

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

lsi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer



diana teixeira_up201906220
diogo sousa_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

lsi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

índice

resumo _02
 parágrafo de sinopse _02
 palavras-chave _02
introdução _03
produção _04
 identidade visual_04
 parte sonora_05
 criação do código_06
 criação do vídeo_07
conclusão e análise crítica_08
referências_10
anexos_12_

agradecimentos

obrigado pela grande ajuda na
resolução do código_
 daniel shiffman
 golan levin
 xin xin
 pedro amado
 joão cruz (jp)

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

lsi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

CARTAZ



GENERATIVO



SEQUENCIADOR



JAM SESSION



BA * DUM * TSS

resumo

“ba dum tss” é o novo convidado das tuas jam sessions. Um sequenciador de som personalizável que, para além de criar uma sequência base para uma boa “guitarrada” com amigos, assenta na base de um cartaz generativo que podes até imprimir para guardar a memória desses momentos. Finally, the jam for your jam session... ba dum tss.

key-words

cartaz
generativo
sequenciador
jam session

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer



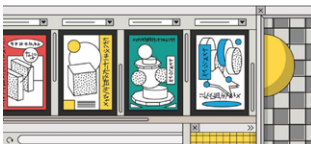
1_



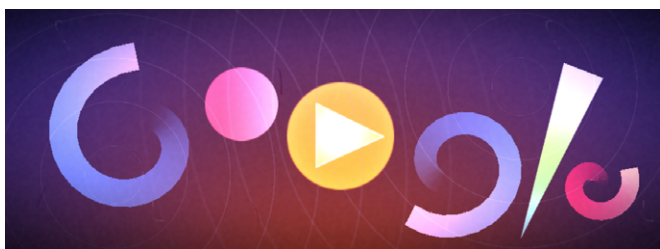
2_



3_



4_



5_

1_ serafim mendes; 2_ sebastian onufszak; 3_ mel cerri;
4_ mariano pascoal; 5_ google

introdução

O projeto que criamos em resposta à proposta “P2 - AudioVisual Sequencer” potenciou logo à partida um grande brainstorm de ideias por parte dos vários membros do grupo. O conceito final da aplicação surgiu principalmente da vontade de criação de um programa que acompanhasse “jam sessions” ao criar uma batida base, a partir da qual se poderia iniciar toda esta atividade. Achamos também que deveria estar ligado a um elemento visual esteticamente apazível e que fosse capaz de perpetuar visualmente todo este momento.

Assim sendo, inspiramo-nos visualmente em artistas como Serafim Mendes e o seu projeto “Post-Print”; Sebastian Onufszak, criador do projeto “Play with Four Egos”; Mel Cerri e o “Hello 70s Trio”; Mariano Pascual. Isto só para mencionar alguns (os restantes nomes estarão devidamente listados em anexo nas referências. Ver anexo 01).

Chegamos então à ideia de criar um cartaz animado com uma linguagem visual bastante eclética e divertida, cuja principal função fosse a criação de uma sequência de sons que, quando reproduzidos em loop, criassem a batida de uma música, ao mesmo tempo que aparecem no cartaz animações que refletem essa mesma sequência selecionada. O cartaz reproduziria então som ao escolher, numa sequência de símbolos (cada um com uma sample associada), num grupo de no máximo 8 “gráficos” que, quando ativados, reproduziriam a batida escolhida pelo utilizador. Esta seria a função e modo lógico de utilizar a aplicação, juntamente com alguns botões funcionais que iriam permitir a mais fácil composição do beat, permitindo opções tais como os botões que permitem apagar um ou a totalidade dos sons. Uma grande inspiração para a criação da ideia de aplicação foi o jogo também de som sequencial que a Google criou no 117º aniversário de Oskar Fischinger (mais imagens do jogo nas referências. Ver anexo 02).

Com estes objetivos em mente, e uma ideia base para a concretização do sequenciador, nos próximos parâmetros do relatório procuraremos falar de todo o desenvolvimento do projeto, quer do processo de conceção da parte visual do trabalho, quer da estrutura em código que faz toda a aplicação funcionar.

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

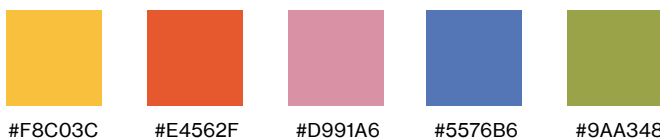
Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

produção

Neste processo de produção, houve uma clara divisão de tarefas para uma maior organização entre o grupo, o que, contudo, não impediu de toda uma troca de ideias e um trabalho sempre interdisciplinar, acabando por haver trocas constantes e momentos onde toda a gente conseguiu, de forma mais direta ou indireta, resolver algum aspecto em todos os parâmetros do projeto. Mesmo havendo 4 elementos do grupo, toda a gente estava ocupada com algum afazer e, para o avanço da aplicação, decorriam sempre múltiplas reuniões onde se decidia em conjunto e em absoluta concordância, os próximos passos a percorrer.

identidade visual

A primeira etapa consistiu na conversão da pesquisa mais geral feita por todos relativamente à parte visual, numa pesquisa um pouco mais reduzida e coerente que servisse de referência à criação de uma identidade para a aplicação. A parte visual foi uma tarefa mais direcionada ao Diogo, que criou principalmente a base de todo o layout, concebeu os símbolos, uma grelha modular, etc... Todo este processo de ajuste na relação código-imagem, foi mais tarde alterado e comentado em conformidade com as opiniões de todos os colegas e as necessidades de cada um (Ver anexo 03). Criou-se então 3 cartazes diferentes (na prática, 3 opções para cada botão), com 13 botões correspondentes a um som e animação cada, e em adição, alguns botões associados a ações. Devido ao tempo estimado para o projeto e tendo em atenção o peso que iríamos colocar ao correr a aplicação, tentamos tirar o melhor partido das conjugações que poderíamos obter. Terminamos com um total de 39 gráficos a preto e branco, 39 gráficos coloridos e 39 animações que se iriam misturar criando cerca de 8122425444 possibilidades de combinações diferentes de cartazes (informação retirada do seguinte [site](#)). Contudo, as alterações generativas apenas se aplicam aos 13 gráficos sendo que cada lugar na grelha do sequenciador (que tem como função a criação da batida) teria sempre um único sample correspondente. Criou-se também um botão



COCONAT

bold extended_colllettivo

BRULE

bold_andrei robu_typeeverything

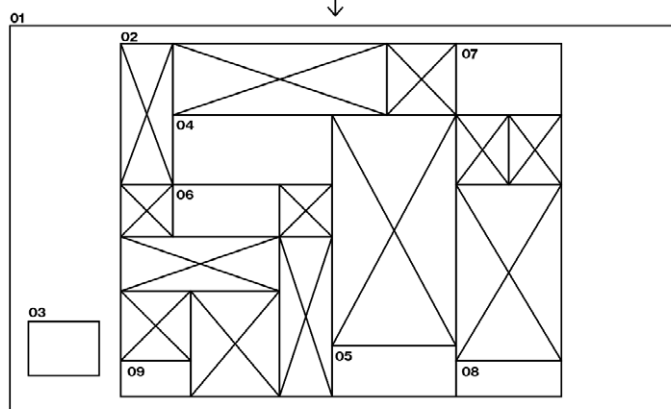
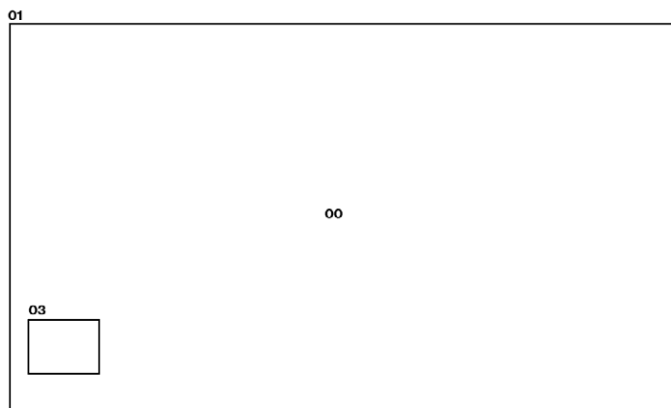
MARQUES

regular_andrei robu_typeeverything

BOUUUUUH

revenge_lift type

cartazes (a preto e branco e a cores); paleta cromática; tipografias usadas



- 00_ imagem de carregamento
- 01_ janela do browser
- 02_ display
- 03_ assinatura e nome da aplicação
- 04_ botão começar/parar o loop
- 05_ botão para apagar a última seleção
- 06_ botão para apagar todas as seleções
- 07_ indicador das seleções
- 08_ botão para produzir novo cartaz
- ×_ botão para imprimir
- 00_ botões para a criação das sequências



para o refresh da página (NEW) que gera sempre um novo cartaz; dois botões para apagar os samples selecionados (todos de uma vez com o TRASH, ou um de cada vez com o DELETE); um PLAY/STOP para tocar ou parar a sequência criada; e um botão que permite guardar e/ou imprimir um pdf do cartaz criado (PRINT). Para além disto, para que o utilizador soubesse quantos samples poderia selecionar e manter-se a par daquilo que escolheu, criamos uma barra que se vai atualizando em tempo real conforme a atividade do programa. Outros aspetos que incluímos foi a versão positiva e negativa dos botões que aparecerem à passagem do rato (hover) mas não correspondem a sons.

Na criação das animações, o Diogo e a Sara ficaram responsáveis pela concepção das pequenas e simples ações correspondentes a cada símbolo e da totalidade das animações, procurando simplicidade e harmonia entre elas até para que não tivessem muito peso, para que a aplicação carregasse o mais rapidamente possível. Este processo foi algo baseado no processo de “dois passos para a frente, um passo para trás”, por haver sempre algo que ou escapava ou não ficava bem o suficiente, que tinha que ser corrigido, etc.. Mesmo assim foi um processo bastante contínuo e mesmo havendo opiniões e gostos diferentes dentro do grupo, sentimos que cada membro sente as suas ideias representadas.

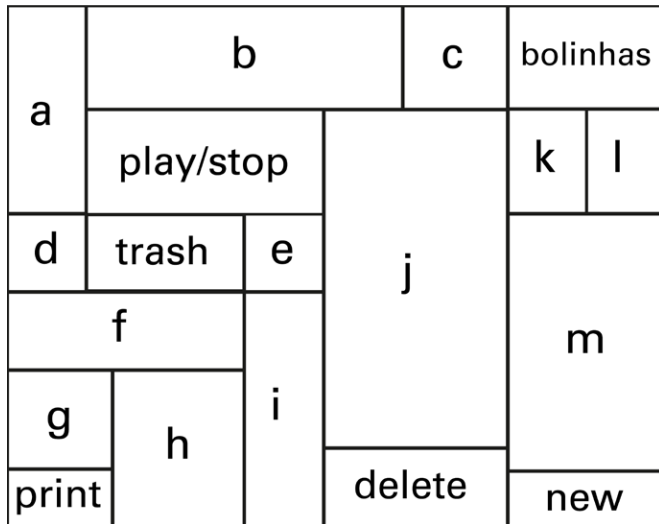
parte sonora

Esta parte ficou maioritariamente da responsabilidade do Tiago. Todos reuniram sons que achávamos relevantes e engraçados, fizemos uma seleção dos 13 finais, atendendo a certas características. Pelo menos alguns sons tinham que se assemelhar a certos elementos de uma bateria, neste caso snare ou caixa; hi-hat ou chimbau; e kick drum ou bombo. Os restantes sons foram selecionados por gosto pessoal, incluindo sons mais ou menos texturizados (harpas, flatos etc... até mesmo sons que simulassem o tal ba dum tss). A componente sonora do trabalho foi gravada e editada pelo Tiago usando o programa Ableton Live. (Algumas das nossas inspirações vão em anexo. Ver anexo 04)

diana teixeira_up201906220
 diogo souza_up201906221
 sara pinto_up201906676
 tiago santos_up201906934

design de comunicação_
 fbaup_faculdade
 de belas artes da
 universidade do porto

Isi_laboratório de
 som e imagem 20_21
 projeto_2
 audiovisual_sequencer



organização e nomenclatura dos botões

criação do código

Juntamente com a pesquisa visual, todos fizemos pesquisa sobre código, quer vendo vídeos do Daniel Shiffman, pesquisando online, etc... A Diana e o Tiago acabaram por ficar principalmente encarregues da escrita do código enquanto o Diogo e a Sara tratavam da parte visual, para haver um melhor fluxo de trabalho. No entanto, principalmente na criação de “pseudocódigo” e desenvolvimento de um pensamento lógico, em todas as reuniões verificava-se o ritmo do trabalho, apresentando problemas em conjunto e tentando da mesma forma resolvê-los. (Ver anexo 05)

O código que faz correr a aplicação desenvolvida, parte então de uma premissa de fácil interatividade e procura estabelecer uma dinâmica de exploração intuitiva, livre e autónoma com o utilizador. Para tal, procuramos empregar desde o início conceitos de código que nos foram transmitidos durante o semestre, tais como variáveis, condições, loops, funções, arrays, classes, etc. bem como achamos também importante explorar novas funcionalidades que o p5.js nos permitia, adquirindo assim novos conteúdos frutos de uma aprendizagem mais autónoma (Ver anexo 06).

Com a ideia da generatividade em mente, gerava-se um problema: o programa iria demorar mais do que o necessário ao carregar devido a uma enorme quantidade de gráficos, sendo que só daria uso a uma terça parte dos mesmos. Foi assim que chegamos à ideia de aplicar a generatividade, através do uso da função random, no preload (local onde se carrega atempadamente todos os elementos que deverão estar prontos a figurar a qualquer momento no jogo). Assim, seleciona-se logo ao início apenas os elementos que serão efetivamente utilizados. Esta tarefa, conforme é descrito no diagrama, acontece do seguinte modo: o programa gera um número decimal entre 0 a 3 e, consoante o intervalo em que esse número se insere, é selecionado 1 de 3 packs (contendo um gráfico preto e branco, um hover a cor e um gif também colorido) de um determinado botão. Isto repete-se e aplica-se aos restantes 12 botões sonoros. Os botões funcionais, o gráfico de bolinhas que conta os sons e os próprios sons são únicos e carregam sempre da mesma forma.

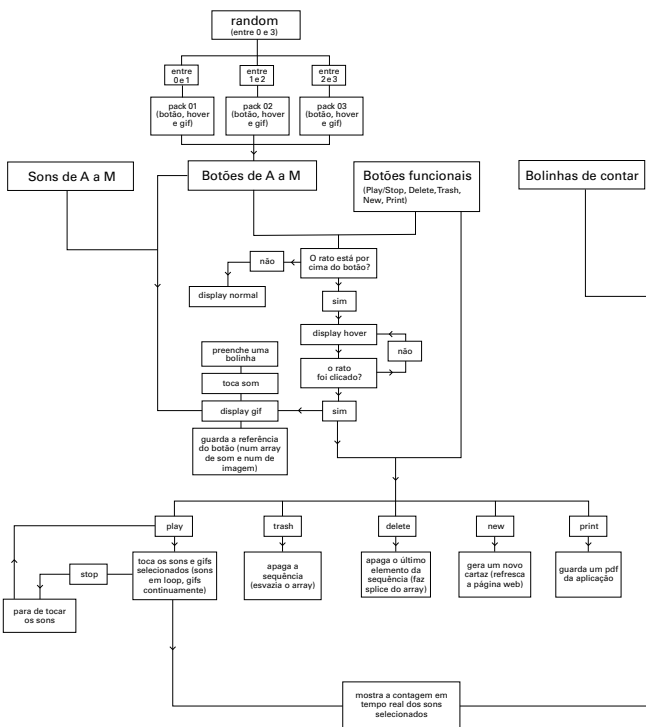


diagrama da aplicação (Ver anexo 09)

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

```
function preload() {  
  //gera um número random entre 0 e 3  
  //escolhe um set de imagens de acordo com o resultado  
  //dá preload a esse set de imagens  
  
  let randomizea = random(0,3);  
  
  if (randomizea < 1){  
    cartaza = loadImage("assets/cartaz_1/cartaz1a.png");  
    h_cartaza = loadImage("assets/cartaz_1/h_cartaz1a.png");  
    g_cartaza = loadImage("assets/cartaz_1/g_cartaz1a.gif");  
  } else if (randomizea >= 1 && randomizea < 2) {  
    cartaza = loadImage("assets/cartaz_2/cartaz2a.png");  
    h_cartaza = loadImage("assets/cartaz_2/h_cartaz2a.png");  
    g_cartaza = loadImage("assets/cartaz_2/g_cartaz2a.gif");  
  } else if (randomizea >= 2) {  
    cartaza = loadImage("assets/cartaz_3/cartaz3a.png");  
    h_cartaza = loadImage("assets/cartaz_3/h_cartaz3a.png");  
    g_cartaza = loadImage("assets/cartaz_3/g_cartaz3a.gif");  
  }  
}
```

generatividade no preload

```
class Botao {  
  constructor(x, _y, _w, _h, images, imageh, imageg, som) {  
    this.x = _x;  
    this.y = _y;  
    this.w = _w;  
    this.h = _h;  
  
    this.image = images; //imagem standard - como o botão começa  
    this.correntimage = this.image; //imagem atual (vai sendo alterada)  
    this.imagehover = imageh; //imagem hover  
    this.clicked = imageg; // gif (quando clica)  
    this.switchsound = false; // "foi clicado?" (começa não clicado)  
    this.switchimage = false; // "foi clicado?" (começa não clicado)  
    this.som = som; //som de cada botão  
  }  
}
```

class Botão (define as regras de todos os botões)

```
//guarda a variável dos sons selecionados num array  
guardarSons[guardarSons.length] = this.som;  
  
//guarda as propriedades de cada gif clicado  
let gifObject = {  
  gif: this.clicked,  
  x: this.x,  
  y: this.y,  
  w: this.w,  
  h: this.h,  
};  
  
//guarda a informação do gifObject num array  
//esse array comunica ao resto do código (fora da class)  
guardarGifs.push(gifObject);
```

guardar as referências de sons e de gifs em arrays

```
// botão DELETE --> apaga um elemento do fim do array  
if (  
  mouseX > width * 0.48 &&  
  mouseX < width * 0.76 &&  
  mouseY > height * 0.85 &&  
  mouseY < height  
) {  
  tocarSons = false; // "estratégia anti-bug"  
  guardarSons.splice(guardarSons.length - 1, 1);  
  guardarGifs.splice(guardarGifs.length - 1, 1);  
}
```

funcionamento do botão delete

No desenvolver da escrita, fomos duplicando os ficheiros em que íamos trabalhando de forma a progredirmos e podermos experimentar livremente de uma forma não destrutiva, o que nos deixou, no final, com vários ficheiros que ilustram o progresso do código (Ver anexo 07). Ao escrever, tivemos também cuidado para que a aplicação fosse "bug-proof" e "user-proof", mantendo como única forma de interação o rato do utilizador e corrigindo erros que encontrávamos pelo caminho.

De forma a explorar uma biblioteca e enfatizando as diferentes possibilidades de cartaz, quisemos possibilitar o guardar de um pdf (Ver anexo 08) da aplicação no momento de uso. Esta por vezes mostra alguns erros nativos mas na sua maioria, consegue o pretendido. Como um extra, achamos também engraçado poder assinar a aplicação diretamente na janela do browser e criar até um gráfico de "loading" (distinguindo-se, no entanto, pelo humor inerente ao projeto). Aqui devemos um obrigado ao professor pela ajuda.

Por fim, achamos também relevante descrever sucintamente o importante uso que demos à criação de uma classe de objetos, o que nos permitiu uma grande simplificação lógica do funcionamento de cada botão, estabelecendo aqui os seus parâmetros e ações comuns (que depois variavam consoante a necessidade de cada botão). Acrescido a isso, o uso dos arrays foi também fundamental, sendo que são o que permite guardar a referência dos sons e imagens para reproduzir mais tarde. Os botões funcionais ligam-se aos arrays e dão uso a funções existentes no programa para modificar a sequência à vontade do utilizador.

criação do vídeo

Para nós foi muito importante a criação de um vídeo que não só mostrasse como poderia ser utilizado o programa, como também o pudesse, de alguma forma, "vender". Assim sendo, com a direção da Sara, criamos um vídeo mais cómico (em coerência com a linguagem do jogo) de forma a prender a audiência à explicação da utilização da app. Para este efeito criamos alguns objetos que surgiram como aplicações de design de comunicação (presentes e explicados mais à frente no relatório). ([Ver aqui](#))

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

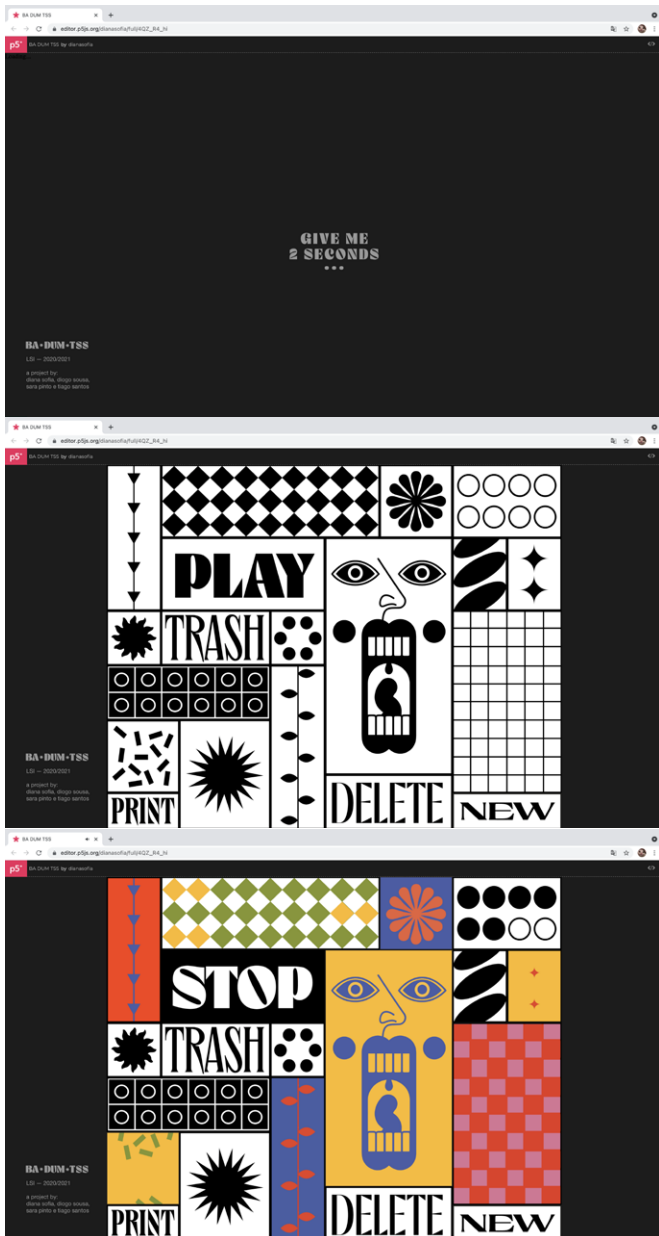
design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

conclusão e análise crítica

Chegando ao fim, achamos que, dado o tempo e as habilidades dos elementos do grupo, estamos extremamente contentes com o resultado final. Verdadeiramente achamos que conseguimos dar o nosso melhor. A nível de funcionamento dentro do grupo, pensamos que tenha sido bastante equilibrado e todos conseguiram dar o seu melhor e mostrar os seus “strong suits”, espalhando-se pelas diferentes componentes do trabalho, conseguindo haver uma grande interdisciplinaridade em vez de uma divisão convicta e uma assemblagem final bastante apressada, toda a gente sabia exatamente aquilo que se passava e dava o seu melhor para ajudar em todos os campos.

Ao expor o trabalho a amigos, estes divertiram-se bastante com todo o programa, achando a utilização fácil, divertida e flexível. No entanto, ao pedirmos que fossem críticos, estes apontaram com honestidade alguns aspetos a melhorar e com os quais concordamos. Comentou-se a possibilidade de guardar o próprio cartaz através de outro suporte no qual a aplicação pudesse viver fora do ecrã. A ideia pareceu-nos bem. Talvez pudesse haver também um layout mais próprio e reconfigurável. Talvez dar opções de layout de cartaz (horizontal mas também vertical)? Talvez poder exportar todos os elementos do cartaz em svg para haver melhor qualidade e poder ser até montado de outra forma? Talvez conseguir visualizar imediatamente por cor os botões selecionados (antes de clicar play). As opções são muitas e quiçá se não as retomamos futuramente. Na realidade, nenhuma retira valor ao projeto desenvolvido, do qual temos imenso orgulho, mas sim ajudam a vê-lo de outra forma e apontam numa direção de melhoria.



print, posters, tshirts

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer



Posteriormente, para além do aperfeiçoamento milimétrico das animações e de alguns outros acertos, achamos também possível uma revisão geral do código com visão a uma funcionalidade mais otimizada, possivelmente mais leve, permitindo um mais rápido carregamento. Supomos a possibilidade de acrescentar até novas funcionalidades, podendo haver até versões temáticas dependendo, por exemplo, de um feriado (com novos grafismos, animações, samples...).

A nível de suportes de design gráfico, criamos alguns extras à proposta do próprio programa: um frasco de geleia ("the jam for the jam session"), que apareceu no vídeo, logo merecia uma aplicação específica (pensada pela Sara); T-Shirts (utilizadas no vídeo e na apresentação, mas que poderiam funcionar por si só como merchandising) e um cartaz decorativo que reflete a vertente da impressão e aponta já para um possível redesenho de uma exportação de melhor qualidade.

aplicações do design de comunicação: jam for the jam session;
poster; t-shirt

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

anexo 01

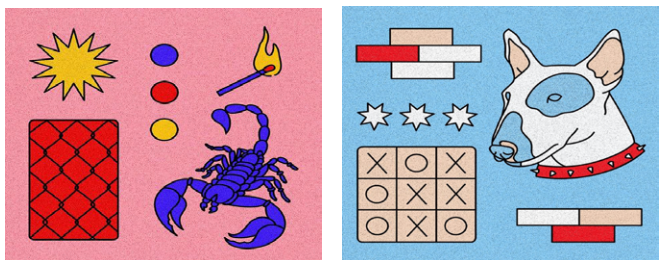
serafim mendes_post-print_2017



sebastian unofskan_play with four egos_2016
mel cerri_hello 70's trio_2020
mariano pascual_vários títulos_2016_20
logo for staf magazine_2016
matt wellins & music for the menphis movement_2017 (abaixo)
wild & free' shapes on shapes_2017
trendmark barcelona_2017_18 (abaixo, direita)
wild & free' ferns and stuff_2018
zero negative space for it's nice that_2020
website layout
ilustrações várias



victoria sieczka_scorpion & woof_2018



nasa_visions of the future mars poster_2020
mary kate mcdevitt_flyer for the republicans_2019
benny moore_isolation ideation_2020 (abaixo, esquerda)
lorenzo ballarini, andrea ricciarini, elena caricasoletto_poster promocional do
rebranding do estúdio mezzodetto_2019 (abaixo direita)



hannah pahl_ trying times_2020
go outside, look inside, 2020
alexander girard_fósforos la fonda del sol_meados do séculos 20
(abaixo, direita)
trevor basset_poster inspirado em yuragi_2020 (abaixo, esquerda)



bernardo henning_maria_2020
lisa congdon_ ferns_2020
allow something everyday_2020

tipografias utilizadas_
collettivo_coconat bold extended_2019
andrei robu for typeeverything_brule bold_2020
andrei robu for typeeverything_marques regular_2020
lyft type_boooooh revenge_2020

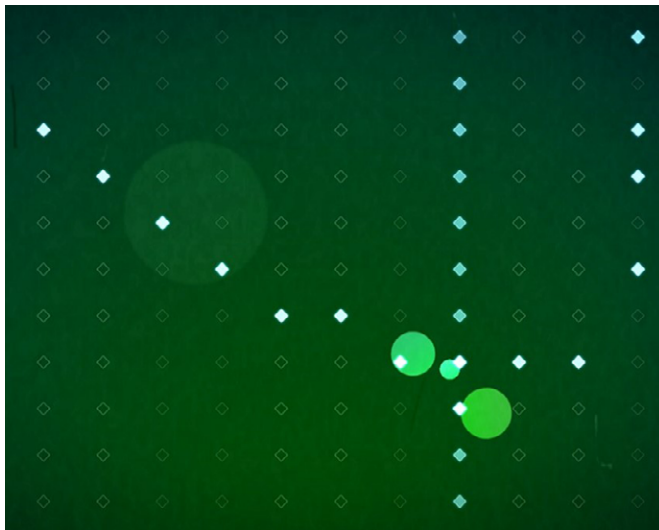
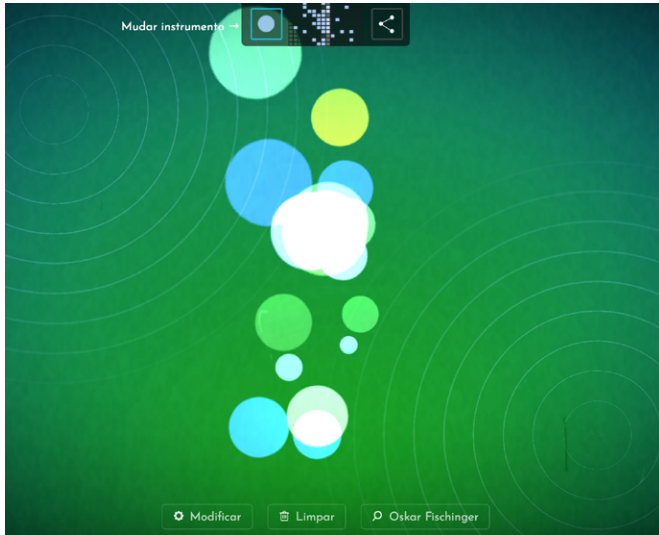
diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

Isi_laboratório de
som e imagem 20_21
projeto_2
audiovisual_sequencer

anexo 02

referências de código lógica e funcionamento_
google_117º aniversário de oskar fischinger_2017



anexo 03

versões antigas e tentativas de poster_



anexo 04

músicas que nos inspiraram_
make me feel_janelle monae_2018
introdução do álbum "song machine,
season one: strange times_gorillaz_2020



diana teixeira_up201906220
 diogo sousa_up201906221
 sara pinto_up201906676
 tiago santos_up201906934


design de comunicação_
 fbaup_faculdade
 de belas artes da
 universidade do porto

Isi_laboratório de
 som e imagem 20_21
 projeto_2
 audiovisual_sequencer

anexo 05

esboços aleatórios e registo de ideias durante reuniões de grupo

- testar deteção com alphas
 - monocromático - cor quando clara
 - formas geométricas (vectorial)
 - sons "random"
 - cartaz horizontal
 - tentar incluir
 - modular mas não quadrado

Botão  por dentro da composição

↳ play/stop → play parar de existir e aparece (substitui)
 ↳ guardar loop
 ↳ guardar pdf
 ↳ ctrl + z #/backspace
 ↳ parar lixo
 ↳ botão play e stop

bibliotecas
 matter.js
 ps. collide 2D
 save strings
 load strings

← ver para guardar sons

02:12 → gain código

FRAME COUNT % (modules)
 width: 0.76
 width: 0.24
 height: 0.2

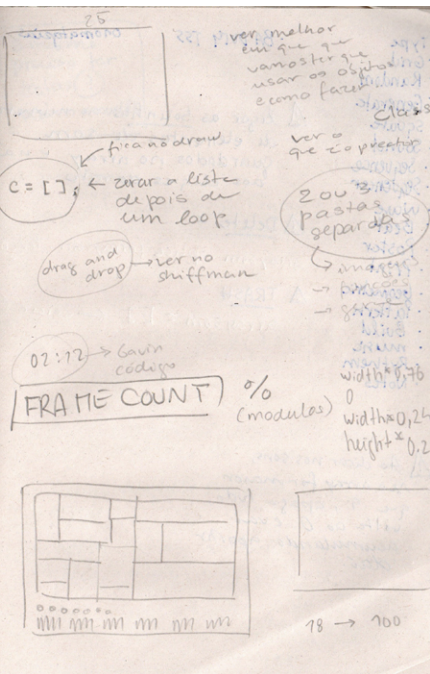
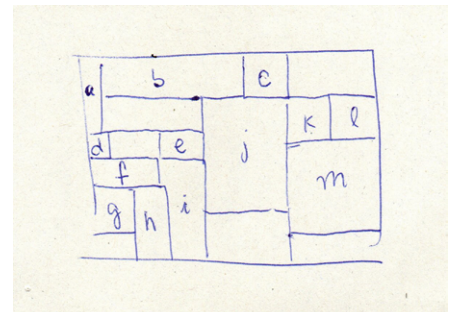
20 ou 30 pastas separadas

ficar no draw
 zerar a lista de pois de um loop

drag and drop - iter no shuffleman

ver o que é o player

ver melhor eu que quem vamos usar os objetos e como fazer

← não está a tocar de play

← toca de stop

← o som não está a play loop

• A cada x tempo, aumenta o (1)
 quando o (1) chega a 8, reinicia

tempo = 0
 f (frameCount % 30 == 0)
 tempo ++
 f (tempo == 8)
 tempo = 0;
 guardar Sons [tempo].play

tempo
 tem de ser um loop

0 1 2 3 4 5 6 7 8

0:37
 1:12
 2:33
 11/12:55

• Type
 • Grid
 • Random
 • Generate
 • Square
 • Sound
 • Sequence
 • Sequencer
 • Wave
 • Beat
 • Poster
 • Print
 • Geometric
 • Pattern
 • Build
 • Music
 • Rhythm
 • Notes

• BA Dum TSS onomatopias

⚠ ligar as bolinhas ao número de elementos de som guardados no array e não aos cliques do rato

⚠ Delete =
 arrayson.splice (arrayson.length - 1, 1);

⚠ TRASH
 arrayson = [] ← zerar (?)

⚠ Ao clicar nos sons, se o array for maior que 9, apaga tudo, volta ao 0 e vai acumulando apagar daí.

É mesmo preciso ter telas? diz no moodle

if (mousePressed) {
 stop = true; else if (!mousePressed) { stop = false; }

son.play ()
 son.stop ()
 son.setVolume (0.5); (entre 0 e 1)

dentro do botão play que nunca ultrapassa o 8

for (i = 0; i < arrayson.length; i++) {
 arrayson[i].play (); ou array [i].play ();
 }

Se clicares aqui, então adiciona isto ao array dos sons

arrayson.length + 1 = som_a

4 parâmetros na classe dos Botões: images, amaseh, images, som

adiciona este variável ao array

ps. sound.js → isPlaying ()
 ↳ retorna true or false

mousePressed ()
 f (!song.isPlaying ()) {
 f (!song.isPlaying ()) {
 f (!song.isPlaying ()) {
 } else {
 song.pause ();
 muda botao para (play)
 }
 }

poster generate
 jump
 Audio
 change
 new
 make
 create
 construct
 another one
 compare
 compare

⚠ loop (for) tem de ser infinitamente
 ↳ tem de parar quando o ato for deixado (se estiver a dar)

↳ se o rato clicar sobre um som, reinicia o loop infinito

↳ os sons se podem ativar de x em x tempo (ritmo)

While

no classe dos Botões, se o som for parar, não se deve fazer for clicado quando

5/1.5 = 0
 6/1.5 = 1
 7/1.5 = 2
 8/1.5 = 3
 9/1.5 = 4
 10/1.5 = 7.0

botão Amassouca
 ↳ função repetitiva

diana teixeira_up201906220
diogo souza_up201906221
sara pinto_up201906676
tiago santos_up201906934

design de comunicação_
fbaup_faculdade
de belas artes da
universidade do porto

lsi_ laboratório de
som e imagem 20_ 21
projeto_ 2
audiovisual_ sequencer

anexo 06

daniel shiffman_ the coding train_ vídeos vários
golan levin <https://www.youtube.com/watch?v=r5ly3v1co0A>
xin xin <https://www.youtube.com/watch?v=lWt5GYvnQtE>
entre outros

anexo 07

link com as tentativas e o progresso do código
<https://editor.p5js.org/dianasofia/collections/SRvg1G8KB>

anexo 08

link da biblioteca utilizada
<https://github.com/zenozeng/p5.js-pdf>

anexo 09

diagrama em maior escala

