Game Data File (GDF) Format – Version 2.0

The following describes the general format of version 2.0 of a game data file. ==You may create higher versions with added features if you wish, but they should be backwards compatible with version 2.0.== (If you do this, it is recommended you use 2.1, 2.2, etc., as there are already plans for 3.0, etc. to be issued by the instructor.)

GDF 2.0   Environment Name   // Magic word, version number and environment name.
                                     // Comments are ignored, as are blank lines

PLACES nPlaces   // Place section keyword, followed by an integer number of places
// By default the first place listed is the starting point for this environment

nPlaces times
{
ID long_name_with_spaces   // ID is a unique integer in the range of a 32-bit signed int
                           // ID numbers 0 and 1 are reserved
                           // Normally all ID numbers are positive (see below)
                           // followed by the name of the Place (remainder of line except comments)
*description   // One or more lines of text describing the Place, starting with an *

PATHS nPaths   // Path section keyword, followed by an integer number of paths

nPaths times
{
ID source direction destination lockPattern
        // ID is a unique integer in the range of 32-bit signed ints
        // source and destination correspond to Place IDs
        // A negative destination indicates a locked door
        // Destination 1 exits the program
        // Destination 0 leads "nowhere", and implies the door is locked and must stay locked
                // (Provides for future expansion of the environment)
        // direction indicates the direction from source to destination, and must be one of:
                // N, S, E, W, U, D, NE, NW, SE, SW,
                // NNE, NNW, SSE, SSW, ENE, ESE, WNW, WSW
        // lockPattern is a 32-bit int indicating the pattern (combination) of lock on this path
                // lockPattern = 0 indicates there is no key that can change this lock status

LIGHTING nEntries // Lighting section keyword, followed by an integer number of entries

nEntries times:
{
ID placeID lightLevel
        // ID is a unique integer in the range of 32-bit signed ints
        // placeID corresponds to a previously defined place.
        // lightLevel is an integer on the following scale:
                // 0 = pitch dark
                // 50 = normal inside room lighting
                // 75 = bright sunlight
                // 100 = blinding, as in looking straight into a bright spotlight or arc welder
        // Default light level of all places if not otherwise specified is 50
        // Assign a light level to place 0 to change the overall default level

ARTIFACTS nArtifacts // Artifacts section keyword, followed by an integer number of artifacts

nArtifacts
times:
ID placeID value movability name
   // ID is a unique integer in the range of 32-bit signed ints
   // placeID corresponds to a Place, indicating its original location.
   // value can be used for scoring points, or eventually for bartering & trade (integer)
   // movability is 0 for immovable objects, non-zero for movable objects (integer)
   // name allows user to refer to this object, and may contain spaces but not tabs
*description // One or more lines of text describing the Artifact, starting with an *

KEYS nKeys // Key section keyword, followed by an integer number of keys

nKeys
times:
keyID placeID keyPattern masterCode value movability name
   // keyID is a unique integer in the range of 32-bit signed ints
   // placeID corresponds to a Place, indicating its original location.
   // keyPattern is a 32-bit int. This key operates any item with a matching lockPattern
     // Divide lockPattern by $10\wedge$masterCode, then check if lockPattern == keyPattern
     // Example: keyPattern 52 with masterCode 2 works all locks 5200 to 5299
     // Note that $10\wedge0 = 1$, so masterCode = 0 is an ordinary key fitting only 1 lock
   // value, movability, and name are as for general Artifacts
*description // One or more lines of text describing the Key, starting with an *

LIGHTS nLights // Light section keyword, followed by an integer number of lights

nLights
times:
lightID placeID lightLevel value movability name
   // lightID is a unique integer in the range of 32-bit signed ints
   // placeID corresponds to a Place, indicating its original location.
   // lightLevel is an integer for the amount of light this Light adds to a place
     // The sum of ambient plus carried lights cannot exceed 100
   // value, movability, and name are as for general Artifacts
*description // One or more lines of text describing the Light, starting with an *

// Higher versions of the GDF file format may contain additional sections
// such as properties of the items indicated above, tools, nutrition, or other kinds of artifacts.