**Version 4 Updates:**

* Visual Map and Moveable Player
  + Version 4 allows for the creation of a moveable player of any type.
  + This character moves through an ASCII environment of a three vector
* True consumable items.
  + Moved from two item separated list to three item separated vector.
* Combat System (hack and slash)
  + Combat has now been enabled so that you can
    - Equip items.
    - Attack Enemies.
    - Use potions in and out of combat.

All these changes can be widely implemented within the design parameters explained below.

**Updated Specifications for the IFD Format:**

The Interactive Fiction Document (IFD) format is a mark-up style language (think HTML) for specifying interactive fiction games in a simple plain-text file. With such languages, meta-data “tags” are used to represent the meaning of text in the file. Tags have the following format: < tagname > is the “open” label of the tag (where “tagname” is the name of the tag) and </tagname > is the “close” label. Tags always come in label pairs (open and close label pairs): the first without the ‘/’ opens the tag and the second with the ‘/’ before the tagname closes the tag. All text (strings, values, or other nested tags) inside the tag label pairs are part of that tag’s information. When a tag’s open label is nested inside another tag, then it’s close label must also be inside that same tag, and vice versa.

For IFD Version 2, representable interactive fiction games consist of any number of discrete game “areas” (rooms, zones, etc.) that are interconnected with one another via “links”. This structure represents the areas that a player can move around in, constituting the “map” or “world” of the game. Additionally, in-game interactive “items” are supported. Items have certain characteristics that are detailed by the language (see below), one of which is the area (by number) that they begin the game in.

For IFD Version 3, it has items that that can be consumed but may be used multiple times. The HPSP player also has a very basic hit point system that does not include the different damage mitigators such as attack and defense stats for both players and NPCs.

**Tag Specification:**

Tags will be fully explained and organized by the tag leveling system.

**First Level:**

<game></game>: All other tags and information for the game must be nested between the <game> and </game> labels.

**Second Level:**

<ptype></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”, “hpsp”, NEW\* “combat”.

<area></area>: the area tag indicates the creation of a new area in the game; information about the area is nested within.

<link></link>: this is the container tag for all the game’s linksets.

<item></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, consumable, use, and NEW\* equipment.

**Third Level:** Each of these tags will now be nested in their parent tags.

Area)

<desc></desc>: This tag when nested in the area includes the description of the area itself.

<feats></feats>: this is the “features” tag that describes an area; only two features are captured: instadeath and goal (additional information below).

<map></map> NEW\*: This new feature allows for the creation of ASCII maps. ‘#’ indicates a wall/ object that can not be moved onto. ‘\*’ indicates the player spawn spot per area. ‘.’ Indicates an empty space that can be moved onto. All other characters can be given to enemies or items in a particular area.

<enemy></enemy> NEW\*: This new feature allows for the creation of stationary enemy NPC. Each enemy will have struct attached that describes their stats. There are no limits to the number of enemies except the individual ASCII characters assigned as each needs a unique character.

Link)

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

Item)

<\*annotated\*>: Listed as either “consumable”, “use”, or “equipment” in the \*annotated\* marks. All further information is listed in-between opening and closing of these tags.

**Fourth Level:**

Enemy)

<name></name>: contains the name of each NPC.

<desc></desc>: contains the description of each NPC.

<stats></stats>: contains attack, defense, and hit points respectively.

<code></code>: this is the unique character code that must be ASCII minus the “#,\*,.” which are exclusive to the map vectors.

Item)

<name></name>: name of the item goes here.

<desc></desc>: this contains the description of the item. You can choose to describe the item effect or leave the description vague and require the user to deduce the effect.

<star></star>: the data in this tag represents the “start area” of the item.

<actmess></actmess>: the message displayed when an item is “used”, “consumed”, or “equipped”.

<actar></actar>: the number of the area where the item can be activated (0 means any area).

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

<stat></stat> NEW\*: This is exclusive to the equipment tag. This is organized into the doble integer that represents the stat changes an equipped item gives the player. It is organized into “+defense, + attack” respectively.

**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.