribute it effects, and whether it does damage or gives points to said attribute
* <omen>: these items are basically the goal of the game. When 3 of these items are picked up by the user, they mold into an item that allows the user to win the game by exiting the house

The following is a list of valid tags in this version of the format:

• game: top-level tag, all other tags and information for the game must be nested between the *< game >* and *< /game >* labels

• \*ptype: the ptype tag indicates the creation of a new player; the text inside the tag should list the player type; this is extensible but the two provided types are “basic” and “hpsp”

• area: the area tag indicates the creation of a new area in the game; information about the area is nested within

• desc: this is the “description” tag which contains the textual description (of an area when nested inside an area tag)

• feat: this is the “features” tag that describes an area; only two features are captured: instadeath and goal (additional information below)

• linkset: this tag represents the in-game connections from one area to four other areas (oriented up, down, left and right)

• links: this is the container tag for all the game’s linksets

• item: the item tag indicates the creation of a new in-game item; any number of item tags can be present; information about the item is nested within; each item must be specified as a specific type (by a single, nested third-level tag, see next tag)

• basic, use, consume: these are the three types of items

• name: this is the tag that identifies the “name” of the item in which the name tag is nested

• desc: this is the “description” tag which contains the textual description (of an item when nested inside an area tag)

• star: the data in this tag represents the “*st*art *ar*ea” of the number

• actmess: the message displayed when an item is “used” or “consumed”

• actar: the number of the area where the item can be activated (0 means any area)

• rule: for “use” items, there are any number of nested rule tags. Each tag contains a triples; affected area number, direction, and destination area number. So a rule that contains 1*, u,* 2 will update the up link in area 1 to point to area 2.

• effect: for “consume” items, there are any number of effect tags. Each tag contains a comma separated integer couplet. It is up to the specific Player object to interpret the included couplet.

Required Structure

Tags in the IFD format exist in one of three hierarchical “levels”, listed below as top-level through level 3.

Top-Level Tag: game

All IFD files begin with the *< game >*, and end with the *< /game >* label (game is the top level tag). Components inside this tag mean various things, but are all part of the defined game. Specifically, the second-level tags inside the game tag include: *< ptype >* . . . *< ptype >*, *< area >* . . . *< /area >*, *< links >* . . . *< /links >*, and *< item >* . . . *< item >*.

\*NEW\* Second-Level Tag: ptype

There should be only one ptype tag. The text inside the tag should correspond to a specific instance of a Player type. This is extensible: derive a new class to implement the Player class, and update the code to respond to that specific text. The three player types currently are the BasicPlayer class, the HPSPPlayer class, and the BasicCharacter class which will be created with the text, “basic”, “hpsp”, and “BasicB” respectively.

Second-Level Tag: area

There can be any number of area tags. Internally, areas will be assigned a number from 1 *. . . n* where *n* is the total number of areas. The first area tag will be assigned area 1, the second 2, etc. The player will always begin the game in area 1.

Each area tag must contain exactly two third level tags (both are required, both may appear only once per area tag): *< desc >* (description) and *< feats >* (features). These are the only tags that can appear nested in an area tag.

\*NEW\* area Third-Level Tag: basement, ground, upper, stairs

Nested in the <area> tag, the area type must be determined using either <basement>, <ground>, <upper>, or <stairs>. This will determine which rooms can connect to which. If the area type is followed by an ‘R’, this area is one that will be randomly generated when the player moves.

area Fourth-Level Tag: desc

Between the *< desc >* and *< /desc >* labels a string of characters representing the area description must appear. This string may take up more than a single line of text but not contain a line break.

area Fourth-Level Tag: feat

Between the *< feats >* and *< /feats >* are two comma-separated boolean values representing the insta-death and goal area features respectively. (1 for true, 0 for false, no spaces may appear in the comma separated list.

\*NEW\* area Fourth-Level Tag: rmDmg

Inside the rmDmg tag, more tags are placed to determine the type of damage, and how much is inflicted against the player.

\*NEW\* area Fifth-Level Tag: act

This integer value will determine the number of times the player will be affected by the room damage when entering the room

\*NEW\* area Fifth-Level Tag: Dmg

This value will determine the type (positive or negative) damage the player will take, the attribute it will affect, and the amount of damage dealt.

\*NEW\* area Fifth-Level Tag: rmMess

This is the message that will print when the player is affected by the damage.

\*NEW \* area Fourth-Level Tag: rmEft

Room Effects are a group of possible effects upon the player determined by a random, simulated dice roll. The different effects are specified by a minimum dice roll, and each can affect the player differently.

\*NEW\* area Fifth-Level Tag: rmMess

For the Room Effects, it gives a broad statement as to what event has taken place to affect the player.

\*NEW\* area Fifth-Level Tag: act

Same as the <rmDmg>, this integer value will determine the number of times the player will be affected by the room damage when entering the room.

\*NEW\* area Fifth-Level Tag: #d

The checks the character placed within this tag, a/m/s/k, checking its value to go against the simulated dice roll when a Room Effect takes place

\*NEW\* area Fifth-Level Tag: hRoll

This tag contains clauses determining the random dice roll amount, which attribute is affected, whether it is positively or negatively affected, and the amount by which it is affected.

\*NEW\* area Fifth-Level Tag: hMsg

This message is printed if the hRoll is enacted

\*NEW\* area Fifth-Level Tag: mRoll

This tag contains clauses determining the random dice roll amount, which attribute is affected, whether it is positively or negatively affected, and the amount by which it is affected.

\*NEW\* area Fifth-Level Tag: mMsg

This message is printed if the mRoll is enacted

\*NEW\*area Fifth-Level Tag: lRoll

This tag contains clauses determining the random dice roll amount, which attribute is affected, whether it is positively or negatively affected, and the amount by which it is affected.

\*NEW\* area Fifth-Level Tag: lMsg

This message is printed if the lRoll is enacted

Second-Level Tag: links

Exactly one *links* tag must appear in each IFD file. Only one kind of third-level tag can appear inside the links tag, the *linkset* tag. There should be precisely *n linkset* tags where *n* is the number of total *area* tags.

IMPORTANT: the order of the *linkset* tags nested in the *link* tag is critical. The first linkset represents the in-game connections emanating from the first area; the second from the second, so forth and so on.

links Third-Level Tag: linkset

Each *linkset* tag contains a single set of 4 comma-separated values. The first linkset represents the links (up, down, left, and right, respectively) for area 1. The second linkset for area 2, etc. as before.

Second-Level Tag: item

There can be any number of item tags. They *do not* have to appear in the file immediately after the area that they are starting in (though this was the style used in “DragonCrawl2.ifd”). It is perfectly okay to choose to put all item tags together at the end of the file, for instance, it’s just that approach is less organized.

item Third-Level Tag: basic, use, consume, interact, omen

Each *item* tag must be immediately followed by a nested type tag; either, *basic*, *use*, *consume*, *interact*, or *omen* representing basic items, items that respond to the use command, and items that respond to the consume command.

Each of these type tags must contain the other tags related to the item. There are three required fourth-level tags (these three are required and may appear only once per item): *< name >*, *< desc >*, and *< star >* (start area). In addition, there may also appear, *< actmess >* (“activate message”), *< actar >* (“activate area”), *< rule >* (representing a rule for modifying the game map), and *< effect >* (representing an effect on the player).

item Fourth-Level Tag: name

Between the *< name >* and *< /name >* labels a string of characters representing the item name must appear. This string may take up more than a single line of text (again, no line breaks) but likely will only be a few words long.

item Fourth-Level Tag: desc

As before, between the *< desc >* and *< /desc >* labels a string of characters representing the item description must appear. This string may take up more than a single line of text but contain no line breaks.

item Fourth-Level Tag: star

Between the *< star >* and *< /star >* labels a single integer must appear. This value represents the “start area” of the item (aka, the number of the area the item begins the game in).

item Fourth-Level Tag: actmess

Only items of type *use,* *consume,* or *interact* should contain the *< actmess >* tag. Each item of these types of items should have only one such tag. Inside the tag is the text string that will appear when the item is used or consumed.

item Fourth-Level Tag: actar

Only items of type *use, consume,* or *interact* should contain the *< actar >* tag. Each item of these types of items should have only one such tag. Inside the tag is a single integer representing the number of the area where the item may be used or consumed. The value 0 represents that an item may be used or consumed in any area.

item Fourth-Level Tag: rule

Only items of type *use* or *interact* may contain the *< rule >* tag. Such items may contain any number of *< rule >* tags. Each *< rule >* tag contains a comma separated triplet representing a single change in a link between area nodes on the game map. For instance, the triplet 3*, d,* 6 should be understood to mean to change the “down” link for area 3 to now point to area 6. When an item is successfully used (remember, an item may only be used in the correct “activation area”), all of it’s rules are executed on the map.

item Fourth-Level Tag: effect

Only items of type *consume* may contain the *< effect >* tag. Such items may contain any number of *< effect >* tags. Each *< effect >* tag contains a comma separated couplet representing a change to the player (often to player stats). The interpretation of the specific values in the couplet is dependant upon the type of player being used for the game.

Additional Notes

One additional note, IFD format also allows for comments. Any line that begins with the % symbol is a comment line. These are particularly useful when trying to keep track of an area’s number (which are automatically assigned in the order that the area tags are listed from top to bottom in the file). Comments can be used for this and, of course, other reasons. Effectively, each comment takes up one line of text; there is no block comment structure supported except for successive comment lines. As expected, comments should be ignored by any game engine that processes IFD formatted files.

Blank lines in the IFD format are ignored.