

Items

Monday, October 20, 2014

9:42 AM

Requirements

1. Items with player or in places
2. Pick up or put down
3. Items required/forbidden for particular actions

Note: The objects with a star in front are the ones I have decided are best suited for the purpose at hand.

Place

- Label, desc
 - ★ • B. Items[] items
1. Foyer
 2. Hallway
 3. A. Player

Action

- Label, desc
- From, to
- ★ • Reqs, forbs
- ★ • Get/set methods() for req/forb items
- ★ • Method that checks for player items and compares them to the reqs/forb fields

Game

- Label, desc
- Places
- Actions
- Item[] items

Item

- Label, desc
- A. place

Player

- Label, desc
- ★ • B. Items[]



abstractAction

- moveAction PtoP
- TakeAction I
- DropAction I

Design

A. new Item("key", "an ornate gold key", place,



New Place ("t048", "basement room", item, item, item, or items[])

new Action ("open", "you open door", hall, lockedroom)

new Action ("open", "you open door", hall, lockedroom, item:key)

new Action ("enter", "you enter airport", curb, airport)

new Action ("enter", "you enter airport", curb, airport, gun)

new Action (lab, desc, fr, to, reqItems, frbItems)

new Action ("open box", "You open Pandora's box, and..." , boxroom, boxroom, {key},
{fuzzy mittens})



New Action ("take key", "you take key", classroom, key)

place item



New Action ("dropkey", "you drop key", key, classroom)

Item place



Method that checks for player items versus required/forbidden items

For instance: if player.items.contains(required items) {...}

else if player.items.contains(forbidden items) {...}