**Social Network Interaction Documentation**

The current document presents a description of the public variables and functions which are available to the user.

**socialnetworkinteractionclass.h**

**Class: KPartiteParams**

Contains the below variables used to the K-Partite graph’s configuration.

int MaxFRRuns:

Number of iterations for computing the vertices’ position.

float dampenLast:

Damping factor.

float Dimension:

Dimension of the graph.

float restDistance:

Rest Distance.

float repulseConstant:

Repulse constant.

float springConstant:  
 Spring constant.

**Class: SocialNetworkInteractionClass**

Public Variables:

KPartiteParams kparams

Public Functions:

**SocialNetworkInteractionClass** ();

Description: Creates an object of class **SocialNetworkInteractionClass**.

~ **SocialNetworkInteractionClass** ();

Description: Destroys the object of class **SocialNetworkInteractionClass**.

void **addFile**(string inputPath)

Description: Function that gets as input the path of data csv file(s) and pushes them back into a vector.

**Parameters: 1. inputPath** –path of data file(s).

void **computePositions**()

Description: Function that reads the data from the vector of the string(s) of input file path(s) and computes the positions of vertices and the number of connections of each node, of the main K-partite graph.

void **computeAbstractPositions**()

Description: Function that reads the data from the vector of the string(s) of input file path(s) and computes the positions of vertices and the number of connections of each node of the Abstract graph.

void **savePositionsToFile**(string outputPath)

Description: Function that saves vertices’ position and number of connections to a file.

**Parameters: 1. outputPath** –path of output file.

void **setMaxFRRuns**(int MaxFRRuns)

Description: Function that sets the value of the KPartitesParams’s MaxFRRuns variable to the new passed value.

**Parameters: 1. MaxFRRuns** – Number of iterations for computing the vertices’ position -Default value: 300

void **setdampenLast**(float dampenLast)

Description: Function that sets the value of the KPartitesParams’s dampenLast variable to the new passed value.

**Parameters: 1. dampenLast** – Damping factor -Default value: 0.65

void **setDimension**(float Dimension)

Description: Function that sets the value of the KPartitesParams’s Dimension variable to the new passed value.

**Parameters: 1. Dimension** – Dimension of the graph - Default value: 3000.0

void **setrestDistance**(float restDistance)

Description: Function that sets the value of the KPartitesParams’s restDistance variable to the new passed value.

**Parameters: 1. restDistance** – Rest distance -Default value: 300

void **setrepulseConstant**(float repulseConstant)

Description: Function that sets the value of the KPartitesParams’s repulseConstant variable to the new passed value.

**Parameters: 1. repulseConstant** – Repulse constant - Default value: -1 (if the user input a value that is greater than zero, the algorithms accepts selected value. Otherwise, the algorithm uses a default value).

void **setspringConstant**(float springConstant)

Description: Function that sets the value of the KPartitesParams’s springConstant variable to the new passed value.

**Parameters: 1. springConstant** – Spring constant - Default value: -1 (if the user input a value that is greater than zero, the algorithms accepts selected value. Otherwise, the algorithm uses a default value).

void **setKPartiteParamsValues**(int MaxFRRuns , float dampenLast, float Dimension, float restDistance , float repulseConstant, float springConstant)

Description: Function that sets the values of the KPartite’s basic parameters to the new passed values.

**Parameters: 1. MaxFRRuns** – Number of iterations for computing the vertices’ position -Default value: 300

**2. dampenLast** – Damping factor -Default value: 0.65

**3. Dimension** – Dimension of the graph - Default value: 3000.0

**4. restDistance** – Rest Distance - Default value: 10.0

**5. repulseConstant** – Repulse constant - Default value: -1 (if the user input a value that is greater than zero, the algorithms accepts selected value. Otherwise, the algorithm uses a default value).

**6. springConstant** – Spring constant - Default value: -1 (if the user input a value that is greater than zero, the algorithms accepts selected value. Otherwise, the algorithm uses a default value).