

Turtle Tuesdays Challenge for July 2026

MYSTERY MANSION

STEP 6: Set up your objects.

Have you decided what objects are in each room? Not every room needs to have an object.

To see what object is in a room, the player will type `EXPLORE`. We'll get to that procedure shortly.

First let's determine what properties an object might have and what conditions might occur.

- Can the player take the object? (Some may be too large or heavy.)
- What do you tell the player after they take the object?
- Does the player need to take the object in order to complete part of the adventure?
- Has the room been explored before and the object taken?
- Is there any particular action that occurs when they visit the room or take the object?

Create a `SETUP_ITEMS` procedure that will store all this information for each room. It is similar in structure to the `SETUP_MAP` procedure, but deals specifically with the objects the player finds. The beginning of this procedure might look like this:

```
TO SETUP_ITEMS
; set up the items, what is REQUIRED, etc.
  PPROPS "ENTRY [
    ITEM HAT ; the item that lies here
    REQUIRED NOTHING ; nothing REQUIRED to TAKE that item
    REMOVABLE TRUE ; you can take that item
    EXPLORED FALSE ; true after EXPLORE has been called, FALSE to prevent TAKE
    EXPLORE [THERE IS A HAT ON A HOOK.]
    TAKE [YOU PUT THE HAT ON.]
    ACTION [] ; an action that TAKE performs in the background
  ]
END
```

Here is what this code means and the details of each of these properties for objects:

- The first property in the room's property list names the object that the player will find in the current room. The name of the property is `ITEM` and its value is what the item is, a `HAT` in this example.
- In the next line, the `REQUIRED` property indicates whether or not the item must be in Inventory when the player is in a specific room. If the value is `NOTHING`, then it is not required to complete the adventure. For example, if the game requires that the player has a `MAP` when in the `LIBRARY`, the property list in the `LIBRARY` property list would be `REQUIRED MAP`. The `MAP` would need to be an object that can be taken in a different room.

- The `REMOVABLE` property says whether or not the object in the room can be taken. Its value is either `TRUE` or `FALSE`. An object that can't be removed is not required to finish the game.
- The `EXPLORED` property is either `TRUE` or `FALSE`. Its initial value is `FALSE`. When you explore a room, the `EXPLORE` procedure sets it to `TRUE`. This prevents the player from taking an object more than once. You can also use this property to determine if the player has explored every room.
- The value of the `EXPLORE` property is what the player sees when they type `EXPLORE`.
- The value of the `TAKE` property is what the player sees when they type `TAKE`. When they take an object, it goes into their Inventory.
- The value of the `ACTION` command is what might occur in that room. There is no `ACTION` command for the player to type. The action takes place behind the scenes. It might set a property, give you a hint, or even open up a new room! Most of the time the value of action will be `[]`, or nothing.

The first set of properties in the procedure above is for the room named `ENTRY`. You can create a similar property list for each room. Give it a try!

You will need an `EXPLORE` procedure. This procedure, shown below, does two things:

1. It displays text for the player to read.
2. It changes the value of the `EXPLORED` property of that room to `TRUE`. That allows you to take it item (if it is removable).

```
TO EXPLORE
  PRINT GPROP :ROOM "EXPLORE
  PPROP :ROOM "EXPLORED TRUE
END
```

Now you need to add `SETUP_ITEMS` to the `SETUP` procedure you type to run the code.

```
TO SETUP
  SETUP_MAP
  SETUP_ITEMS
  MAKE "ROOM "ENTRY
  MAKE "INVENTORY []
  TEXTSCREEN
  CLEARTEXT
  PRINT "|You are in the main entry of the Mystery Mansion.|
END
```

This part of the adventure is a bit tricky. You will have a lot of testing to do!

Remember to save your program!