argument(s) – *math*

Q: What are the mathematics requirements for *creative technology*?

A: This depends, even from a *new media* perspective, whether the focus is on animation, information visualisation, game development or even more general on an artful application of mathematics in for example web design or interactive installation(s).

Nevertheless, to some extent, at least on an introductory level, topics that should be addressed are [1]:

- points, vectors, planes & isometries
- linear (matrix & vector) algebra
- combinatorial and numerical algorithms

It seems worthwhile to proceed from isometries in euclidean geometry, to transformations in linear algebra, via an introduction of coordinate systems.

Q: And more specifically for animation and visualisation?

A: In summary:

- graph theory
- basic calculus, newtonian physics
- particle systems in 2D space

It must be emphasized that students will generally start from an intuitive idea of a phenomenon, for which they should be able to find a more or less adequate formalism or model that embodies this intuition. In practice this might mean that they merely adapt an existing animation or visalization to fit their needs.

Q: Any special wishes for game development?

A: Yes, in particular the mathematics needed for *game physics*, *dynamic lighting* and *visual effects* [2]:

- game physics – motion, speed, collision detection
- dynamics lighting – normal, reflections & refractions
- natural phenomena – waves, perturbations, explosions

Evidently, these may be considered a generalization of the issues treated for animation and information visualisation, extended to 3D space. However, the natural phenomena are more complex, and the computational requirements may be considered to be significantly higher.

Q: What about the (he)art of mathematics?

A: There are nowadays many tools, including the *processing.org*[1] software, to produce *generative art*, or more general visual patterns. Artists speak of such methods as using *algorithms as a paintbrush*. However, where the methods are often used in a naive way, is seems desirably to create algorithms that are to some extend well-founded in mathematical theory.

Topics from mathematics that provide a useful background include [1]:

- basic topology, iterated functions
- quaternions & rotations
- fractals, mandelbrot & julia sets
- lindenmayer systems

Q: Would that be all?

A: Very likely, most students will find the topics treated thus far already quite difficult. However if we look at the area of *visual computing*, which issues such as image processing, triangulation and rasterization, additional topics need to be covered, such as [3,4]:

[1] processing.org

- delaunay triangulation, voronoi diagrams
- convolution filters
- and what else?

In addition, some students may even express an interest for *geometrical algebra* [5]. Apart from being somewhat offstream, it may also be observed that these topics require perhaps a more computational than mathematical approach.

statement(s)

line(s) – theme(s) / coffee / grind(s) / thesis / _ .. / ... / / theme(s) / create / common(s) / art(s) / space(s) / _ / .

a mathematician is a device for turning coffee into theorems

/ Erdos / idea(s) / joke(s) / perspective(s) ?
/ science(s) / role(s) / digital(s) / flow(s) / mechanic(s)

math(s) – resource(s) / perspective(s) / method(s) / graph(s) _ .. / ... / common(s) / idea(s) / change(s) / topic(s) / _ / .

scenario(s) / a (wide) variety of perspective(s)

tool(s) / classic(s) / statement(s) / opinion(s)

- we eat problems for breakfast [VU/olympiade]
- we study pattern(s) & structure(s) [UvA/mathematics]
- we grind ... coffee / game(s) for ... boy(s)/girl(s)?
- we ... love story ...? UT/.CREATE

/ flow(s) / dynamic(s) / particle(s) / game(s) / screen(s) / cycle(s)

math(s) – media / perspective(s) / resource(s) / theme(s) / _ .. / ... / common(s) / idea(s) / change(s) / topic(s) / _ / .

- (casual) game(s) – toss paper(s) / crayon physic(s) / color(s)
- graphic effect(s) – tunnel / example(s) / convolution(s)
- animation(s) – blob(s) / example(s) / eye(s) / particle(s)
- student tutorial(s) – physics / linear math / animation(s)
- physics engine(s) – PhysX / unity example(s) / resource(s)
- sound synthesis – basic(s) / fm tool(s) / music pad / drum box
- generative art(s) – nano webber / stripe(s) / bulb(s)

/ requirement(s) / formula(s) / game(s) / resource(s) / X / cycle(s)

reference(s) – math

1. Mathematics of Digital Images: Creation, Compression, Restoration, Recognition (Hardcover) by S. G. Hoggar – (amazon)
2. Mathematics for 3D Game Programming and Computer Graphics, Second Edition (Game Development Series) (Hardcover) by Eric Lengyel – (amazon)
3. Visual Computing: Geometry, Graphics, And Vision by Frank Nielsen – (amazon)
4. Geometry and Topology for Mesh Generation (Cambridge Monographs on Applied and Computational Mathematics) (Paperback) by Herbert Edelsbrunner – amazon)
5. Geometric Algebra for Computer Science – An Object-Oriented Approach by Leo Dorst, Daniel Fontijne, Stephen Mann – (online)