# Test Quest

## Purpose

We're making a game that can be played quickly by people who aren't necessarily familiar with lots of gaming experiences. The audience is the casual gamer who wants to play a game for about fifteen minutes to a half hour.

## Goal

The player controls a game character that is trapped in a room where the door is locked. The player has to get the game character out of the room by interacting with objects in the room.

We want to make sure the game is solvable with a basic set of steps (see Walkthrough below). But we also want to make sure that we provide a good gaming experience. The game user manual that the player will be given is provided. Some of the most necessary elements from that manual are called out here (see Game Mechanics below). There are some specific elements of design that we want to make sure are in place (see Game Design below).

## Walkthrough

The following is the minimal solution to solve the game:

- LOOK IN BOX
- TAKE KNIFE
- TAKE SPEAKER
- USE KNIFE WITH SPEAKER
- STAND ON CHAIR
- TAKE BULB
- USE SPEAKER WITH LAMP

## Game Mechanics

- To get a description of an object, the player can type "LOOK AT <object>".
- To get a description of a room, the player can type "LOOK".
    - Alternates: "LOOK AT ROOM" or "LOOK AROUND".
- To open an object, the player can type "OPEN <object>".
- To pick up an object, the player can type "TAKE <object>".

## Game Design

1. Wall and object boundaries should be respected by the game character at all times. The game character should appear appropriately 'behind' or 'in front' of on-screen game elements.

2. Words that a player uses in a command that aren't understood should result in the game printing a message to the player that indicates what word wasn't understood.

3. The game should make it clear what objects the player can interact with: boxes, lamp, barrel, chair, table, bulb, briefcase, paint can, speaker. Every object referenced should provide a description.

4. The game character has an "inventory" because they can take things. The inventory should always reflect what was taken or dropped. The inventory should show a picture of each item.

5. Objects that have a before and after state should have a before and after description. Examples are the light bulb and the box.

6. Graphically, the room views should reflect the state they are in. If something changes with respect to an object, the room views should reflect that in terms of the graphics.

7. The game should be 'situationally aware.' The game should recognize the game character is near enough to objects to try certain actions. If the player is not near enough to an object, an action on that object shouldn't be allowed. (Unless the action doesn't require proximity to the object.)