**Notes on Red Animation DSL**

Glossary:

Tween

Easing

Timeline

Frame

Stagger – an offset in time in the transformation of the items in a group (array) ([https://vimeo.com/371944028 at 9:00](https://vimeo.com/371944028%20at%209:00) - Getting Started with GSAP)

Yoyo – animation is played repeatedly from start to end and then backwards.

**Structure**

Dispatcher->Timeline->Tween->Easing

**Main Features:**

**Easings**

Linear, Sine, Quad, Cubic, Quart, Quint, Expo, Circ, Back, Elastic, Bounce, Steps (with a parameter N)

**Tweening**

* The change of target parameters for each tween can be specified in native form (integer, float, pair, color, tuple) or as percentage of the starting parameter.
* **from** and **to** parameters can be numeric values (or pairs / tuples) or **functions** of such values, for example **random**!

Tween works on float values. How to use it for other types?

* Integer! – casting
* Percent! – casting
* Pair! – must be called for both x and y components after casting to a float value.
* Tuple! - must be called for each components after casting to a float value.

**Timelines**

* Offset for animations that play simultaneously:  
  The alignment could be measured relatively to the start or to the end of the previous (in the description block) animation, in seconds or as percentage.

Follow a path

* **Actions:**- play- stop ; premature – stopped by the user   
  - pause  
  - resume  
  - reverse  
  - step  
  - slow/fast
* Actors

- on-start  
- on-exit  
- on-stop  
- on-pause  
- on-resume

Stagger can be not just a number, but also a function of the target’s index (stagger has meaning for collections of object only)

**Additional Features:**

* In/Out effects – Fade, Fly, Interlaced stripes…
* Highlighting of text
* 2D Arrays – rectangular and polar  
  Each array item could be a target for animation; the creation (placement) should be able to be animated – the axes (rectangular: x and y; polar: radius and angle)
* Stagger for arrays controlled by axis (x and y) or radius / angle (for polar ones)
* Physics – collision detection, gravitation, wind, friction
* Particle systems
* Text effects on characters, words and lines  
  - Scramble (https://greensock.com/docs/v3/Plugins/ScrambleTextPlugin)  
  - Random words (https://codepen.io/GreenSock/pen/waCoe)
* Layers
* Parallax effect (with layers)
* Motion blur
* 2D Morphing for polygonal shapes
* IK

**Particle systems**

**Shape** – how does a single particle look like. A draw primitive (or a set of primitives, a list of characters and so on)

**Emitter** – where the particles are born.

* Point
* Region
* Curve

**Number of particles** – how many particles in total

**Delay** (stagger) – the time (or frames) between the birth of particles

**Dynamics** – how particles move. Can be

* an end coordinate, reached by straight line
* a function of time
* governed by physics – gravitation, wind, air drag, friction

**Birth** – how does a particle come to life – fade in, grow…

**Demise** – how does a particle’s life end – fade out, peter out…

(There might be more stages in the particle’s life, like peak)

**Ideas for tests:**

* Pythagorean theorem
* Коляно-мотовилков механизъм (бутало и цилиндър на ДВГ)
* Sci-fi movie like Interface (https://codepen.io/shunyadezain/pen/GRNEyZW)