Assignment 1
Entity Relationship Modeling
Points: 50

Instructions: For this assignment you must submit a **picture** file(s) of the ER diagram that must be drawn using any ER modeling tool or any digital drawing tool. “Handmade” diagrams won’t be accepted. I recommend using TerraER but others can be used.

Consider an application that stores information about “point and click adventure” games similar to “The day of the Tentacle” or “Monkey Island”.

Note: Of course, these games are a lot more complicated than the description that I’m providing and surely a lot of them are not using at all this design of database to store the information, but remember that this is just an assignment, not a real life exercise.

In case you don’t know them, for example in The day of the Tentacle you have some characters that can interact with elements in the screen, pick items, conversate with other characters in the game, etc (more details below).

The data requirements of our database are summarized as follows:

- We can have more than one game in the database.
- For each game we will store the id (unique), the title, the description, the year of publication, and the company or companies that created it.
- Each game has been published at least by one company (sometimes there are collaborations between companies to release one game, so a game can have more than one company too), and we can have companies without games released (maybe they are new and they are still developing their new games).
- A company will have a name, an id (unique) and an address (can be just one string attribute, no need to divide it).
- The games must have scenes (at least one). The scenes are the different scenarios that we will see in the game. For example in the picture we can see the scene called "Laundry" from the game Day of the Tentacle.
- The scenes will belong only to one game.
- Each scene will have an name (unique for each game, but 2 scenes on 2 games can have the same name), one and only one background image, could have many items, and on that case we will need to know the position x and position y of the items in the scene, and it can have "AI characters" too with the the initial position of the character.
- An item can be related to several scenes (but maybe it is not used on any scene at all...).
- Each item will have an id, a name (ex: coin), a description, and a list of images to use to display the item, because the item can be animated and have more than one image.
- Each item will have a list of actions supported (one at least or more).
- To keep it simple, each action will just have an attribute with the command allowed (for example: "Look" or "Open"...)
    - Note (as said, in a real game we would need a lot of more information… but for our exercise this is enough)
- We will have 2 types of characters, "Playables" (the ones the player will use) and "AI" (artificial intelligence.,.. the ones that will interact with the character)
- A character will have an Id, a name (that maybe is not unique) and a list of images (meaning all the different images that the character can use, for example "Bernard0001.jpg", "Bernard0002.jpg", etc...)
- The games can have Characters … at least one playable character per game, but some games don't have AI characters.
- The characters belong always to one and only one game.
- The AI characters can be associated with many scenes, maybe to more than one or to none. The playable ones won't be associated with scenes because they can move to any scene.
- Each character, item and scene must have at least one image related

- Each Image (we will store all the images information in the same table) will have a unique id, the path of the file , and a description.
- The same image can be used for items, characters and backgrounds (so, we don't have different types of images). The same image can be used on different items (we can have for example two types of coins that use the same image)… but if used in a character, it will be used only in one character and if used as background, it will be used only in one scene. And we can have images in the database not used at all anymore, just in case for future use.

Design an entity-relationship diagram for the streaming platform database described above. Along with all the relevant entities, relationships, and attributes make sure to specify the following:

- **Key** attributes of entities (remember, key attributes are underlined)
- **Cardinality ratio and participation** constraints for relationships (Preferred in Chen notation, that means double line or single line for participation and a number for cardinality, but other notations can be accepted too if your ER diagram tool has not Chen notation)
- **IMPORTANT: Any assumptions you make that are not in the requirements, but affect your design must be written in the document.**

RUBRIC:

20 points: for entities and their attributes

- -4 points for missing entity;
- -1 for missing attributes or incorrect attribute type
- -2 for missing keys
- -1 point if any weak entity is not defined correctly

15 points: for relationships with correct participating entities and attributes where applicable

- -3 points for missing relationship
- -1 for missing attributes or incorrect attribute type
- -2 for missing keys

10 points: for cardinality

- -2 points for missing or wrong cardinality

5 points: for participation constraints

- -2 points for missing or wrong participation constraint