MSc Computer Games and Entertainment

Maths & Graphics Unit 2011/12

Lecturer: Gareth Edwards

Procedural Programming

(including Algorithmic Animation)

A Very, Very, Big Topic – indeed!

Procedural Imaging/Animation/Modelling/etc.....

- Every aspect of image synthesis – using a computer to make pictures – has aspects which naturally lend themselves to procedural programming

- From simple parametric procedural surfaces of rotation to REALLY complex parametric procedural surfaces such as waves

- But there is a catch...

Clever Stuff

Procedural == Algorithmic == Programming

- It is not for every one
- It is for those for whom programming is a natural, creative activity
- It is invariably clever stuff!

Procedural Programmers

Must be like a watchmaker

- Able to enjoy the mechanism as end in itself....
- Be an architect of change
- For whom time (and numbers, colours, texture, etc., in time) have a touch, taste and even an individual smell ....
- It is a passion embodied in an Algorithm

Procedural Programming

- Can sometimes be used as a synonym for imperative programming (specifying the steps the program must take to reach the desired state)

- Can also refer to a programming paradigm, derived from structured programming, based upon the concept of the procedure call

Procedure Call

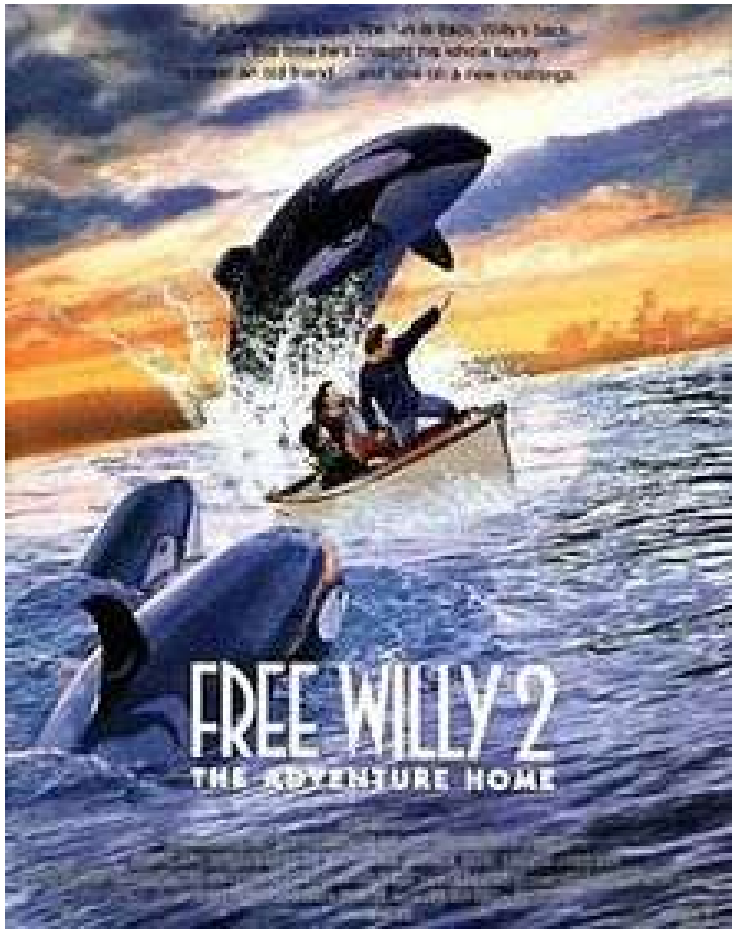
- Routines, subroutines, methods, or functions (not to be confused with mathematical functions, but similar to those used in functional programming) simply contain a series of computational steps to be carried out

- Any given procedure might be called at any point during a program's execution, including by other procedures or even itself (recursion)

Procedure Visual Programming

- Visual effects have pushed modelling software in the film industry further and faster than the tools typically used by architects

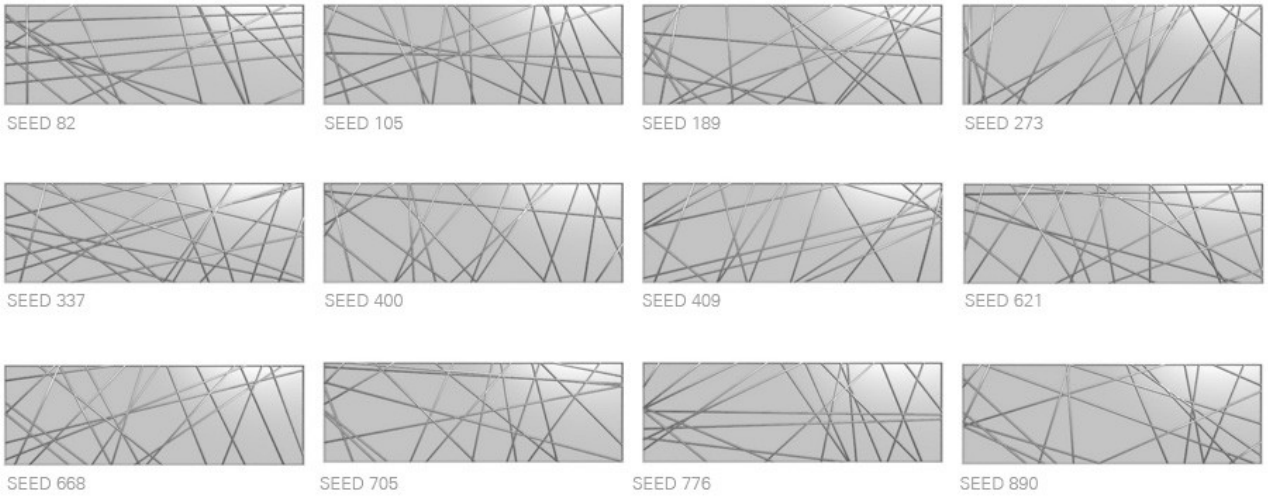
- While the demands of feature film are much different than architectural practice the same tools can be used to achieve very different goals



Procedure Visual Programming

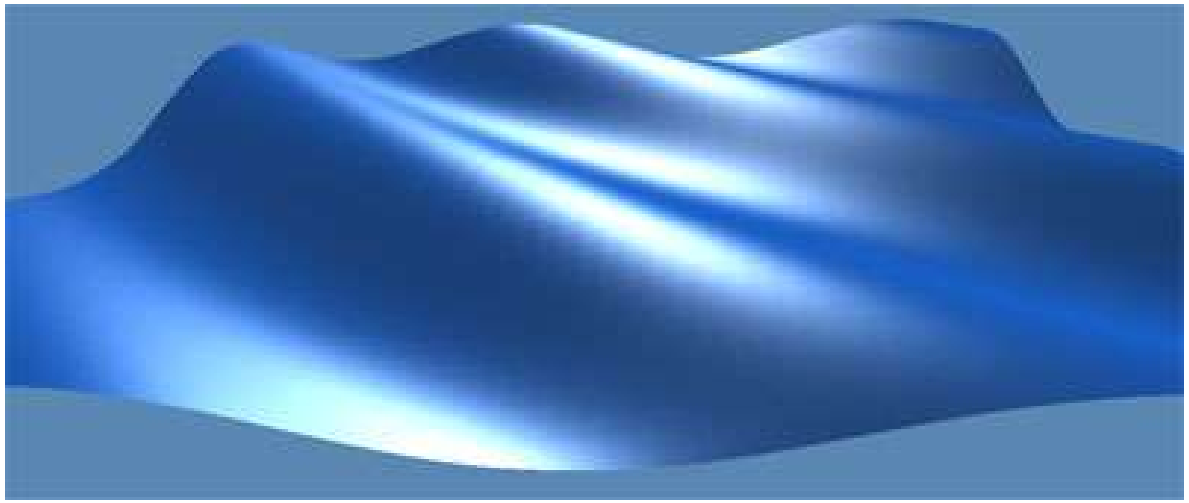
- The first step with any design is to synthesize the needs of the client into a building program which describes spaces and relationships

- Often what follows in the schematic design phase is series of sketches, models, and computer models to work out the size, orientation, and placement of different spaces



Example

- Procedural real-time image synthesis
- Modelling deep ocean waves
- Assignment 2

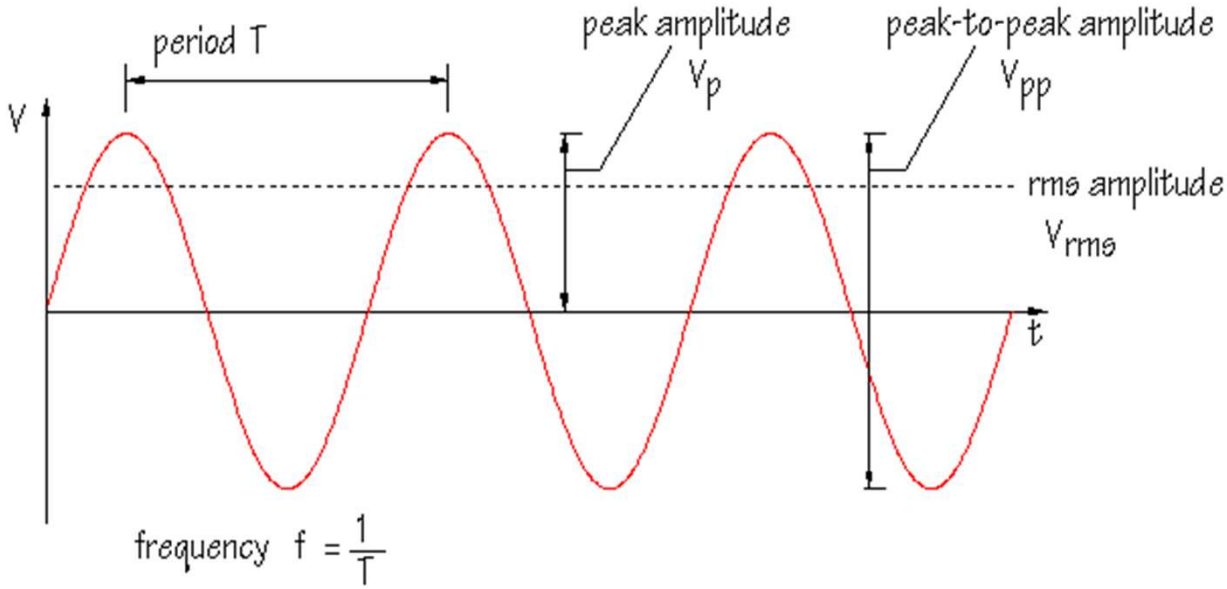


Note

- Assignment 2 might be better realised using GPU programming such as GLSL, HLSL, CG, CUDA, etc...
- Looking at these technologies will be part of the course in the second term...
- Following Assignment 2 You will have a better appreciation of the above compared to a high level CPU solution

Waves

- Model each wave as a sine function
- Gives smooth waves, good enough but not realistic
- Phase shift for different points
- Add multiple frequences (cf. noise / turbulence)



More Realistic Water

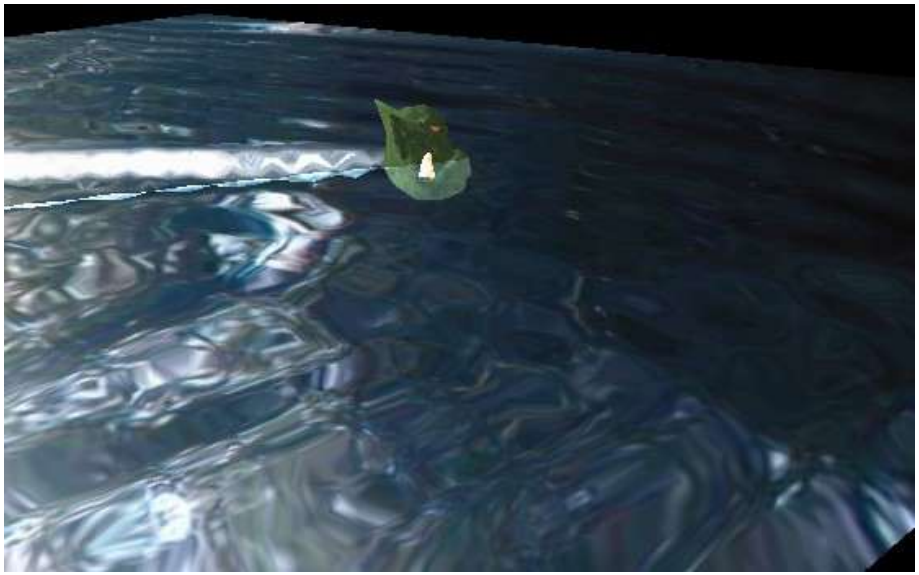
- **Advanced waves**
  - More complex wave functions
  - Depend on water depth (attribute variables!)

- **Model transparency**
  - Dependent on viewing angle (Fresnel coefficient)
  - Blend with background, colour dependent on depth

- **Reflections**
  - Environment map,
  - Render scene from mirror viewpoint

However – For REALLY Realistic Water

- **Model objects interactions with water**
  - Finite element method: solve Navier-Stokes
  - Particle based methods
  - CPU or GPU based solvers



First – EXPERIENCE!

- **BIG Waves**
  - 01 - Surf
  - 02 - Surf

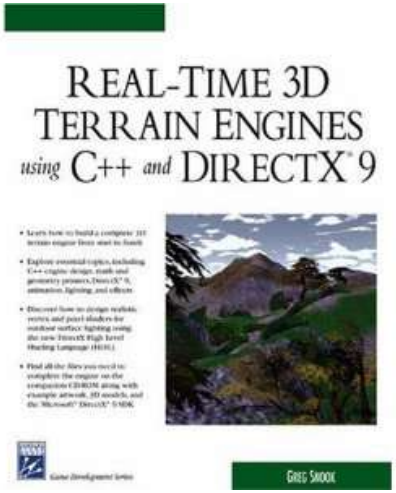
- **Natural Waves**
  - 03 – Ocean Waves

First – EXPERIENCE!

- **SWIM**
  - Even if only in the local swimming pool
  - Dive, feel the weight of water above you
  - Push, feel the medium slow you

- **Remember**
  - Family & Childhood memories
  - Holidays
  - Sea water tastes really salty!

Second – RESEARCH!



- **World Wide Web**
  - Images (Google)
  - Movies (Google)
  - Examples - Source Code (Google)
  - Sites
    - Information - http://vterrain.org/Water/
    - Papers - http://www.ebook-pdf.co.uk/pdf/animation-and-rendering-of-complex-water-surfaces.html
    - Engine - http://www.programsdb.com/download/14/59088/Aquatica_Engine.html
- **Read**
  - Books - Real-time 3D Terrain Engines Using C++ and DirectX
  - Research Papers....

Second – RESEARCH!

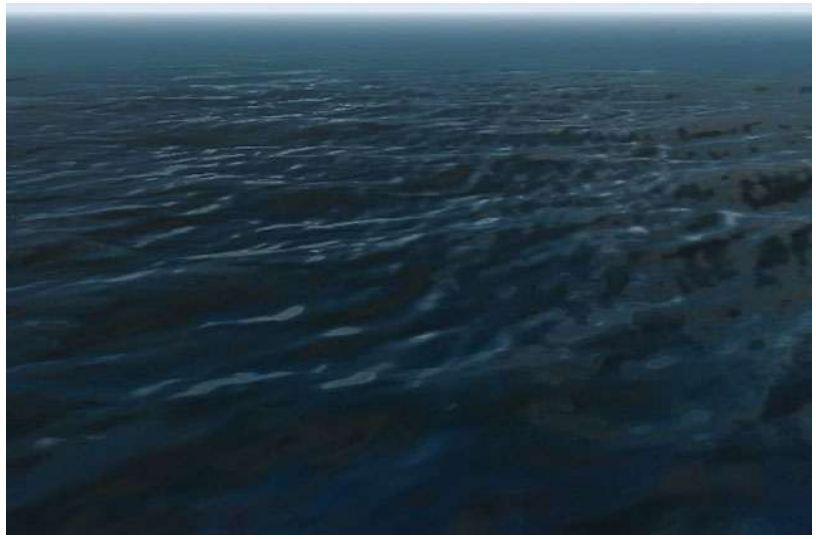
- **Research Papers**
  - 1986 – Simple Ocean Waves - Alain Fournier
  - 1994 – A Simple Model Of Ships Wakes - Khan
  - 2000 – Rendering Natural Waters – Premoze & Ashikhmin
  - 2001 – Simulating Ocean Water - Tessendorf
  - 2002 – Real-Time Rendering Of Ocean Waves
  - 2003 – Procedural Model For Animation Breaking Waves - Jeschke, Birkholz & Schmann
  - 2005 – Mitchell Real-time Synthesis And Rendering Of Ocean Water
  - 2009 – Rendering Water And Land Interaction

Third – EXPERIMENT!

- **Prototype:**
  - 01 - CG Waves
  - 02 - CG Waves
  - 03 - CG Waves

Third – EXPERIMENT!

- **Develop**
  - It takes time to “grow into” an understanding
  - Never evolve a prototype into the real thing
  - Accept that several prototypes might be necessary
  - Cut and paste code snippets
  - Document
  - Archive



End