Room Field = new Room

{

name = "A Field",

information = "sdfsdf",

characters = new List<Character>(),

roomInventory = new Dictionary<string, Item>{

{"north", ItemSetup.gear[0]},

}

};

Wie macht man das roomInventory? Kein Zugriff auf Items direkt, da Item nicht static ist.

Auch kein Zugriff auf Characters durch CharakterSetup.link z.B.

List<Character>()/ Dictionnary<string,Item> speichert nur Character /Item. Eigenschaften von Enemy, Avatar, Friend / Gear, Health gehen verloren. Wie organisiert man alle drei Typen in ein Dictionary/ eine Liste.

Field.neighbours = new Dictionary<string, Room>

{

{"north", Kakariko},

{"south", null},

{"east", null},

{"west", null},

};

Kakariko.neighbours = new Dictionary<string, Room>

{

{"north", null},

{"south", null},

{"east", ZoraDomain},

{"west", SheikahShrine},

};

ZoraDomain.neighbours = new Dictionary<string, Room>

{

{"north", HyruleCastle},

{"south", null},

{"east", null},

{"west", Kakariko},

};

SheikahShrine.neighbours = new Dictionary<string, Room>

{

{"north", null},

{"south", Field},

{"east", Kakariko},

{"west", null},

};

HyruleCastle.neighbours = new Dictionary<string, Room>

{

{"north", null},

{"south", ZoraDomain},

{"east", null},

{"west", null},

};

Neighbours können erst zugeteilt werden, nachdem der Raum deklariert wurde. Gibt es da einen schöneren Weg?

Room SheikahShrine = new Room

{

name = "The Sheikahshrine",

information = "sdfsdf",

characters = new List<Character>(),

roomInventory = new Dictionary<string, Item>()

};

rooms.Add(SheikahShrine);

Room HyruleCastle = new Room

{

name = "The castle of Hyrule",

information = "sdfsdf",

characters = new List<Character>(),

roomInventory = new Dictionary<string, Item>()

};

rooms.Add(HyruleCastle)

Hier genauso. Add- schönerer Weg?

Müssen Items auch in einer Liste gespeichert werden, so wie Räume?