**Team 11 Initial Project Description**

**Members/Contact:**

* Brandon Jensen, brandon.jensen@ku.edu
* Jonah Mooradian, j211m087@ku.edu
* John Russell, j362r647@ku.edu
* Sierra Seacat, s831s804@ku.edu
* Nicholas Shaheed, nicholasshaheed@gmail.com

**Project Sponsor (if any):** n/a

**Project Description**

The project we are going to develop is a GPU-accelerated synthesizer. Many synthesizers (both digital and analog) create sound using simple oscillators: sine waves, square waves, sawtooth waves, etc. Generally, these synthesizers use somewhere between 3-15 of these oscillators concurrently to create different sounds, with a focus on fine tuning the parameters of these oscillators to achieve particular sounds. The goal of this project is to scale up the standard synthesizer, and use the GPU to create a synth that plays thousands of oscillators at once. This project will be an opportunity for musicians/programmers/etc. to create sounds, textures, and music that are not possible with conventional tools, hopefully allowing for new approaches to music making.

 The end result will include two interfaces to this synth: the first will be a domain specific language that will allow the user the maximal flexibility in controlling the synthesizer. Scripts in this language will be commands that control the oscillators at a higher level than fiddling with the parameters of a single oscillator. For example: have all oscillators slide to a single frequency over 3 seconds, or assign the frequency of each oscillator based off of its index.

 The second interface will be a GUI that allows for interactive controlling of certain subsets of the DSL, expanding the possible users of the synthesizer beyond people with experience programming.

 Both interfaces will be plugins to existing digital audio software.

**Project Milestones**

* 3-5 specific and measurable objectives per semester for first & second semester
	+ Fall:
		- Project proposal video
		- Project proposal report
		- Coding goal: Achieve basic audio output.
			* Get to a point where we can use the GPU for some basic tasks, and be able to translate those tasks to an actual sound that plays on a speaker.
		- Create DSL specification
			* This will include what commands to include in the DSL, specifying the control flow from code to speaker, and the overall syntactic structure of the language.
	+ Spring:
		- Create DSL parser/interpreter
		- Create virtual instrument plugin/GUI
		- Create GPU backend
		- Create demo video
		- Documentation
* Estimated completion date for each milestone
	+ Both implementation and documentation milestones
	+ Fall:
		- Project proposal video: October 27 - November 10
		- Project proposal report: October 23
		- Coding goal: create something that makes a sound: End of semester
		- Create DSL specification: End of semester
	+ Spring:
		- Create DSL parser/interpreter: Late February
		- Create virtual instrument plugin/GUI: mid-late March
		- Create GPU backend: mid-late March
		- Create demo video: End of semester
		- Documentation: End of semester

**Project Budget**

* Hardware, software, and/or computing resources
	+ Licenses for the Max/MSP software, should we decide to use Max as the audio “engine” we will write the synth for we will need 1+ licenses for the software.
		- <https://cycling74.com/shop?educational=1>
			* We could probably reuse the key
	+ GPU
		- GPUs are standard in most PCs, and the lab has a couple computers with hefty GPUs already. We will likely not need to purchase any additional hardware.
* Estimated cost
	+ $59 yearly license for Max/MSP subscription
		- Will potentially need up to 4 copies, should we go that route.
* Vendor
* When they will be required?
	+ Mid October to early November

**Work Plan**

* Who will do what?
	+ Create DSL parser/interpreter
		- Nick, Brandon
	+ Create virtual instrument plugin/GUI
		- Jonah
	+ Create GPU backend
		- Sierra, John
	+ Create demo video
		- Everyone
	+ Documentation
		- Everyone

**Github link:** <https://github.com/nshaheed/gigasynth> (It is currently private)