Classok

* OnEvent interface ??
* Item <<abstract>>
  + description
    - ami leírja, hogy mit csinál a tárgy
    - ha rávisszük a kurzort a tárgyra, kiírja a gui
  + id: Int
    - minden tárgy felülírná a sajátjával
    - a megjelenítés miatt amúgy is kell
    - lehet ehelyett toString()
  + onPickedUp()
  + onRoundStart()
  + onAttacked()
  + esetleg plusz onValami() függvények
  + action()
  + Söröspohár
  + Rongy
  + Dobókocka
  + Camembert
  + FFP2
  + Tranzisztor
  + TVSZ
  + Logarléc
  + Tranzisztor
  + List<MyInterface> list = List<MyInterface>();
  + list.add(myItem);
* Csöcs Balázs
  + absztrakt
  + showTiddies()
* Karakter
  + bool védett
    - tárgy onAttacked függvénye
  + int masked - mennyi körig van még rajta maszk
  + Hallgató
  + Tanár
* Game Controller
  + Körök kezelése
  + Szobák kezelése
  + Tanárok irányítása
  + Játék irányítása:
    - Mindenkinek ad számot
    - Körök figyelése *(Karakter listát végignyálazza)*
    - Szobák kezelése *(Minden kör elején)*
    - Logarléc
* GameInitializer
  + main, ami előállít mindent, ami a játék elindításához kell
    - pl pálya beolvasása
    - karakterek létrehozása
    - majd képek beolvasása
    - itemek létrehozása
    - ezeket átadja konstruktorban a game controllerbe
    - egy lista, amiben
* Szoba
  + Tranzisztor változó
    - Ha nincs tranzisztor, akkor = null
      * így könnyen lehet checkelni, és nem típuscheck
  + lista változó amely tartalmazza a hallgatókat
  + lista, amely tartalmazza, hogy hova lehet belőle eljutni
  + lista, amely tartalmazza, hogy honnan tudtak hozzá eljutni
    - bár belegondolva, ezt nem nagyon használjuk ki
  + onEntered() függvény
    - hogyha van effektje a szobának belépéskor a playerre
    - gázos szobánál stun
    - regular szobánál semmi
  + regular szoba
  + gas room

Szekvencia Diagram

Dani

Jeges

Done

1. Karakter mozgás
   * check, hogy az ajtó iránya megengedi-e
   * szomszédos szobába szeretne-e menni
2. hallgató felvesz egy tárgyat
   * bekerül inventoryba, és meghívódik az onPickedUp()
3. tanár felvesz egy tárgyat
   * tárgy destroy
   * kiszedni a tárgyak listája közül
4. tárgy letevése
   * kiszedni a player inventoryból
   * szobához hozzáadni
5. tárgyhasználat
   * meghívni a tárgy use függvényét
6. szobák osztódása
   * as in the use case (1. beadás)
7. szobák egyesülése
   * as in the use case (1. beadás)
8. körök kezelése
   * gamecontroller meghívja a karakterek action függvényét x-szer
9. StudentGoesToInstructor
10. InstructorGoesToStudent
    * az egyiknél azt kezeli le, hogyha a szobába belép egy tanár, akkor a benne lévő hallgatók meghaljanak
    * a másik meg azt kezeli le, hogyha a szobába egy hallgató lép be, akkor haljon meg xd
    * Jah rájöttem. Pedig majdnem ujrahasználható a kód

Amúgy +2 easy diagram

Jó az

De csak addig csináld meg hogy removePlayer()

Mert azt már én csinálom

1. Logarléc felvétele :/
2. Character action függvénye
   * meghívja a tárgyainak a onRoundStart függvényét
   * loop az actionök számával
     1. megkérdezi, hogy milyen actiont hajtson végre (TODO)
        1. getActionFromPlayer()
3. Tranzisztor használata - “teleportálás tranzisztorral”
   * Milyen függvénnyel? Hol hívódik meg
   * leteszünk 2 tranzisztort
   * szoba eltárolja a tranzisztort
   * mindvégig van egy 5. opció, “tranzisztor ugrás” néven
     1. guiban csak akkor jelenik meg, ha jogosult vagy rá
   * amikor kiválasztod ezt az 5. opciót, akkor a mezőn lévő tranzisztor -> párjának -> szobájába áttesz téged a “tranzisztor ugrás”, és kikapcsolja a 2 tranzisztort
   * ÁÁÁÁÁÁÁÁÁ
   * Ha ezt az 5.action-t választod ki, mikor nem vagy jogosult, akkor nem fog történni semmi
4. GameController removePlayer
   * kiszedi a listából a playert, és checholja, hogy utolsó volt-e
     1. ha igen, beállítja a gameEnded-t true-ra
5. Tranzisztor párosodása az inventroyban
   * TODO kidolgozni a tranzisztor classot
6. Camambert letevése
   * szoba “gázosítása”
     1. új, gázos szoba létrehozása ugyanazokkal a tárgyakk al, paraméterekkel
     2. régi szoba törlése
7. Szobába belépés
   * szobának az onEntered() függvényének hívása

Állapotgép ötletek

1. isVédett
2. isMasked
3. isAlive
4. Tranzisztor
   1. bepakcsolás
5. Game
   1. a játék még nem indult el
   2. a játék fut
      1. hallgatók száma == 0 => Tanárok nyertek
      2. logarléc found => Hallgatók nyertek