

Main - I Must Play

[I Must Play](#) is my entry in [IFComp](#) 2004. The game is sort of like the [IFArcade](#) games, but those games were all fairly short and simple. [I Must Play](#) is fully fleshed out in ways that are hopefully humorous.

I Must Play now even has its very own [WURB page](#) !

Note that if you're stuck in the game you can type "walkthrough" in the game and it'll give you a URL to where to get the walkthrough.

Reactions thus far:

- Reviews on [rec.games.int-fiction](#) for my game are pretty mixed. Clearly some people didn't like the subject matter and as a result didn't like the game at all. Others liked it but sadly weren't wowed by it. That's too bad because my favorite project theme is classic video games.

Things I have learned while beta testing this game:

- When people see something labeled "power switch" they for some reason try to flick it, push it, turn it and hit it. But for some reason they don't try to turn it on or flip it. I found that very amusing. Nothing else in my game seemed to cause any guess the verb problems. But that switch just made life hard for people until I added all the extra verbs.

I made a few interesting things when developing this.

- A callback system that calls an arbitrary method when a player enters a group of rooms and then calls a different method when the player leaves those rooms. For instance, let's say I have have a daemon running in one room that outputs text like "the ball is getting closer" every turn. If this should stop executing when a player exits a certain group of rooms; then if the player leaves this group of rooms, no matter how they leave, this callback will stop the daemon.
- A class I call `FurtherDelayableAgendaItem` that allows you to have an agenda item delayed a few additional turns after the original requirements where met. For instance, if you want an actor to make a comment only after the player has accomplished a certain task, but you want them to wait an additional 2 turns after that, then you would use `FurtherDelayableAgendaItem`.
- I figured out how to automatically call a script when starting the game in debug mode. Then, to start in different states of the game I just need to edit a `start.cmd` file and then start the game.
- I created a macro called `perInstanceWithParent` which does the same thing as `perInstance`, plus it sets the `instanceLexicalParent` property in the newly created object to point at the parent (like you would expect the `lexicalParent` property to be set).

LINKS:

- [Tads3Lang](#)
- [Interactive Fiction](#)