## OCTA: overview changes

* Update shapes: Image, FitImage, Path, PathSvg, Text, Polygon, RegularPolygon:
  + parameters given as argument to shape name
  + in code add info to data parameter (to make it work when using json file as input for svg creation) 🡪 solves issues with json load and json save, but now data argument is necessary for swaps when different polygons present eg
* Change in Image shape: image info is saved within output svg (so saving external image file separately is not strictly necessary; and option to create Image shape from url)
* Positions.CreateCustomPositions(x,y)
* correct name of mirror\_value parameter in octa.shapes
* add class and id labels to svg outputs in octa.shapes
* add possibility for gradient for fillcolor and bordercolor to octa.shapes
* PNG generation now using html2image library (works for more generated svgs, also those including gradients etc)
* include opacity as element parameter
* add possibility for background\_shape (clipPath)
* add possibility for masks
  + only works in full SVG profile instead of tiny profile
* update Text shape to work in full SVG profile

<https://www.magicpattern.design/>

* Image and FitImage as different shapes?
* how to add relativejitter consistently?
* how to implement animation? Working example but not sure what best way is to actually implement it?
  + As additional parameters to shapes?
  + As separate animate module?
  + …?
  + Animation of elements vs animation of display as a whole vs animation of positions of elements in whole?
* Better to create javascript app than shiny app?
* Better to create sphinx documentation than R documentation? Both?
* Sphinx readthedocs documentation?
* Learnr tutorial?
* waarom zou het beter zijn indien aantal rijen en kolommen niet flexibel zou zijn??? wat zou dat makkelijker maken? kan er niet gewoon foutmelding komen indien patroon niet mogelijk voor dat aantal rijen/kolommen?

TO DO?

* Add Gert functionality?
* Only show elements within certain shape? 🡪 fill with pattern object?
* <https://stackoverflow.com/questions/46515355/best-way-to-fill-a-svg-shape-with-other-elements-without-clipping-them-at-bounda>
* <https://stackoverflow.com/questions/61718701/how-to-get-all-points-contained-within-an-svg-path-string-without-checking-every>
* <http://gestaltrevision.be/GERT/GERT_v130_manual.pdf>
* <https://css-tricks.com/snippets/svg/svg-patterns/>
* Only place elements on border of a shape?
* Spiral pattern?
* Setting the bounding box does not take into account border linewidth -> Add disclaimer that for stimuli to be the same size, the borderwidth also has to be the same
* Outputfiles & orde- en complexiteitsmaten
* Documentatie
* Combine multiple svg stimuli
* allow parameters to be updated for specific ids/categories
* Implement a directional component (default option of left->right and top->bottom is now assumed) ; direction parameter for patterns
* Allow gradients to be generated by providing a step size and a starting value.
* Implement non-linear gradients (e.g. exponential increase/decrease of parameter values)
* eventueel extra jitter/gradientopties (increasing, decreasing, op deel van elementen) en deviantopties (add\_element, remove\_element, switch specific element, replace specific element, etc.)
* Get path from svg better (especially positioning and size) 🡪 to be able to change color svg shapes etc
* Add example classes & ids use
* Create regulartriangle vs triangle? 🡪 or rename to CustomTriangle vs Triangle?
* Change in shapes Polygon/Text/Images/Paths
* Create custom positions
* Topfuncties voor complexere patronen?
* LOC en LOCI for layered grid?
* Meerdere afbeeldingen in 1 bestand?
* Patterntype en orientation for more complex patterns
* Mirroring elements: global & local symmetry?
* Rounded shapes
* TRELLO comments

Hyperlinks?

## ADDITIONAL PATTERNS / PATTERN FEATURES

## do

* make clear in documentation that top->bottom left-> right is standard in OCTA
* add in documentation:
  + replace value (of one feature of an element): arguments = element\_id and new\_value (feature\_dimension could be an additional argument OR could be part of the name of the function, like replace\_color)
  + transparent background color? (also png?)
  + image: externe png/jpg inladen
* **Different types of order (taxonomy) on different axes (row-based, column-based, leftrightdiagonal, rightleftdiagonal) – row/col zigzag, iterate, & row+colsymmetry, translation to do**

### think about

* naming RepeaterPattern / Duplicate functions: Duplicate/Repeat/Replicate/Iterate/Alternate/...?
* clarify use of the word "pattern":
  + set of values to create ordering with (pattern argument)
  + resulting ordering (RepeaterPattern, SymmetryPattern, etc.)
* structural change: all grid patterns in same file:
  + add the possibility to switch between patterns e.g.: stimulus.colors.pattern = RepeatAcrossColumns (instead of eg MirrorAcrossRows)
* extend function for BasicPattern: BasicPattern([1,2,3]) and BasicPattern([4,5,6]) --> BasicPattern([1,2,3,4,5,6]) bijv.: pattern1 = BasicPattern([1,2,3]) pattern2 = BasicPattern([4,5,6]) BasicPattern([pattern1, pattern2])

of bijv.: pattern1.extend([4,5,6])

### very nice

* additional jitter options:
  + regularly increasing jitter
  + regularly decreasing jitter
* in/out and out/in as additional dimensions (like row/column/leftdiagonal/rightdiagonal) thus: repeater/gradient/symmetry patterns along this dimension
* extra deviant options:
  + replace\_element (& all features of element): arguments = element\_id and all new values (for different feature dimensions)
  + switch\_value (like implemented, but not random switch)
  + switch\_element (like implemented, but not random switch)
  + add\_element
  + [add\_value (only if not all features need to be given and defaults are implemented)]

### nice

* RandomizeAcrossRows / Columns / ... ? (ie randomizeorder but not across whole stimulus; only switch row / column /leftdiag / rightdiag / inout/ outin values)
* add other directions than top->bottom left-> right
* additional jitter options: jitter not on all elements
* extra gradient options?
* better color gradients?
* extra deviant options: transform one, some, or all elements (e.g. change on condition, e.g. = rectangle)
* [let user give structure and apply it to given values? eg alternate (abcb)]

## ADDITIONAL STIMULUS FEATURE DIMENSIONS

### nice

* [animate ? (changing attributes across time: moving elements, changing orientation / color / ...)]
* Output measures LAB space for colours? (https://fairyonice.github.io/Color-space-defenitions-in-python-RGB-and-LAB.html)
* Other color package for gradients (more scientifically based)
* **Animate module (see also** [https://svgwrite.readthedocs.io/en/latest/classes/animate.html#](https://svgwrite.readthedocs.io/en/latest/classes/animate.html)<https://theartificial.com/blog/2018/05/23/svg-animation.html>)
  + **How long each frame?**
  + **Jitter building coherent/incoherent**
  + **From first / from previous display**
  + [**https://css-tricks.com/guide-svg-animations-smil/**](https://css-tricks.com/guide-svg-animations-smil/)
  + **attributes** [**https://slides.com/sarasoueidan/styling-animating-svgs-with-css#/10**](https://slides.com/sarasoueidan/styling-animating-svgs-with-css#/10)
  + **shape tweening (morphing shapes):** [**https://codepen.io/noahblon/post/an-intro-to-svg-animation-with-smil**](https://codepen.io/noahblon/post/an-intro-to-svg-animation-with-smil)

## ADDITIONAL POSITIONS / POSITION FUNCTIONS

### important

* Circle: bug in # elements on circle --> Eline implemented quick fix but please check :-)
* **Larger than gridplace: not structurally implemented, but possible**
* **Different templates: some implemented (to do: concentric circles, all elements within circle form, …)**

### very nice

* jitter relative to bounding box function
* patterns in positions: different distances in 1 stimulus (eg regularly increasing x/y distance, symmetric distances, ...)
* distances including shapes vs between shapes
* change positions after initialization: eg. stimulus.positions + 20
* contour only options (eg for grids: only elements on outline) as well as filled options for closed shapes (eg for circle: circle filled with elements)
* Spiral
* different / custom overall shape = specific boundary of the complete pattern (ideally same options as for element shapes) --> look at GERT?

### nice

* [masks = pattern going to infinity and cut (ideally same options as for element shapes)] --> look at GERT?
* [dot lattices as one set of grids?]
* [more curved lines, eg. literature dot lattices Kubovy?]
* [zigzag lattices Peter Claessens]
* [diffeomorphic transformations]

## ADDITIONAL STIMULUS FUNCTIONS

### nice

* change orientation of overall pattern?
* Mask?
* Cut in shape

## SHAPES

CRUCIAL SHAPES: ellipse, rectangle, triangle, polygon, none, image, text, path, (rounded\_rectangle, rounded\_triangle), (curve)

### important

* warning PathSvg en Text: experimental features
* scaling images working?
* **Svgpathtools** <https://github.com/mathandy/svgpathtools>
* triangle:
  + check function used (now Eline's but can probably be improved)
  + also not equilateral triangles
* text:
  + center text in bounding box
* [curve:]
  + border color same as fill color

### very nice

* image:
  + externe svg code embedden/inladen
* converter module bepaling grootte (different parameters possible at initialization --> bounding box params)
* [rounded\_rectangle, rounded\_triangle, rounded\_polygon]
* [spiral]
* [newshape]
  + ook externe code inladen en kleur aanpassen?
* text:
  + different font types [& sizes]?

## INPUT

OCTA code (= now)

important:

* OCTA json file (implemented now but will need to changed/adapted together with outputfiles and additional functionality toolbox) [NEEDS DEBUGGING FOR SHAPES Polygon, RegularPolygon, Text, Image, FitImage, Path, PathSvg]

### [nice ?]

* [OCTA text file: only element info?]

## OUTPUT

### important

* inhoud outputfiles verder aanvullen
* json file + json inline
* svg file + svg inline

### very nice

* order & complexity measures (of elements & position = location & orientation)
* [txt file + txt inline]
* png file + png inline / jpg file + jpg inline

### nice

* [OCTA code]

## USER INTERFACE [not part of the toolbox itself]

### important

* Shiny (?) app, on more stable server (paid?):
  + also output OCTA code to generate same stimulus in python
* or Python?
* **Use nbinteract, public github and mybinder.org to share notebook online**
* **Notebook: only save on button click**
* OCTA API?
* Sphinx documentation? readthedocs
* documentation (gitbook via Rmd); look at documentation GERT
  + AND automatic Python documentation

### very nice

* tutorial (learnr tutorial via Rmd)

### nice

* create elements > draggable elements in javascript

## GENERAL ISSUES

### important

* problem with image and text stimuli:
  + can not be switched
  + complexity calculation

--> can be switched and complexity calculation solved now?

* output files verder aanvullen

## EXTRA'S

### very nice

* **in general: look at consistency options on element and stimulus level**

### [ nice ? ]

* svg.js (<https://svgjs.com/docs/3.0/>)

## OCTA 2.0: adding on basic functionality

### very nice

* **Function to create grid of different stimuli varying on parameter – not function but for loop for now**
* generate range of stimuli
  + e.g. give values for each feature dimension + dataframe with different order combinations + names stimuli --> create several svgs
* when generating range of stimuli: html image grid as output option = create\_grid function to create combination of several stimuli varying in order & complexity dimensions
* combine different displays / stimuli / elementgroups (e.g. 2 concentric circles with elements); group elements together? extend\_stimuli

merge\_svg function: does viewbox change as well?

* generate shape with certain level of order / complexity?

### nice

* plot different elements (with different sizes) on top of each other?
* integration online svg database/repository (inject svg + change color):
  + svgrepo (<https://www.svgrepo.com/>)
  + publicdomainvectors (<https://publicdomainvectors.org/en/tag/svg>)

## OCTA applications

* **Also use for grouping: similarity judgments?**