*Learning Outcomes: Overall*

* Students should leave the workshop with a good working knowledge of Bifrost and a level of knowledge enabling them to work with Bifrost, create and provide simple tools to Bifrost users, troubleshoot, and support those tools in a basic manner. Most importantly, they should be comfortable doing so.
* Students should be familiar and comfortable with Bifrost concepts including compounds and ports, workflow, noodles, scattering, displacement, fields, arrays, solvers, fluids, particles, MPM, USD, publishing and sharing.
* Students should see Bifrost as a viable and smooth tool to use in their post-education careers – they should be able to bring the skills they learn in this workshop to the companies they are hired by.

Week 1 : Displacement and Basic Fields

**Motivation:**  
Introduction, motivation and team building, mentorship

**Learning Outcomes:**  
Methodology of: node-based creation, Boolean logic and visual programming  
Concepts of: vectors (position and colour) arrays, fields, normals and displacement

**Schedule:**

Part 1:

* **Sizzle!** Who are you as a mentor, what have you done in your career, play the BF sizzle (to be supplied). Your reel can go here.
* Icebreaking, informal talk about proceduralism + getting a feel for what they already know from Blender/Houdini/Blueprints. Explaining that BF is not Blender, is not Houdini, but shares concepts with both. This is a good time to talk about the differences between procedural and analogue.
* Very basics of Bifrost:
  + Bifrost works left-to-right
  + Bifrost deals in data
  + Bifrost is a programming language
  + Bifrost is procedural and non-destructive

Part 2:

* Recap of programming concepts, Bifrost **IS** (visual) programming and therefore basic programming concepts are incredibly important going forward.   
  + Variables (float/float3/etc etc)
  + Conditionals – including **basic** logic
  + Arrays
  + Datatypes
  + Loops – for/each, iterate, do-while
  + These do not have to be Bifrost specific – these are basic general programming concepts
* Start going through the **week 1 sideshow**. Lead them through it, making sure they know what each thing does – the concept of fields needs concentration here. Relate them back to arrays, albeit “infinite” ones.  
  **Slideshow:** week1.pptx  
  **Standard file:** rubika\_week1.ma

**Break: 10m**

Part 3  
  
Finish with the **week 1 slideshow** with the class.

* For both sessions, the students are to complete the exercises in the slideshow as we go.
* Possible problematic topics: fields, logic
* Introduce this weeks “advanced” compounds and talk about them – the “advanced” file is designed to take what the week’s lesson is and push it to a slightly more advanced level. It is possible to get to the result of the advanced lesson from the standard lesson. The advanced lesson might use Rebel Pack nodes. The week, the advanced concepts are:
  + Materials – diagnostic vs constant surface vs standard
  + Assigning materials, achieving the look you want in the viewport (transparency/colour) and render
  + Basic logic (One if to rule them all)
  + Compound editing – UI/Colour picker
  + fCurve technique (broken Tangents)
* **Advanced file:** rubika\_week1\_advanced.ma

Wrap-up:

* Take questions, give answers. If you have time, do a quick recap of today’s concepts and results. Continue the icebreaking (talk to them about it) and give them their “fake homework” and any resources they can use to push forward over the week. These resources can include YouTube videos, bf Discord, etc.