**What still needs to be done**

First slide : Overview

We managed to extract the data, now we have to process it. Here are the main objectives we have set ourselves *(schema)*.

Second slide : Improve the PCA, Select different axes

Our PCA allows us to reduce our vectors to two dimensions. However there is a loss of information. The two axes chosen to represent the vectors are the axes carrying the most information. The graph shows us that there are different eigenvalues and therefore different possible axes to represent. The higher the eigenvalue, the more important the information it carries. But it can be interesting to visualize the other axes carrying other information. This may allows the user to discover other similarities between our vectors.

Third slide : Improve the PCA, change PCA type

There are different types of PCA : linear and non-linear. Linear PCA is the most common. But a non-linear PCA can reduce the size of the vectors with less data loss. It is interesting to know what PCA type describes better our data. This would allow the user to obtain more information. On the graph it seems obvious that a linear PCA is not suitable for all types of data.

Fourth slide : Question Tree

As said before, a question tree allows to reduce the number of possible characters. The first 14 questions are calculated thanks to an algorithm. Our goal is to display the question tree. Of course this tree is updated if the data changes. The tree can potentially stop before the first 14 questions. But it does not complete all possible characters with only 14 questions. We want to associate a key character to each leaf. For example the most requested character of this branch.

The tree is large because it is a binary tree with a depth of 14. So it will have to be interactive. We have already found with D3.JS *(slide)* a way to display the tree. We still have to extract the questions from the database and process them.

Fifth slide : Get information from a group

PCA makes it possible to group together characters that look the same *(slide)* . Our goal is to determine how characters look alike. We select a group of characters on the graph and the questions/answers that define this group are displayed. We need to get the meaning of the axes of the PCA. We then have to determine the responses of the selected characters to this axis. PCA reduces the size of the vectors. There is a loss of information. This intermediate step would allow us to know the differences between an individual and a selected group.

Sixth slide : Replace a vector in a group

A character who looks like one group may find himself away from another. We want to put him back with that group. Knowing the answers defining this group, we can answer the missing questions to move this character. For example *(slide)* , this circle is not located with the other circles. We want to know the data characterizing the group in order to be able to replace it. This would allow us to quickly fill a data gap. However, this is not automatic and requires human attention. Since we modify the character's data by answering the questions, it might be possible to directly move a character into a corresponding group and modify/complete his answers. One of the difficulties is the large size of the database. It has a lot of questions and characters. Some characters are close to each other without having any real similarities. Moving a character would imply changing the data of that character in the database.

No slide : quick « conclusion » :

These different functionalities will allow a user to visualize the data. He will be able to understand the data more easily. Moreover it will allow to complete data without playing.

**What still needs to be done (Summary) :**

1st slide Main objectives

2nd slide Select different axes :

* Two dimensions = loss of information
* Eigenvalue and axes
* Visualize other axes for more information

3rd slide Change the PCA Type :

* Linear and non-linear
* Depend on the type of information
* Lose as little information as possible

4th slide Question Tree :

* Tree calculated from an algorithm
* Depth of 14 = a lot of data
* Interactive thanks to D3.JS

5th slide Get information from a group

* PCA groups vector
* Vector’s information
* What defines a group (questions/answers)

6th slide Replace a vector in a group

* Differences between a vector and a group
* Possibilty of complete vector’s dara
* Drag a vector into a group (seems hard)