Scientific Figures and Presentations in Ipe
Winter Term 2026/27
scale.iti.kit.edu

Karlsruhe Institute of Technology

Exercise Sheet 3: Layers, Views, Pages

Submision due by December 13

1 Snappy's Favorite Things

This task is a mostly guided introduction to the basic features for creating/editing presentations in Ipe, but also covers some more advanced topics and unguided challenges.

It consists of multiple subtasks that you should all complete in the same file. As a reference or if anything is unclear, you can peek at *snappys-favorites-done.pdf* to see what your final result should look like. You will find instructions regarding the submission at the end of Section 1.3.

1.1 Presentation & Alignment Basics

Open the file *snappys-favorites.ipe* and follow the instructions.

1.2 Links

Snappy has some follow-up requests for the presentation: when presenting the pdf file, you should be able to jump back to the overview slide from every page by clicking at some icon in the corner (you can find the icon that Snappy prepared for you in the file *link.png*). Also it should be possible to jump forward to a certain topic from the overview slide by clicking on the respective title. Finally, the link to Snappy's favorite website should be clickable and extended by a clickable QR code.

Realize this as follows:

A Link to the Past

- On page 3 ("My Favorite Colors"), import the image *link.png* [Ctrl + Shift + O] (or [File → Insert image]).
- Scale down the image to exactly 20% of its original size [Ctrl + K].

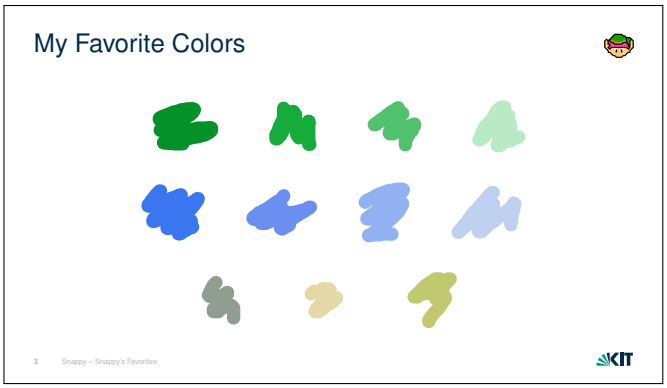


Figure 1: Example placement of the link icon in the upper right corner

- Cut out the icon from the image (removing the white space) by adding a clipping path. For this, draw a polygon [Shift + P] around the icon (it doesn't have to be perfect) and group the image [Ctrl + G] (without the polygon). Then, select both the polygon and the image and do [Right click → Add clipping path].
- Move the clipped icon somewhere to the upper right corner; it should look more or less like in Figure 1.
- assign the section title [Ctrl + P] "overview" to the overview slide (page 2).
- Set the link action of the icon on page 3 to `goto:overview` [Right click on the image → Set link action]. You can test the link action by saving your file as pdf, opening it in a pdf viewer of your choice (not Ipe), and clicking on the icon.
- Copy over the finished icon to all subsequent pages.

A Link to the Future Add section titles to pages 3-6 and assign link actions to the paragraph objects on the overview slide (page 2) as described above so that you jump to the respective pages upon clicking on the paragraphs.

A Link to the Internet

- Import the Latex packages *url*, and *qrcode* in the preamble [Ctrl + Shift + P].
- Go to the page "My Favorite Website" and edit the label [Ctrl + E].
- Wrap the link with the command `\url{}`
- Copy the link and set it as a link action to the (grouped) label object (without the prefix `goto:`).
- Add another label [L] which contains the same link, wrapped in the `\qrcode{}` command and center-align it with the page [Shift + C].
- Set the link action of the QR code label to the same link as well.
- Move the other label below the QR code and center-align it horizontally [Shift + H] with the page.

A Link to a Video? Importing video files in Ipe also requires the creation of a link action. Add one more page to the presentation [Pages → New page] with the title "My Favorite Video" which should automatically start playing the video file *hyperbolic.mp4* in an infinite loop as soon as you open the page[1]. For that, first move the video file to the same folder as the Ipe file *snappys-favorites.ipe* that you have been working on in this task. Then, add a box, group it and add a link action as follows.

Similarly to the prefix `goto:` we used above for navigating inside the file, for launching media, you need the prefix `launch:`. After the prefix, simply add the name of the video file. For additional options such as autoplay, looping, or playing only a certain part of the video, the following suffixes are available (you can use more than one suffix): `autostart`, `loop`, `start=`x, `stop=`y, where x and y should be speficied in seconds. The suffix starts with a `?`, and, if multiple options are used, are separated with an `&`. The following example would play the video test.avi automatically, but only the part between 3 and 10 seconds.

```
launch:test.avi?autostart&start=3&stop=10
```

Your box should have the same aspect ratio as the video file since otherwise the video will be visually stretched. To achieve this, you can e.g. extract a frame from the video file. Instead of a box, you can also directly use an imported image file of the

[1] Assuming that your PDF viewer supports playing videos. For example, you can use pdfpc to test this.

extracted frame and add the link action to it if you prefer. You do not have to add the title of this page to the index page (Snappy wants this page to be a surprise for their audience).

1.3 Handout Version

Create a second version of the presentation. This version should only contain the following views:

- Page 2
- Page 3: only last view
- Page 4: only last view
- Page 5: all views except the first
- Page 6

To do this, first mark the respective pages and views. You may use the check boxes in the left panel (where the current view number and page number are displayed).

Then, open a terminal in the directory where your .ipe file is located and run the following command

```
ipetoipe -pdf -markedview snappys-favorites.ipe
              snappys-handout.pdf
```

Note that if a page is marked and no view on that page is marked (which is the case if you do not touch any marks), then the default of the above command is to only export the last view of the page.

What to hand in? Hand in two files: one **pdf** file *snappys-favorite.pdf* that contains the complete presentation after completing all instructions from Section 1.1 and Section 1.2 that can be opened with Ipe, and one file *snappys-handout.pdf* that you created in Section 1.3.

2 The Story of Hyde and Edith-Viewletta

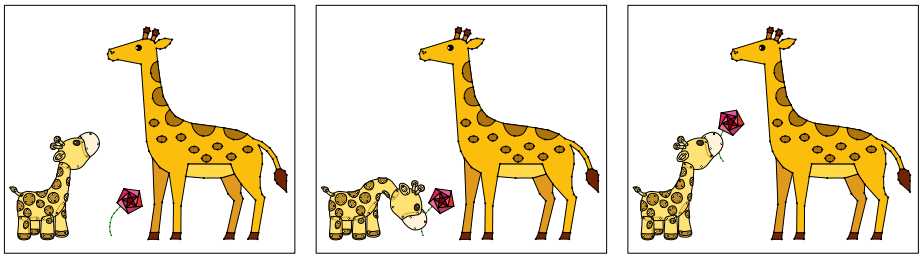
Little hyper-giraph Hyde is head-over-heels in love with the elegant giraph Edith-Viewletta. To get her attention, Hyde picked the most beautiful flower he could find[2]. Help Hyde presenting the flower to get Edith-Viewletta's attention. To do so, your task is to create a multiple view presentation as shown in Figure 2 telling their story.

In the file *giraphs.ipe*, you find all necessary building blocks to create this presentation. No worries, you do not have to draw anything yourself! Open the file and you should see all parts as labeled in Figure 3. The plan is to create separate layers for the different parts of the picture and create views while adjusting the visibility of layers accordingly. The following guide introduces two interesting techniques that you are supposed to use to achieve this task, namely the Hide Ipelet and the Edit View functionality.

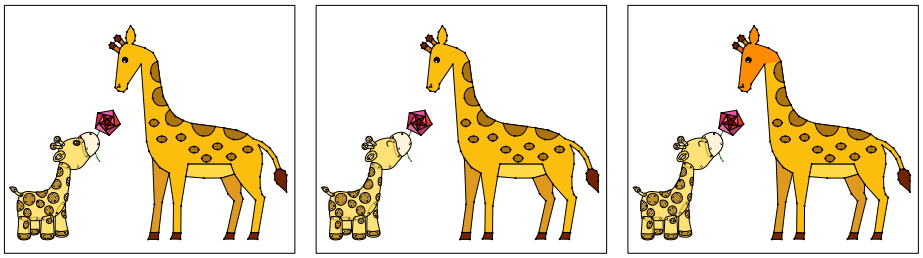
2.1 Hyde

First, we animate the hyper-giraph. We challenge you for this part to achieve everything wihout the manual creation of any new views and layers. Instead, you should

[2]It resembles the embedding of the Peterson graph where all edges have equal length!



(a) View 1 **(b)** View 2 **(c)** View 3



(d) View 4 **(e)** View 5 **(f)** View 6

Figure 2: This is what your presentation should look like in the end

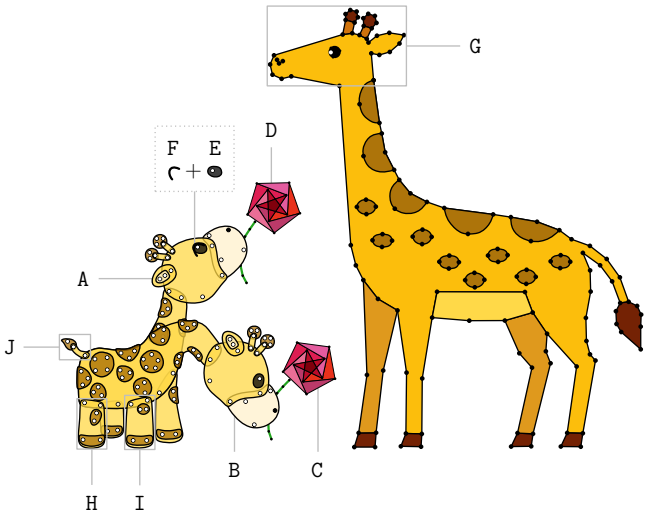


Figure 3: Parts you are given. All parts you have to interact with are labeled with a letter here.

use the Hide Ipelet. If you have not installed it yet, download it from here before you continue:

https://github.com/thobl/ipelets/tree/master/hide

and add it to your ipelets. We recommend to define shortcuts for this ipelet, e.g. by adding the following lines to *customization.lua* (feel free to choose different keys!).

```
shortcuts.ipelet_1_hide = "H" -- hide in current view
shortcuts.ipelet_2_hide = "N" -- hide in new view
```

Bending down

- There are some objects that we do not need for a while: the upper flower (D) and the smiling version of Hyde's eye (F). Select the flower and use [Ipelets → Hide → in current view] (or your shortcut) to hide it. You should observe that now the flower vanished and a new (currently invisible) layer (*alpha_hide0*) was automatically created that contains the flower.
- Repeat the same process with the similing eye and notice the creation of the new layer (*alpha_hide1*).
- We want the upright body of Hyde (A) to disappear when going to View 2. Select it and click [Ipelets → Hide → in new view]. This moves A to a separate layer named *alpha_hide2*, and also creates a new view where that layer is invisible.
- Press [PgDown] to check if the upright body indead disappears in View 2. Then go back to View 1 [PgUp].
- Now select the bent down body (B) and hide it [Ipelets → Hide → in current view], which again creates a new layer (*alpha_hide3*) for it.
  Views 1 and 2 should now look exactly like Figure 2a and Figure 2b.

Picking up the flower Create View 3 (Figure 2c) as follows:

- In View 2, select the flower on the ground (C) and hide it in a new view.
- Go to view 3 and hide the bent down body there using the Hide Ipelet. Notice that this did *not* create a new layer since the selection already had its own layer!
- Now we want the upright body and the upper flower to be visible again. Do you remember on which layers they are? You can always remind yourself of the contents of a layer by switching a layer's visibility on and off by clicking the checkmark next to the layer name, but this is rather annoying. So this is a great moment to take a step back and rename your layers to keep a better overview. Do this with [Right click on layer → Rename][3]. Choose some descriptive names such as *flower-up, smiling-eye, hyde-body-up, hyde-body-down, flower-ground.* Make the layers which contain the upright body (A) and the upper flower (D) visible in View 3 using the checkmarks in the list of layers. You should now see the same picture as in Figure 2c.

Cheese! Once Hyde gains Edith-Viewlettas attention, he wants to give her a really big smile (see Figure 2e).

- First, we have to hide the non-smiling eye (E). This is a little more tricky since it is currently part of the group that contains its entire upright body. Thus, we have to extract it from the group first: select the body (A) and open Group Edit. Select the eye and move it to layer *hyper-body-up* using [Right click → Layer

[3]You can generally also rename the active layer using [Ctrl + Shift + R], but here *alpha* is our only active layer on all views.

→ hyper-body-up]. We use this layer so that it keeps its (in-)visibility in all views. Notice that it is not part of the group anymore now; you cannot interact with it since its new layer *alpha* is locked in the Group Edit. End the Group Edit.

- Observe that the Group Edit moved the group to the front so now the eye landed behind the head and the the order of some other body parts got messed up too. Fix this by selecting the eye (E), the two relevant legs (H and I) and the tail (J) and put them in front [Ctrl + F].
- Select the eye again and hide it in a new view, which creates a fourth view.
- Go to View 4 and make the smiling eye visible that you hid in the beginning. Also move it to the front [Ctrl + F].
  This completes the part about Hyde.

2.2 Edith-Viewletta

For the following, we use a different technique: here, you are not allowed to hide, copy or paste any parts of the giraph. Instead you should achieve all tasks using only the Edit View [V] feature.[4]

Head Rotation Our plan is to use a transformation on a layer that contains only the giraph's head on all views were the head should be bent down (we are about to create this layer). The tricky part here is to find out the transformation matrix that does the job. This is not as intuitive as the usage of other Ipe features, but can be achieved using the *Edit as XML* function. Here is one way to do it:

- Go to View 3 and create a new view and a new layer [Ctrl + Shift + I] (or alternatively click [Views → New layer, new view]). This operation automatically brings us to the newly created view, where the new layer was made the active one.
- Select the giraph's head (G) and press [Ctrl + Shift + M] to move it to the newly created layer.
- Press [Ctrl + Shift + R] to rename the layer to *head*.
- Go to View 1 [Pos1] and make the layer *head* visible in all views [Right click on layer *head* → make *head* visible from view 1 onwards].
- Go back to View 4. Now take a look at the anatomy of the head in Figure 4. Then, select the head and rotate it around the point R such that the new position of point P is the old position of Q. Use snapping and a custom coordinate system to achieve this.
- Open the XML file of the head object [Right click → Edit as xml]. The first line of the xml file should now look like this:

```
<group matrix="a b c d e f">
```

  with some numbers $a, b, c, d, e, f \in \mathbb{R}$. Copy these numbers.
- Go back to your Ipe document and reverse the rotation you just did [Ctrl + Z].
- Then, press [V] to open Edit View and add the following line to "Transformations":

```
head=[a b c d e f];
```

  by pasting the numbers you just copied and click the "Ok" Button (make sure to not use spaces before or after the equality sign!). You should see a rotated head now. This is because the transformation matrix gets applied to all objects on the

[4] If you are using Ipe 7.3.1 this task might not be possible the way it is described in the following.

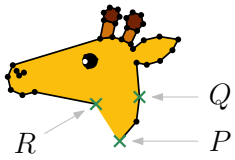


Figure 4: Head of Edith-Viewletta.

layer *head* – but only in the current view! Note that, however, Ipe still treats the object as if it had its old position, it is just displayed elsewhere; select the head to observe that it flashes up red at its original position.

- Repeat the previous step also for View 5 and View 6, which you first have to create from View 5 with [Views → New view].

Blushing Seeing Hyde's really big smile together with the flower (how did he know the Peterson graph was her favorite graph?), Edith-Viewletta starts to blush. Use the attribute map in the Edit View menu to achieve this:

- Open the stylesheet overview [Ctrl + Shift + S], select *giraphs* and click [Edit]
- Add a new color which has the same RGB values as the color *giraph-body*, and name it *giraph-head*.
- Save your stylesheet and go back to your Ipe document.
- Edit the giraph's head (G) with a Group Edit and select the outline of the yellow part. You can pick its current properties with [Q] to make sure you have selected the right part: the fill color should be *giraph-body*. Change it to *giraph-head*. The color should remain the same.
- End the Group Edit and use [Ctrl + B] on the selected head to put it back behind the body.
- Open Edit View [V] and add the following line to "Attributes":

```
color:giraph-head=darkorange;
```

Again make sure not to use spaces around the equality sign. Now the color of the head should be changed only in View 6.

Your presentation is now complete.

What to hand in? Hand in a file named *giraphs.ipe* that contains the final result after completing all steps.

Some Notes *You probably noticed that figuring out the transformation matrix for the head rotation is a little cumbersome. We do not necessarily recommend using this technique*[5]*; it would have been much more efficient to create two layers with separate copies of the heads (one layer containing the non-rotated head and one containing the rotated head) and then to adjust the visibilities of these layers.*

The attribute mapping, however, can be useful sometimes for highlighting objects in multiple-view figures without the need of having multiple copies of the same objects lying on top of each other on different layers with different colors.

[5]No wonder this feature is no longer available in Ipe 7.3.1.

3 Presentation Template

This task guides you through creating your own presentation template from scratch. There will be some fixed requirements, but you can also make a lot of design choices yourself. We want you to use this template for your final presentation in the end of the semester, so keep in mind to create something that you feel comfortable presenting in front of the group.

The goal is to create a "dark mode" design; as an example, see Figure 5. If you do not feel in the mood to make design choices yourself, you can simply recreate this design.

What to hand in? Hand in a file called *template.ipe* that contains your finished template.

3.1 Basic Stylesheet Preparation

Open a new Ipe document and open the stylesheet overview [Ctrl + Shift + S]. Delete everything except *standard.isy*. Then, create a new stylesheet with the name *my-presentation.isy*. Initialize it with the contents of *presentation.isy* that you should find in the directory where you installed ipe in the *styles* folder. Take a moment to look at this stylesheet and observe the differences to the stylesheet *basic.ipe*.

Textstretch In particular, you should notice that it includes some definitions for so-called *textstretch* attributes like the following:

```
<textstretch name="normal" value="2.8"/>
```

This makes severy text object with the size `normal` to be enlarged by a factor of 2.8. The reason to use this is that we need larger text sizes for presentations than for single-view figures and it is considered better style than redefining the textsizes, e.g. setting the textsize `normal` to the value `\Large`. For every existing symbolic attribute of the type `textsize`, you can add a `textstretch` object to a stylesheet with the same name as the textsize. Do not try to use a negative number as the value[6]; 0 is a possible value. Set the textstretch for all available textsizes to 3.

[6] If you do, make sure to save your document before that.



(a) Example layout of title page **(b)** Example layout of other pages

Figure 5: Your template should look somewhat like this in the end.

Paper Size Choose a paper size you like that is different from the predefined one, but make sure the width is at least as large as the height. Do not to change the `crop="no"` attribute. Normally, you would adjust the frame of the grid next. We leave out this step since we are going to use a black background on which you are not able to see the grid anyway[7].

3.2 Title Slide

Next up is the design of the slides and we begin with the title page.

Background For the background of the title slide, you should use a smooth transition between black and another color of your choice: for this, you need to define a gradient object in a stylesheet. The definition of a gradient can look like this:

```
<gradient name="example" type="axial" coords="96 448 288 352"
          extend="yes">
<stop offset="0" color="0.439 0.761 0.922"/> <!-- cyan -->
<stop offset="0.25" color="0.196 0.313 0.549"/> <!-- blue -->
<stop offset="0.9" color="0.635 0.133 0.137"/> <!-- red -->
<stop offset="1" color="0.875 0.608 0.102"/> <!-- orange -->
</gradient>
```

See Figure 6 to see how a box which is assigned this gradient looks like. For creating an axial gradient[8], we need an axis defined by two points. In the above example, these are the points $(96, 448)$ and $(288, 352)$ specified in the `coords` field. The direction of the line segment between these points is the direction of the color transition, i.e. every perpendicular bisector of this line has the same color in all its points. If you set `extend="no"` the gradient will be cut off at the two perpendicular bisectors of the two end points.

Additionally, we need to specify two or more colors by adding a `stop` object for each color. It consists of the decimal rgb values and the `offset`, which determines the proportion of the line segment after which there should be exactly the specified color; for instance, an offset of 0.25 for the second color (blue) here means that in the first quarter of the line segment, the color transitions from color 1 (cyan) to color 2 (blue) and from there starts to transition to color 3 and so on. When using only two colors, you just need the offsets 0 and 1.

Add your own gradient definition to a stylesheet that consists of two colors and a direction of your choice. One of the colors should be black and the other should be a different color of your choice. We recommend using a rather dark color so that the transition is not too strong and distracting. You can choose the endpoints of your axis freely, but we recommend not choosing the points too close together to make the transition smooth. Then, add a box [B] of type *fill only* to page 1 of your document which has the same size and position as the page. Assign the gradient [Right click → Gradient] to this box.

[7] It can still be useful to define the grid: e.g. the shortcut [F10] creates a paragraph object that has a fixed width that equals the grid width. So if you want to use this, it makes sense to have a grid that is horizontally centered at the page while leaving some space to the edges of the page.

[8] There are also radial gradients which we do not cover here.

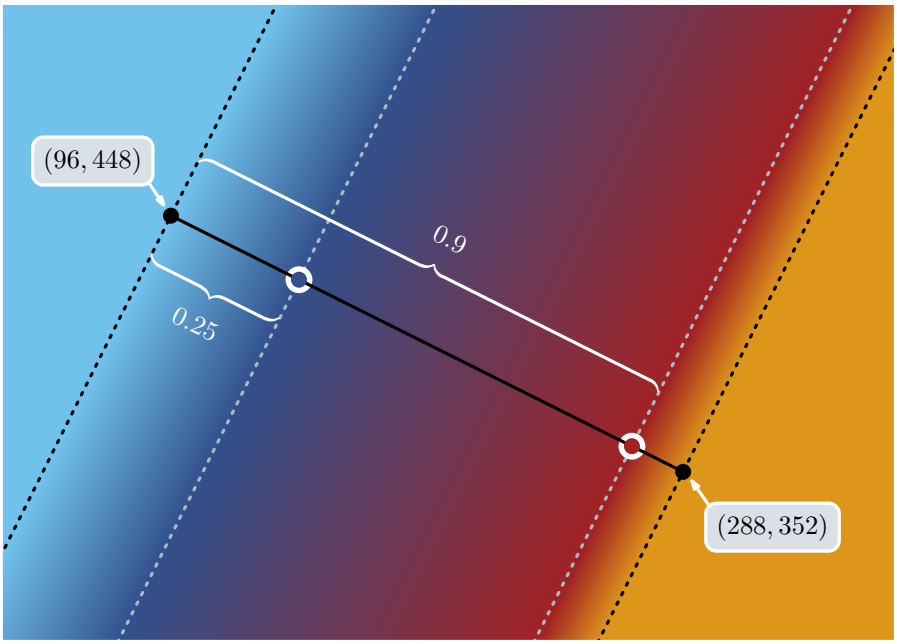


Figure 6: Values that are needed to define a gradient. The numbers 0.25 and 0.9 are an example for the `offset` values of the second and third color; the offset 0.25 for the dark blue color means that the blue point lies at one quarter of the length of the line segment. The color changes only betweeen the dashed lines.

Note that transforming an object that has a gradient assigned can lead to somewhat surprising or confusing behavior. This is because for the effect of an assigned gradient, only the original path coordinates are considered (without application of the transformation matrix), which are generally not the same as the coordinates where the object is actually displayed[9]. That is, whenever you rotate [R], translate [T], or scale [E] an object with an assigned gradient, the gradient colors get rotated/translated/scaled as well since these transformations only change the transformation matrix. However, when you use the Edit Path tool [Ctrl + E] to transform an object, this changes the actual path coordinates and also resets the transformation matrix so the gradient colors do not move along, but are newly calculated from the new coordinates.

Placing Elements The title slide should also include a title, a subtitle and an author name. Since we will need title and author for the footer of later pages as well, define commands for them in the preamble. For that, open the document properties [Ctrl + Shift + P] and add the following lines to your preamble:

```
\renewcommand{\title}{Presentation Title}
\renewcommand{\author}{Author Name}
```

This makes it easy for the user of your template to set both properties quickly without touching the stylesheets. Then, add wide paragraph objects (titles and names can be long!) for both of them containing the commands, and place them wherever

[9] You can check the actual path coordinates and the transformation matrix in the xml of an object [X].

you think they loook nice. Set their textsizes, alignments, and stroke colors such that the text is clearly visible. If you want your text centered, set the paragraph styles to *center* in the drop down menu in the respective paragraph edit windows [Ctrl + E]. Create another paragraph object that should provide the possibility to enter a subtitle. You do not need to define a command for this.

If you want, you can add other graphical elements like separation lines or a picture or logo (see Figure 5a for a possible design), but this is optional.

Rename the layer with all your elements exactly to BACKGROUND. This will suppress the background symbol we will create for later pages from being displayed on this page. To ~~make sure that nobody makes changes~~ make it harder for people to change the positions, sizes, or colors of your elements, lock the layer [Right click on layer → Lock] you worked on. To do this, you need to create an additional layer and make it the active layer first. Move the paragraph for the subtitle to this layer [Ctrl + Shift + M] as it should be easily editable later and should not be part of the background.

3.3 Background for Later Pages

The next goal is to create a background for your presentation that is the same on every page (except the title page) and not editable by the user[10]. This is done by creating a symbol with the name Background. Note that the first letter has to be uppercase and the remaining letter lowercase. If your stylesheets contain a symbol with this name, it will be displayed on every page that does not contain a layer with the name BACKGROUND.

Design a background you like, just make sure it meets these requirements:

- In the very back, it contains a rectangle that fills the whole size of the paper and is filled with the color black.
- It contains a footer that is clearly separated from the main body, e.g. by using a separating line segment or a filled area of a different color.
- The footer has a paragraph object containing title and author (in case the audience was sleeping during the introduction). Use the commands you defined in the preamble.
- The text in the footer should be vertically centered within the footer area.
- There should be some space in the footer that is left free for the page numbers we add later.

Afer creating everything you want, select all and make it one single group. Select the group, use the Symbols Ipelet to create a new symbol [Ipelets → Symbols → create symbol (in top stylesheet)] and name it Background. Then, delete the group (or save it somewhere else if you want to make adjustments later). You should now see that your symbol got automatically added to the background; you cannot select the objects of the symbol anymore or hide any new objects behind them.

3.4 Stylesheet Refinements

Now that we have our background, it is time to make some more edits to our stylesheets to define the properties of page titles, page numbers, bullet points and additional options. You can make all of the following edits in the stylesheet *my-presentation.isy* from the beginning.

[10] at least not without editing the styelsheets

Page Titles Scroll down in *my-presentation.isy* until you find the following line:

```
<titlestyle pos="608 674" size="large" color="title"
            halign="center" valign="baseline"/>
```

Adjust these properties to your likings such that the page title design matches the background you created. Recall that you add page titles with [Ctrl + P].

Page Numbers To add page numbers to your slides, you first have to open your document properties [Ctrl + Shift + P] and check the option [Number pages]. Since we used a black background, we probably won't be able to see anything yet since the default page numbers are black. You can customize their color, position, alignment, size, etc in your stylesheet. A definition can look like this:

```
<pagenumberstyle pos="0 -10" color="black" size="small"
                 valign="baseline">
  \ipeNumber{\arabic{ipePage}}{\arabic{ipePage}}
</pagenumberstyle>
```

The command `\ipeNumber{}{}` returns the first argument if the page has only one view and the second argument if the page has multiple views. In the example above, both arguments are the same. For instance, this can be adjusted to print both page number and view number if there are multiple views (which is the built-in default). The following Latex counters can be used here:

- `ipePage`: current page number
- `ipeView`: current view number,
- `ipePages`: total number of pages,
- `ipeViews`: number of views of the current page.

However, we do not recommend printing view numbers since this can unnecessarily clutter your slides, especially if you have many views.

Include page numbers in your template. You should position them such that they are vertically aligned in your footer. Other than that, you can choose a design and format that you like.

Bullet Points Adjust how bullet points should look like in your presentation. For example, you could color them in some variant of the second color you used in your gradient. As this is just a standard Latex customization, you can add it to your preamble, i.e. inside a `<preamble> </preamble>` tag in your stylesheet.

Additional Styles Add some more stylistic options to your stylesheets. This should include some custom colors[11] and some decorations of your likings. Optionally, you can include custom mark types, arrowtips, textsizes[12], etc. Feel free to include stylesheets from your work on the previous sheet.

[11] Since we use a black background, it is probably nice to have some darker colors as well.
[12] Remember to also define textstretches for custom textsizes.

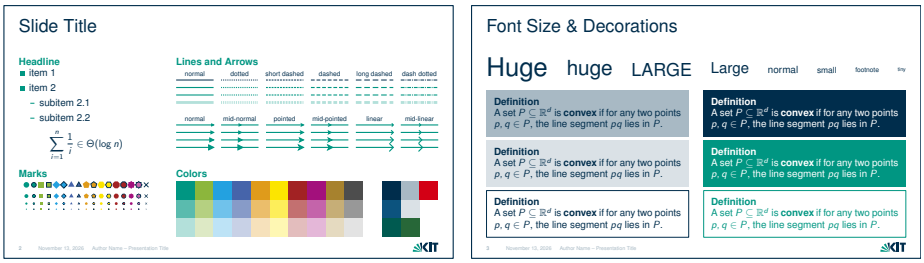


Figure 7: This is more or less how you should demonstrate the available styles.

3.5 Styles Demonstration

Starting from page 2 of your template, demonstrate what styles there are available so that someone who is new to your template[13] gets a quick overview of the available options. This should look more or less like in Figure 7 (example from our KIT template).

4 Use What You Have Learned!

In this task, you should practice creating multiple-view-figures similar to the ones given in the folder *multiple-views-examples*. For this, create a presentation *multiple-views.ipe* where each completed subtask has its own page. The page titles should be the same as the task titles (Sections 4.1, 4.2, 4.3, and 4.4). You can use either your presentation template from Section 3[14] or the KIT template.

What to hand in? A file *multiple-views.ipe* containing one page for each task.

More Useful Tools There are some tools we have not presented yet that might be useful for you in the following:

- Select all elements in one layer: make the layer active, then [Ctrl + Shift + A].
- Select all elements with a specific property (e.g. fill color, line width, mark type): first, select the property you want (either in the drop down menus on the left or by picking from a specfic object [Q]). Then, press [Shift + S], mark all properties you want to match, and press the "Ok" button.
- With the Polyfillet Ipelet, you can round the corners of a polygon by a radius of your choice.

4.1 Solitaire

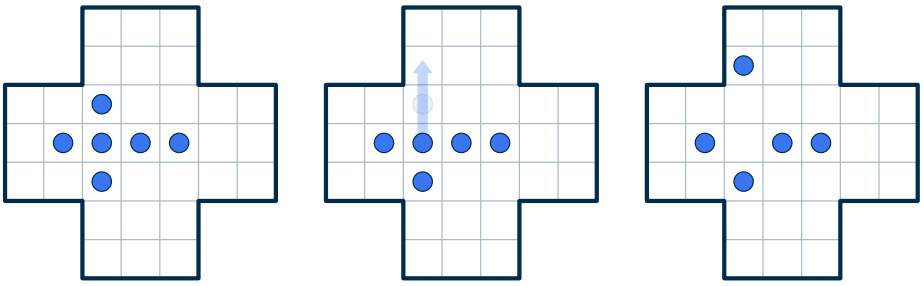
Create a solitaire[15] board as shown in Figure 8a. Draw six tokens on it in your favorite color – you can use the configuration from the example or your own, as long as it has at least six tokens and is solvable. Then, demonstrate how to clear the board by adding two new views for each move: one view that shows which jump is going to happen (cf. Figure 8b) and then one view where the move has been performed (cf. Figure 8c). Make sure of the following:

[13] or someone who checks your submission.

[14] If you use your own template, it probably makes sense to use white or other bright colors instead of black for outlines, edges, etc. Make sure to copy the template before editing it.

[15] See https://en.wikipedia.org/wiki/Peg_solitaire for the rules.

- Each token is center aligned with its square.
- Each arrow is transparent and has rounded corners (radius 1pt).



(a) (b) (c)

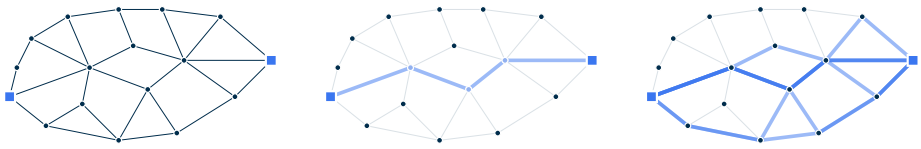
Figure 8: First three views of the Solitaire page

4.2 *k* Shortest Paths

Draw a graph with two highlighted vertices, using the mark types from Figure 9a. You can use your favorite graph under the condition that it has at least seven different paths between the two vertices you mark. Then, highlight the seven shortest paths between the two marked vertices one at a time(each path has its own view), as in Figure 9b. The highlights should satisfy the following:

- The highlighted path edges are thicker than in View 1 and have a different color.
- No other edge obejcts are visible beneath/above the highlighted edges.
- The inner vertices along the paths have the same color as the path edges.
- All non-path edges are thinner than in View 1 and have a light gray color (or dark gray if you use your dark mode template).

The last view shows all seven shortest paths together in such a way that edges that are used in more paths appear darker (brighter if in dark mode template) than edges that lie on fewer paths (cf. Figure 9c).



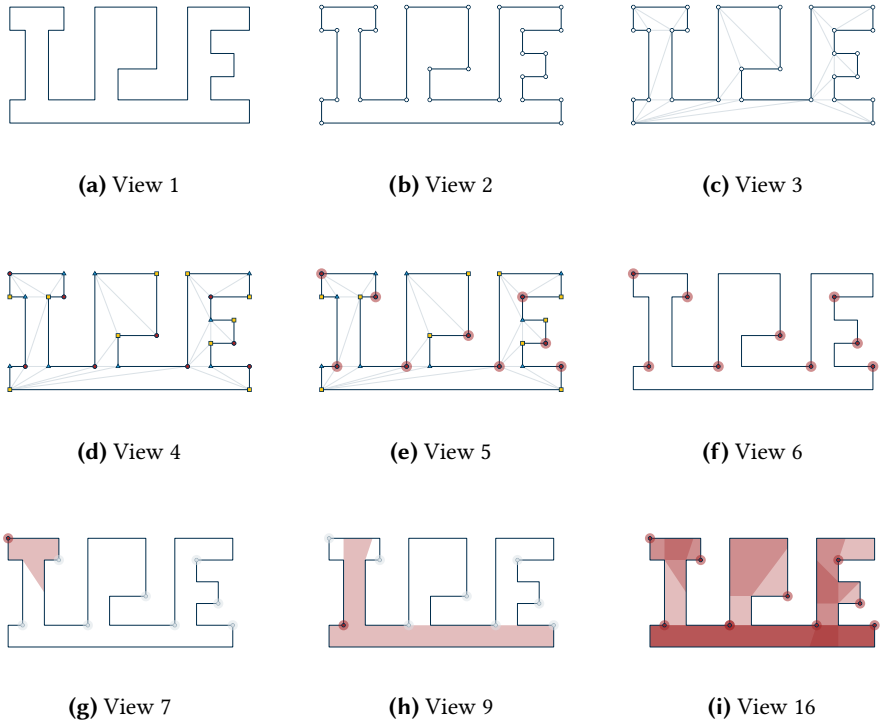
(a) (b) (c)

Figure 9: Parts of the Shortest Paths page

4.3 Art Gallery

We define the Art Gallery problem as follows: given a simple[16] polygon (the "gallery"), what is the minimum number of points ("guards") that together observe the whole polygon? A guard *observes* a point P in a polygon if there is a straight line between P and the guard that lies completely within the polygon. The goal of this task is

[16] no holes or self-crossing edges



(a) View 1 **(b)** View 2 **(c)** View 3

(d) View 4 **(e)** View 5 **(f)** View 6

(g) View 7 **(h)** View 9 **(i)** View 16

Figure 10: Parts of the Art Gallery page

to create a multiple-view-figure that visualizes how to derive that $\lfloor n/3 \rfloor$ guards are always enough to observe a simple polygon with n vertices. No prior knowledge or understanding of this proof is needed to complete this task; we explain below what you should visualize.

Polygon Choose a polygon that is complex enough in the following sense:

- It should have at least 15 vertices.
- It should be simple, i.e. have no holes and no crossing edges.
- It should be possible to position three guards on three different corners of the polygon such that their three observed regions are pairwise disjoint.
- You can use the polygon from Figure 10a but it is probably more interesting to choose your own.

Content of the Views The page for this task should be structured as follows:

- View 1 shows only the polygon (cf. Figure 10a).
- View 2 adds marks on all corners of the polygon (cf. Figure 10b).
- View 3 adds a triangulation of the polygon. The lines of the triangulation should be gray. You can use any triangulation (cf. Figure 10c).
- View 4 adds a 3-coloring[17] to the vertices (cf. Figure 10d).

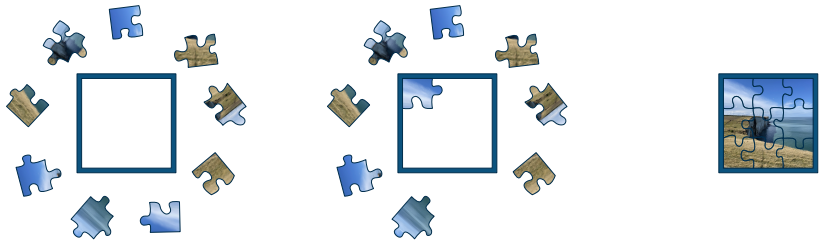
[17] That means each vertex should have one of three available colors and adjacent vertices cannot have the same color.

- View 5 highlights all vertices of one color with the fewest vertices – these represents the guards that are going to be used (cf. Figure 10e).
- View 6 hides all non-highlighted vertices and all triangulation lines (cf. Figure 10f).
- View 7-15 display the (exact!) fields of vision for every guard, respectively, as transparent polygons. In each view the highlight of the selected guard remains, while the highlights of the remaining guards should be faded out (cf. Figures 10g and 10h)
- View 16 shows the observed area of all guards together; the more guards observe a point of the polygon, the darker (brighter if dark mode) this point should appear (cf. Figure 10i).
- Bonus: Add an additional view that shows a solution with the fewest possible amount of guards together with their observed area, in the same manner as in View 16.

4.4 Jigsaw Puzzle

Import an image of your choice[18] and create a quadratic jigsaw puzzle out of it (cf. Figure 11). Your page for this task should satisfy the following:

- The puzzle should consist of exactly nine pieces.
- In the completed puzzle, the corners of the pieces should perfectly align with the corners of a grid of the frame that consists of 3x3 squares.
- The sides of the pieces that do not touch the frame should be splines with at least six different control points; try to make it look like standard jigsaw puzzle pieces.
- There is an exactly quadratic frame that is empty in the beginning. You can choose the color of the frame freely.
- Initially, all pieces are distributed evenly along an invisible circle around the frame in any order (cf. Figure 11a). The center of the circle should be the center of the frame. One of the pieces should be centered at the topmost point of the circle (so horizontally center-aligned with the frame)
- Then, there should be one view for each piece where it moves from the side to its correct position as in Figure 11b. It should remain there for the remaining views.
- The last view shows the completed puzzle (cf. Figure 11c).



(a) View 1 **(b)** View 2 **(c)** View 10

Figure 11: Parts of the Jigsaw page

[18] Cite the picture appropriately using a clickable link if you take a picture from the internet.