The second SDK release of Lightning will be version 0.4, also known as **Alpha 1, Milestone 5, or SDK2.**

It will have many features over Version 0.3 (SDK1/Pre-Alpha), and **will not be 1.0 level feature wise** (that will be **Beta 1 – currently tentatively numbered v0.9**). It will also be refactored to quite a significant degree.

Suggestions are open on the Discord!

M5/SDK2/Alpha 1 features (unfinished)

1. Lightning Package Format

M5 will support a packaging format using LZMA compression that will allow you to launch an entire game from a single file.

1. Video Support

A Video API and Video class will be implemented using ffmpeg to allow video support in Lightning.

1. Universes

The engine will be enhanced in order to support multiple levels. This will be achieved through the creation of a “root universe xml” that will contain references to each DDMS XML present within the folder. A Universe Manager API will be available to load levels.

1. NuRender

A new renderer API will be implemented in order to simplify the rendering pipeline and RenderService, It is currently being developed outside of Lightning.

1. UIService: Dialog Tree API

A dialogue tree API will be implemented.

1. List<T> => InstanceCollection in RenderService, etc

Any method using List<T>s that can use InstanceCollections will be changed to use InstanceCollections.

1. LightingService

A lighting service will be instituted allowing for enhanced lighting of levels.- several types of lights will be implemented, such as PointLight, SurfaceLights (lights that illuminate a surface) and more!

1. Polaris: Feedback Server

Feedback functionality will be implemented into Polaris in order to facilitate developer feedback.

1. Keyframes

Animations will be enhanced to support keyframing and animating specific properties of a PhysicalObject.

1. ImageBrush No Longer Needed for Rendering Non-ImageBrushes

ImageBrushes will no longer be a parameter to the render method so that ImageBrushes do not need to be passed themselves.

1. DataModelSerialiser and DataModelDeserialiser split

DataModelDeserialiser’s serialisation methods will be split into DataModelSerialiser.

1. LoadGame(); global methods

In accordance with the Universe feature, global methods will be introduced for loading games and levels so that Lua scripts can use them, in lieu of exposing DDMS to scripts.

1. [LuaExposed] attribute.

To reduce development time, classes will be able to have a [LuaExposed] attribute added. This will automatically expose them to Lua.

REWRITE DDMS SO THAT EACH COMPONENT HAS ITS OWN PROVIDER TO MAKE IT EASIER TO EXTEND?