Starting Development Structure Inheritance Map

Abbiamo una mappa per spiegare meglio le eredità.

**TDLSprite**: è la nostra classe padre principale. Il nostro noumeno. Estende direttamente Phaser.Physiscs.Arcade.Sprite ….. Perche? Perché abbiam considerato che per avere la classe più generale possibile dovevamo poterci mettere dentro sia gli oggetti che hanno bisogno di fisica, sia quelli che non ne hanno bisogno. Per sopperire al problema abbiamo definito che gli oggetti senza fisica, avranno il body=null, mentre gli oggetti con fisica avranno il body fisico.

In questa classe mettermo le proprietà e i metodi che dovranno interessare tutti gli oggetti, nel caso dovessimo aggiungerne rispetto ai metodi già ereditati da Ph.Ar.Sprite.

**TDLCharachter**: è la classe che comprende tutti i nostri personaggi. In questa classe metteremo tutte le proprietà e i metodi che serviranno ai nostri personaggi, come ad esempio le generazioni generali che ogni personaggio sarà in grado di fare. Ad esempio camminare.

NPCharacter: classe finale per I personaggi controllati dal computer

Player: classe finale per qualsiasi personaggio controllato dal giocatore (es faraone, personaggio principale o altri)

**TDLObject**: Oggetti in generale.

**WorldObject**: Tutti gli oggetti che andranno dentro la room. La capacità di essere interattivi verrà definita come un parametro che verra settato true o false e una funzione che dovrà essere aggiunta in base all’oggetto.

**RoomFurniture**:

**InGameInventroyObject**:

**WallFurniture**:

**Transitioner**: ad esempio le porte scale o qualsiasi cosa ci poti da una scena all’altra.

**LightSource**: Tutte le luci che verrano create nella stanza. Le luci creano anche delle luci del motore grafico di Phaser. Le luci vengono gestite dal LightSourceManager

**Lamp**: tutte le fonti di luce attive, come lamapade. Avranno una doppia luce grafica, una diffusa e una concentrata.

**NoSpriteSource**: Per tutti i casi in cui avremo luci che vengono generate da oggetti che sono parte di animazioni di personaggi o di altre cose.

**OnlyDiffusedLightSource**: Per oggetti che crano luce diffusa e basta, come ad esempio finestre e porte aperte. Fonti di sola luce diffusa.

**InventoryObject**: Solo gli oggetti che andranno nell’inventario e staranno sempre fuori dalla room. Al massimo verranno trascinati. Quando un oggetto viene usato su un componente del gioco, si distruggerà questo oggetto e ne si creerà uno dei WorldObject relativo, perché c’è la classe InGameInventoryObject che eredita da WorldObject.