

```
'\-----(oo)
|| SCREEN FILE METADATA DOCUMENTATION  (_)
||-----||
```

Metadata is placed at the END of a .screen file, after the visual game content.
The metadata section starts with a special separator line:

```
==== METADATA ===
```

After this line, you can write metadata commands:

1. ROOM CONNECTIONS (CONNECT)

Purpose:

Defines where the player goes when exiting the current room.

Format:

```
CONNECT <DIRECTION> <ROOM_NUMBER>
```

Directions:

- LEFT Exit from the left side of the screen
- RIGHT Exit from the right side of the screen
- UP Exit from the top of the screen
- DOWN Exit from the bottom of the screen

Room Numbers:

- 0, 1, 2, 3... (matches the number in the filename adv-world_XX.screen)

Examples:

```
CONNECT RIGHT 1      # Going right leads to room 1
CONNECT LEFT 0       # Going left leads to room 0
CONNECT UP 3         # Going up leads to room 3
CONNECT DOWN 4        # Going down leads to room 4
```

2. SPECIAL DOORS (DOOR, KEYS, SWITCH, TARGET)

Purpose:

Defines special doors (█) that require keys and/or switch combinations.

Format:

```
DOOR <X> <Y>           # Door position on screen
KEYS <key1> <key2> ...    # Required keys (lowercase letters)
SWITCH <X> <Y> <STATE>   # Required switch position and state
TARGET <ROOM> <X> <Y>     # Teleport destination (optional)
---                         # End of door definition
```

Parts Explained:

DOOR <X> <Y>

- X = horizontal position (0-79)
- Y = vertical position (0-24)
- This is where the special door (■) is located on the screen

KEYS <key1> <key2> ...

- List of lowercase letters (a, b, c, etc.)
- Player must collect these keys to open the door
- Keys are lowercase letters on the map
- Doors are the matching uppercase letters (A, B, C, etc.)

SWITCH <X> <Y> <STATE>

- X, Y = position of the switch (0 or 1 on the map)
- STATE = required state (0 = off, 1 = on)
- Player must set all switches to correct states
- Can have multiple SWITCH lines for one door

TARGET <ROOM> <X> <Y>

- ROOM = room number to teleport to
- X, Y = position to teleport to in that room
- This is optional - only use if door teleports player

- Three dashes mark the end of the door definition
- Required if you have more content after the door

Example (Door requiring keys c, o, w and specific switch states):

```
DOOR 31 18
KEYS c o w
SWITCH 31 2 0
SWITCH 31 6 0
SWITCH 31 10 1
SWITCH 31 14 1
---
```

3. PRESSURE BUTTONS (PBUTTON, CLEAR)

Purpose:

Defines '\$' pressure buttons that clear specific walls ONLY while you stand on them.

Format:

```
PBUTTON <X> <Y>
CLEAR <X1> <Y1>
CLEAR <X2> <Y2>
...           # one CLEAR line per wall to remove
---          # end of this pressure button definition
```

Parts Explained:

PBUTTON <X> <Y>

- Position of the '\$' tile on the map

CLEAR <X> <Y>

- A wall position to temporarily erase while someone stands on the button
- Add as many CLEAR lines as needed

Example (a button at 40,12 that clears two walls):

PBUTTON 40 12

CLEAR 10 5

CLEAR 12 5

4. DARK ZONES (DARK, DARKZONES, ZONE)

Purpose:

Defines rectangular areas that are dark and require a torch (!) to see.

There are two ways to define dark zones:

METHOD 1 - Single Dark Zone:

DARK <X1> <Y1> <X2> <Y2>

- X1, Y1 = top-left corner of the dark area
- X2, Y2 = bottom-right corner of the dark area

METHOD 2 - Multiple Dark Zones:

DARKZONES

ZONE <X1> <Y1> <X2> <Y2>

ZONE <X1> <Y1> <X2> <Y2>

...

- DARKZONES starts the multi-zone block
- Each ZONE defines one rectangular area
- You can have as many ZONE lines as you need
- --- ends the block

How Dark Zones Work:

- Areas inside dark zones show darkness characters
- Players need a torch (!) to see and navigate
- Torch creates light in a radius around the player:
 - Distance 0-2: Full visibility
 - Distance 3: Light shade
 - Distance 4: Medium shade
 - Distance 5: Heavy shade
 - Distance 6+: Complete darkness

Example:

DARKZONES

ZONE 14 5 78 15

ZONE 23 1 78 4

5. MESSAGE BOXES (LINE1, LINE2, LINE3)

Purpose:

Defines text to display in an info box on the screen.
The box must have a 'T' marker in its top-left corner.
Text is automatically centered within the box.

Format:

```
LINE1 <text>          # First line of text (top)
LINE2 <text>          # Second line of text (middle)
LINE3 <text>          # Third line of text (bottom)
```

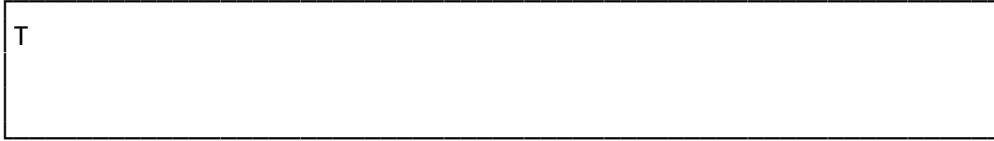
Parts Explained:

LINE1, LINE2, LINE3

- Each line contains the text to display
- Text is automatically centered within the box width
- If text is too long, it gets truncated
- If text is shorter than width, spaces are added on both sides
- Empty lines are allowed (just omit the LINE command)

Screen Setup:

Place a 'T' character in the top-left corner of where text should appear:



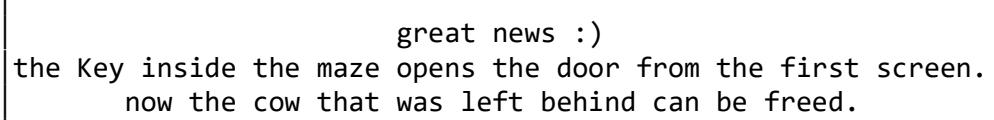
T

- The 'T' marks where the first line starts. The 'T' itself is removed from the screen when the game loads.
- The box width is auto-detected by scanning from the 'T' to the right until a `|` character is found.

Example (3-line message):

```
LINE1 great news :)
LINE2 the Key inside the maze opens the door from the first screen.
LINE3 now the cow that was left behind can be freed.
```

Resulting box on screen:



great news :)
the Key inside the maze opens the door from the first screen.
now the cow that was left behind can be freed.

6. COMMENTS

Purpose:

Add notes that will be ignored by the game.

Format:

```
# Your comment text here
```

Comments can be placed anywhere in the metadata section.
They help explain what the metadata does.

Example:

```
# Room 0 connections
CONNECT RIGHT 1
CONNECT LEFT 6

# Special door requiring keys c, o, w
DOOR 31 18
KEYS c o w
```

