DigiTask 1  
Nuno Hespanhol  
Setembro 2018  
Versão 1

## Criar um esboço para o novo site de síntese granular

Análise inicial para desenvolvimento de aplicação web de síntese granular.

O trabalho desta DigiTask consistiu:

1. na análise de referências algumas aplicações de síntese granular
2. elaboração de um mockup de interface
3. pensar na implmentação técnica futura

### Referências

Achei por bem analisar outras aplicações de síntese granular, para poder melhor pensar a interface, interação, parâmetros, etc..

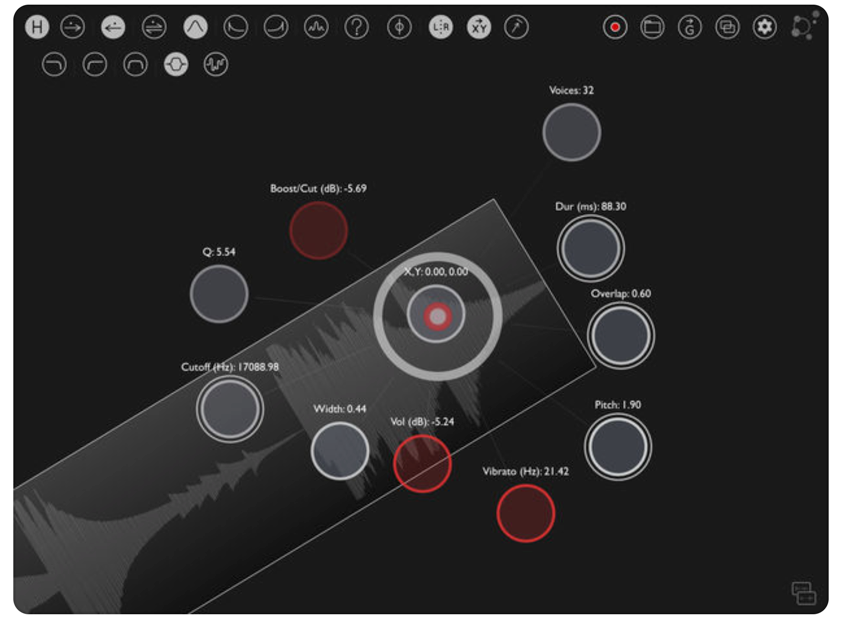
[Granulator II (Ableton Live, Robert Henke)](https://www.ableton.com/en/packs/granulator-ii/)



Esta será provavelmente a mais conhecida. Obviamente muito mais completa do que a que nós produziremos, mas interessante como referência de máximo de complexidade. Dispõe automaticamente de filtros, o que não chegámos a falar na altura, mas penso que seria interessante incluir também. A interface é relativamente complexa, devido à multitude de opções que têm. É essencialmente orientada a knobs, o que apesar de não ser exatamente uma interface inovadora é sempre familiar ao mundo musical.

[Borderlands (iPad App)](http://www.borderlands-granular.com/app/)

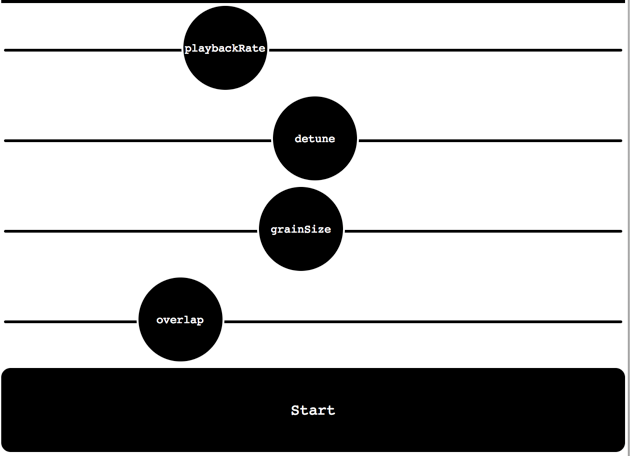
Provavelmente a mais inovadora a nível de interface interativa, sendo que é exatamente assim o próprio autor descreve a sua app, como uma app interativa de síntese granular. Será seguramente um bom exemplo a ter como referência para a nossa app.



Também é interessante que a versão inicial desta app é open-source, ou pelo menos em teoria. No site diz que se pode obter o código fonte, mas logo não é possível encontrá-lo em nenhum lado. Imagino que escrevendo diretamente ao autor conseguíssemos obter, mas de momento acho que também não faz falta.

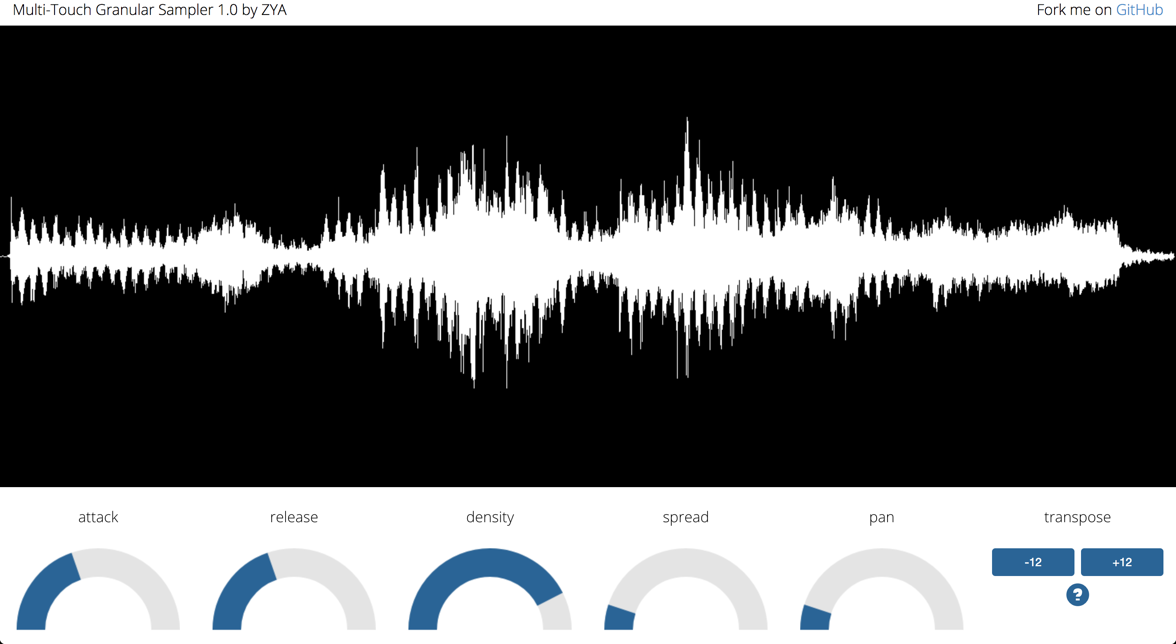
* [Link para site sobre versão inicial da app](https://ccrma.stanford.edu/~carlsonc/256a/Borderlands/#Downloads)
* [Link para app na App Store](https://itunes.apple.com/us/app/borderlands-granular/id561369733?mt=8)

[Exemplo oficial Tone.js](https://tonejs.github.io/examples/#grainPlayer) [[código](https://github.com/Tonejs/Tone.js/blob/dev/examples/grainPlayer.html)]



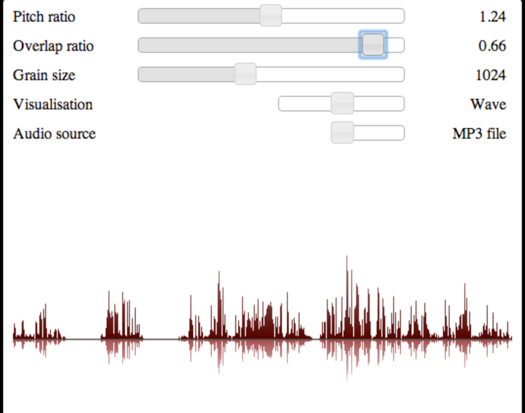
Deixo esta referência aqui, não pela interface ou interesse da app em si, mas como uma referência de futura implementação, uma vez que o mais certo utilizaremos o Tone.js. como biblioteca.

[Multi-Touch Granular Sampler (by ZYA)](http://zya.github.io/granular) [[código]](https://github.com/zya/granular/blob/gh-pages/js/main.js)



Esta é muito interessante porque é a única que já me permitiu experimentar síntese granular com WAA (Web Audio API) com várias vozes em simultâneo – via multitouch – e, ainda por cima, a correr diretamente do telemóvel. Para além disso a implementação também está toda disponível open-source, podendo ser um excelente ponto de partida para uma implementação nossa sem recurso a Tone.js, caso isso seja necessário.

[Web audio API pitch shifter](https://urtzurd.github.io/html-audio/static/) [[código]](https://github.com/urtzurd/html-audio/)



Outra referência implementada em Web Audio API, e que nos poderá vir a ser útil para referência de implementação da nossa, mais do que exatamente como “inspiração” a nível de app em si.

[Narrativas Sonoras ii (Rui Penha)](http://ruipenha.pt/works/)



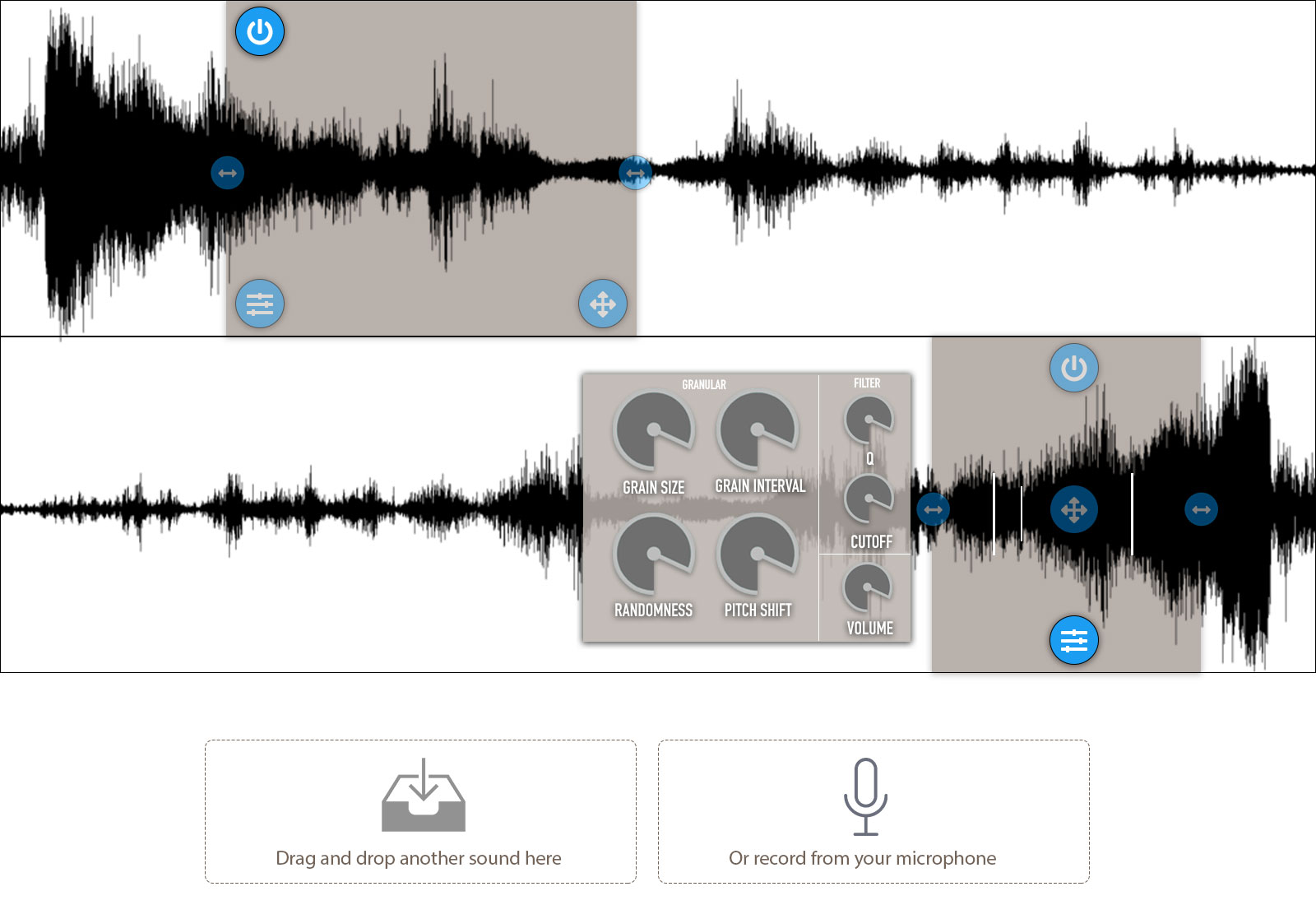
Esta dispensa apresentações. Incluia aqui – porque não o podia deixar de fazer, claro – e porque é naturalmente relevante, sendo que a feature de *repeat* de um “gesto” me parece particularmente interessante, e possivelmente algo a incluir na nossa app também.

### Mockup

Depois da análise feita e em função daquilo que já se tinha falado como intenção para a app, parece-me que as premissas reunidas (a nível de design/interação) eram pelo menos as seguintes:

* possibilidade de várias vozes (e não só de uma como a maioria das apps)
* interface não demasiado standard, evitando sempre que possível o recurso ao fader
* visualização minimante interessante

Naturalmente, que o design não é o meu forte (muito pelo contrário), e serão seguramente necessárias mais iterações – até por outras partes involvidas – para chegar à versão final. De qualquer maneira, a proposta inical seria esta:

Uma vez que temos tido um foco ultimamente em apps para dispositivos móveis, lembrei-me de considerar a possibilidade de gravar sons diretamente do microfone, de maneira a que seja fácil começar a usar a app, mesmo estando num telemóvel, sem ser preciso descarregar sons, carregar sons, etc..

Estão ainda naturalmente muitas coisas em aberto – aliás deixo até na imagem acima pequenas variações da mesma interface (itens centrados vs não-centrados) de uma maneira deliberada para podermos ir pensando em várias alternativas.

O esquema de cores também não é de longe um muito pensado nem imagino que o final. De certa maneira, acho que vai ser mais fácil até pensar nisso assim que tivermos a app feita, uma vez que será uma questão de mudar variáveis.. que eu com o Photoshop confesso que demoro muito mais tempo...

A nível de visualização da granulação achei melhor escolher uma interface que põe em evidência que estão a ser tocados segmentos de áudio vs “o círculo”, como é o caso do Narrativas Sonoras e do Borderlands. Apesar de essa visualização poder resultar potencialmente mais interessante visualmente, achei que por questões potencialmente didáticas, seria melhor adoptar a abordagem da segmentação do áudio, tal como referi antes.

Há várias coisas da imagem que ficam por explicar por texto, mas penso serem facilmente *self-explanatory*.

### Considerações de implementação

De momento, fica decidido que a envolvente (envelope de amplitude a usar) é o mais convencional, o de Gauss.

#### Web Audio API vs Tone.js

A nível de tecnologia há duas hípotese em aberto: usar a Web Audio API ou Tone.js. Depois da conferência em Berlim, fiquei com vontade de começar a usar a API sem recorrer tanto a bibliotecas como o Tone.js, porque apesar da sua conveniência acabam inevitavelmente por “esconder” alguns detalhes de implementação que podem muito bem vir a ser necessários mais tarde para questões de optimização/performance. No entanto, há que considerar também que o Tone.js muito provavelmente vai-nos poupar muito trabalho inicial, até porque já possui um objeto especialmente pensado para fazer síntese granular, o [Tone.GrainPlayer](https://tonejs.github.io/docs/r12/GrainPlayer). Concluindo, a abordagem inicial será a de tentar usar o Tone.js porque aparentemente nos permitirá avançar com a app mais rápido. Caso encontremos algum obstáculo por falta de configuração por exemplo, então nesse caso recorriremos diretamente à Web Audio API.