

**Philosophische** Fakultät III

Sprach-, Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Medieninformatik

Interaktionstechniken und –technologien (ITT)

Modul: MEI-M 32.1 + 2 (M.Sc.)

SS 2017

Leitung: Dr. Raphael Wimmer

IPlanPy – The New Charting Tool

Sebastian Peiser, Julia Sageder

?, 11694688

Medieninformatik

1. Semester M.Sc., 8. Semester B.A.

Email:

[Sebastian.Peiser@stud.uni-regensburg.de](mailto:Sebastian.Peiser@stud.uni-regensburg.de)

[Julia.Sageder@stud.uni-regensburg.de](mailto:Julia.Sageder@stud.uni-regensburg.de)

Abgegeben am 07.08.2017

**List of figures**

[Figure 1 – User Interface 5](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698954)

[Figure 2 – Connection Menu 6](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698955)

[Figure 3 – Save/Load Menu 6](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698956)

[Figure 4 – Card Connections 6](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698957)

[Figure 5 – Card Connection 6](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698958)

[Figure 6 – Standard Card Type 7](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698959)

[Figure 7 – Header Card Type 7](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698960)

[Figure 8 – Card Colors 7](file:///C:\Users\LocalAdmin\Desktop\Dokumentation_ITT_Peiser_Sageder.docx#_Toc489698961)

Contents

[1 Concept and Usage 4](#_Toc489698940)

[2 System 4](#_Toc489698941)

[3 Implementation 5](#_Toc489698942)

[3.1 iplanpy.py 5](#_Toc489698943)

[3.2 iplanpy.ui 5](#_Toc489698944)

[3.3 connectionmanager.py 6](#_Toc489698945)

[3.4 card.py 7](#_Toc489698946)

[3.5 gestureclassifier.py 8](#_Toc489698947)

[3.6 shake.csv 8](#_Toc489698948)

[3.7 vectortransform.py 8](#_Toc489698949)

[3.8 wii.motes 8](#_Toc489698950)

[3.9 wiimote.py 8](#_Toc489698951)

[3.10 demo.chart 8](#_Toc489698952)

[4 Cheat Sheet 8](#_Toc489698953)

# Concept and Usage

In everyday life, organizing is an important part to be prolific and efficient. Not only in private, also in professional life it’s necessary to have structures. Especially at work there are many cases where to build a structure is a big advantage for planning e.g. a work process/tasks etc. This is where our system steps in: IPlanPy is the perfect solution for creating diagrams. It is easy to handle and a great way of presenting, for example, abstract processes, hierarchies or complex systems. IPlanPy is designed for cooperation work in a team, especially with the Wiimote, but can also be used from a single person simply with the mouse instead of the Wii-controller. The support of collaboration is an important feature of the system. Sketching with IPlanPy will prove the capacity for teamwork in that team, because the best way of usage is in splitting the input roles at two different people. There is an necessity to communicate then and additional to this a way for getting a bigger space for ideas and creativity, because “two heads are better than one”[[1]](#footnote-1). Certainly our system can also be used as a single user, just with a mouse and a keyboard. The areas of application are not limited at all. In every use case where a chart is needed, IPlanPy is your system to use.

# System

To use our system IPlanPy you need a laptop or computer with Linux, a keyboard and a mouse (for single usage) or a Wiimote and therefore also IR-Sensors (for team usage). To get the best performance your system may have the following requirements: ?

With IPlanPy it is possible to build diagrams. The system supports the following features:

* Create a new card
* Switch the card type between:
  + Standard (Title and Field)
  + Header (Title)
* Delete a card
* Change the card color between prescribed colors
* Build a connection between two cards presented as a line from card middle to card middle
* Delete a connection between two cards (Undo)
* Delete all connections from one card
* Save your chart
* Load your saved charts
* Connect your Wiimote (Stores connected Wiimotes automatically)
* New chart (Rejects all unsaved data and clears the screen)

# Implementation

## iplanpy.py

This is the main python script, where all the corresponding strings run together. It contains the class *IPlanPy*, which includes all relevant handling processes. The user interface of IPlanPy is loaded from the **iplanpy.ui** script, which contains all start-widgets. The cards are handled in the **card.py** and the connections in the **connectionmanager.py** python scripts. [……]

## iplanpy.ui

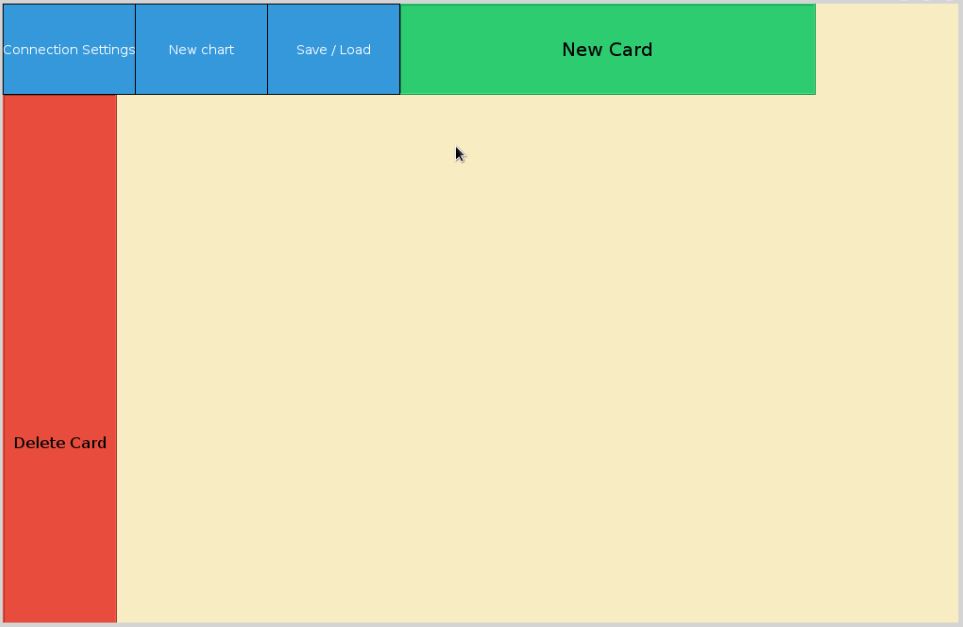
The user interface was built in the *Qt Creator (Community)* and contains all start-widgets. This are the following buttons and labels: “New Card”, “Delete Card”, “Connection Settings”, “New Chart” and “Save/Load”. It is loaded in the *iplanpy.py* script.

Figure 1 – User Interface

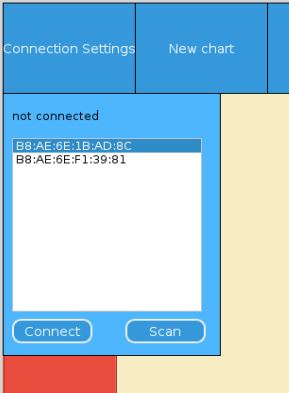
The “Connection Settings” and the “Save/Load” contain in addition menus.

Figure – Connection Menu

Figure – Save/Load Menu

## connectionmanager.py

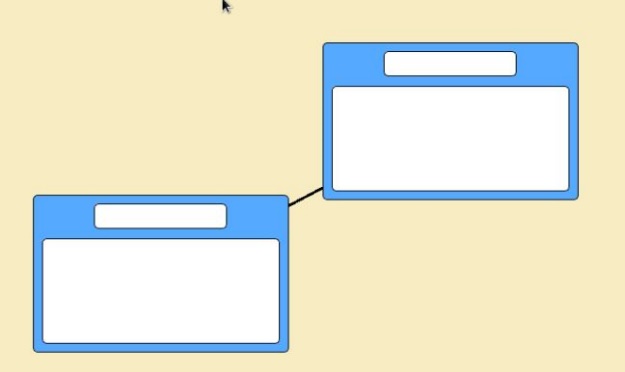
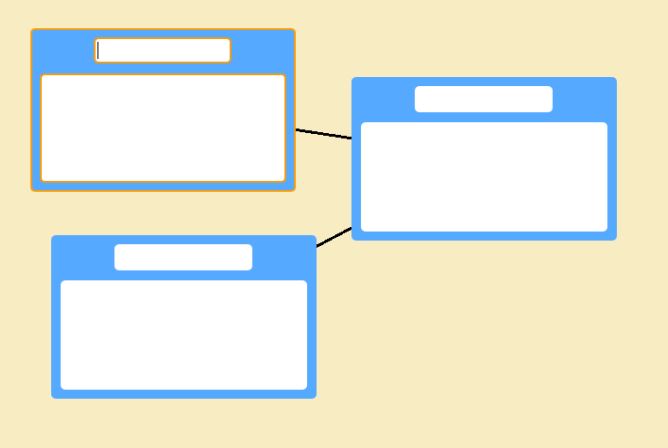
The **connectionmanager.py** script contains the *ConnectionManager* class, which handles the connections between the cards. The definitions contained are:

Figure – Card Connections

Figure – Card Connection

*connect*: This one saves a new connection and proves first if it isn’t already existing.

*delete\_all\_card\_connections*: deletes all connections where the handed over card is involved. In addition the deleted connections are saved in the restorable\_connections list to make them flexible for undo and redo.

*remove\_last\_connection*: deletes the last connection of the *connections* list and restores it in the *restorable\_connections* list to make it flexible for undo and redo.

*restore\_connection*: restores the last deleted connection (if one exists) and adds it to the connections list for redo.

*get\_centers*: determines the centers of the two cards where a connection should be build.

## card.py

The **card.py** script contains the card class, which is a *QFrame*, representing the cards of the system. The *\_\_init\_\_* function establishes the main properties of the *Card*s, such as *DEFAULT\_COLOR* or *available\_colors* list for example. The *setup\_ui* creates the user interface of the created *Card*, which contains a *QLineEdit* *title\_field* and a *QTextEdit* *content\_field*. There is also a type of *Card* where the *has\_text\_field* is *False*, and so there is only a *title\_field*. This switches in the *toggle\_type* and the particular *setup\_* definitions.

Figure – Standard Card Type

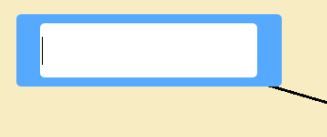


Figure – Header Card Type

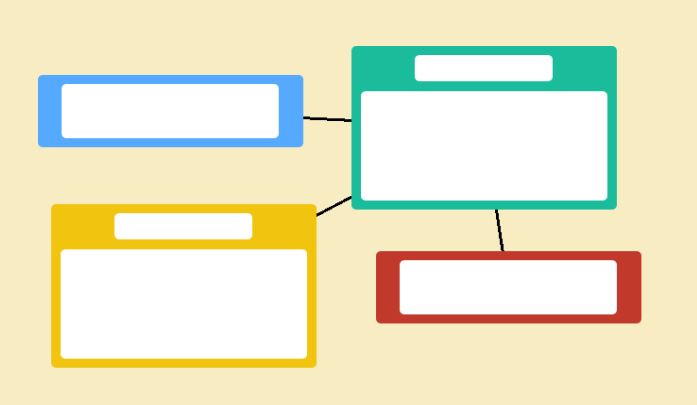
The colors of the *Card*s can be switched by the *next\_color* and *previous\_color* definitions, which handle this by iterating over the *color\_index* list. […]

Figure – Card Colors

## gestureclassifier.py

## shake.csv and steady.csv

This csv file contains the values for classifying the *shuffle gesture* with the **gestureclassifier.py** script. The **steady.csv** file contains the converse data when the wiimote is kept still.

## vectortransform.py

## wii.motes

**wii.motes** contains the saved addresses of the known (already connected) Wiimotes. It is used in the

## wiimote.py

This python script is provided of the university lecturer of the course, Dr. Raphael Wimmer. It is a Wiimote wrapper in in Python 3.[[2]](#footnote-2)

Based on this script it is able to handle the Wiimote callbacks in the **iplanpy.py** main script.

## demo.chart

The **.chart** files represent the saved charts of the user. They encode the card and connection data of the saved chart in a csv-like format and can be loaded to continue the work on previous (saved) charts.

# Cheat Sheet

The list of all possible interactions with IPlanPy:

|  |  |
| --- | --- |
| Create a new card | Mouse: Click Button “New Card”  Wii: Cursor over “New Card” + Button B |
| Focus card | Mouse: Click card  Wii: Cursor over card + Button B |
| Switch the card type | Mouse: Cursor over card + Alt + Left/Right  Wii: Cursor over card + Left/Right |
| Delete a card | Mouse: Drag and Drop card to “Delete”  Wii: Drag and Drop with Button B to “Delete” |
| Change the card color | Mouse: Cursor over card + Alt + Up/Down  Wii: Cursor over card + Up/Down |
| Build a connection between two cards | Mouse: Drag and Drop card to card  Wii: Drag and Drop with Button B to card  OR (Redo case)  Mouse: Alt + Plus  Wii: Plus |
| Delete a connection between two cards  (Undo) | Mouse: Strg + Minus  Wii: Minus |
| Delete all connections from one card | Mouse: Not possible! Wii exclusive!  Wii: Focus card + shuffle gesture |
| Save your chart | Mouse: Click “Save/Load” + new chart name + Click “Save”  Wii: Cursor over “Save/Load” + Button B + new chart name + Cursor over “Save” + Button B |
| Load a saved chart | Mouse: Click “Save/Load” + select chart + click “Load”  Wii: Cursor over “Save/Load” + Button B + select chart with Button B + Cursor over “Save” + Button B |
| New chart | Mouse: Click “New Chart”  Wii: Cursor over “New Chart” + Button B |
| Connect your Wiimote | Mouse: (Click Scan +) select your Wiimote + Click “Connect”  Wii: Not possible! Just for changing performing Wiimote! |

1. <https://en.wiktionary.org/wiki/two_heads_are_better_than_one> [↑](#footnote-ref-1)
2. Copyright © 2014 Raphael Wimmer <Raphael.wimmer@ur.de> [↑](#footnote-ref-2)