

Programació Creativa

Memòria de Projecte Final de Grau/Màster

Grau Multimèdia

Narratives visuals, 2D i 3D

Autor: Rafel Bernabeu Aznar

Consultor: Aniol Marín Atarés

Professor: Antoni Marín Amatller

11/01/2021



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Dedicatòria

A Aida, la meva companya de vida.

Al meu pare, per donar-me sempre el seu suport.

A la meva família, que sempre estan presents.

Als que ja no m'acompanyen en aquesta aventura de la vida.

“Viure sense vosaltres, quin dur aprenentatge.”

Abstract

El present projecte presenta una sèrie d'obres visuals generades amb programació creativa per tractar de ser utilitzades com a eines per a la relaxació i/o meditació.

Per a dur a terme el projecte s'han aplicat els coneixements adquirits al Grau Multimèdia i especialment la narrativa gràfica 2D durant l'execució de les diferents peces. Aquestes s'han creat utilitzant el llenguatge de programació processing per a tractar d'emetre un missatge de pau i tranquil·litat.

Tots els elements que apareixen a les obres han sigut generats per ordinador, utilitzant la programació com a eina per a poder crear-les. S'han generat expressament per al projecte i per a ser distribuïdes a l'entorn digital.

Cada peça representa una interpretació de diferents elements, les quals han sigut inspirades en diferents recursos gràfics existents a la xarxa. El resultat de cada creació s'ha lliurat dins d'una plataforma que ens permet visualitzar-les i interactuar amb elles sense cap instal·lació addicional a l'ordinador dels usuaris.

L'objectiu principal del projecte és que aquestes obres puguin ser utilitzades per a deixar la ment en blanc durant un període curt de temps i així ajudar a mantenir la calma davant de l'estrès quotidià que representa moltes vegades la vida.

Per a desenvolupar la totalitat del projecte s'han actualitzat i adquirits nous coneixements sobre programació i grafisme 2D amb animació, tant coneixement teòrics com artístics i tècnics sobre el software i el seu ús.

Per a l'execució del projecte s'ha fet servir el llenguatge de programació processing. Aquest llenguatge permet generar gràfics per ordinador executats mitjançant codi per a generar obres visuals de molta diversitat.

Per a concluir, el projecte tracta d'emetre un missatge utilitzant la Narrativa 2D com a mitjà en un entorn on l'art digital és predominant en tot el que ens envolta.

Paraules clau: programació, programació creativa, processing, relaxació, meditació, animació 2D, treball fi de grau, memòria.

Abstract (english version)

This project presents a series of visual works generated with creative programming to try to be used as tools for relaxation and / or meditation.

To carry out the project, the knowledge acquired in the Multimedia Degree, and especially the 2D graphic narrative has been applied during the execution of the different pieces. These have been created using the processing programming language trying to issue a message of peace and quiet.

All the elements that appear in the works have been generated by a computer, using programming as a tool to create them. They have been expressly generated for the project and to be distributed in the digital environment.

Each piece represents an interpretation of different elements, which have been inspired by different graphic resources existing in the network. The result of each creation has been delivered within a platform that allows us to view and interact with them without any additional installation on the user's computer.

The main objective of the project is that these works can be used to leave the mind blank for a short period of time and thus help to remain calm in the face of the daily stress that life represents many times.

To develop the entire project, new knowledge about 2D programming and graphics with animation has been updated and acquired, both theoretical and artistic and technical knowledge about the software and its use.

For the execution of the project, the processing programming language has been used. This language allows you to generate computer graphics executed using code to generate visual works of great diversity.

To conclude, the project tries to issue a message using 2D Narrative as a means in an environment where digital art is predominant in everything around us.

Keywords: programming, creative programming, processing, relaxation, meditation, 2D animation, final degree project, memory.

Notacions i Convencions

En aquesta memòria, s'han utilitzat les següents tipografies per a diferenciar els continguts textuais:

- Títol de secció: Arial, bold, 20 pt, color negre.
- Títol de subsecció: Arial, bold, 12 pt, color negre.
- Títol d'apartat: Arial, bold, 10 pt, color negre.
- Text: Arial, regular, 10 pt, color negre.
- Enllaços: Arial, regular, 10 pt, color blau.
- Peu d'imatge i notes a peu de pàgina: Arial, regular, 8 pt, color blau.
- Codi: Courier New, regular, 10 pt, color negre.
- Text imatges de màrqueting: Ailerons¹, regular, 40 pt, color negre.

Es destaca que s'utilitza la lletra cursiva per a mots de llengua diferent del català, excloent els noms propis i les paraules relacionades amb la tecnologia.

¹ <https://www.behance.net/gallery/25541553/Ailerons-Typeface>

Índex

1. Introducció	9
2. Descripció	10
3. Objectius	11
3.1 Principals	11
3.2 Secundaris	11
4. Marc teòric	12
5. Continguts	16
6. Metodologia	22
7. Arquitectura de l'aplicació/sistema/servei	24
8. Plataforma de desenvolupament	25
8.1 Software	25
8.2 Hardware	26
9. Planificació	27
10. Procés de treball/desenvolupament	30
11. APIs utilitzades	40
12. Diagrames UML	42
13. Prototips	44
14. Usabilitat/UX	49
15. Perfils d'usuari	51
16. Seguretat	52
17. Tests	53
18. Versions de l'aplicació/servei	55
19. Requisits d'ús	57
20. Instruccions d'instal·lació/implantació	58
21. Instruccions d'ús	61
22. Projecció a futur	63
23. Pressupost	65
24. Anàlisi de mercat	66
25. Màrqueting i Vendes	69
26. Conclusió	70
Annex 1. Lliurables del projecte	71
Annex 2. Codi font (extractes)	72
Annex 3. Llibreries/Codi extern utilitzat	77
Annex 4. Guia d'usuari	78
Annex 5. Bibliografia	79
Annex 6. Vita	81
Annex 7. Diagrama de Gantt	82

Figures i taules

Índex de figures

Figura 1: George Nees, 1965-1968	12
Figura 2: TFG_Galàxia_01	16
Figura 3: TFG_Mandala_02	17
Figura 4: TFG_Vòrtex_03	18
Figura 5: TFG_Plexus_04	19
Figura 6: TFG_Nervis_05	20
Figura 7: Referència Galàxia	33
Figura 8: Referència Mandala	34
Figura 9: Referència Vòrtex	35
Figura 10: Referència Plexus	36
Figura 11: Referència Nervis	38
Figura 12: Imatges Màrqueting	39
Figura 13: Web Processing	40
Figura 14: Web OpenProcessing	41
Figura 15: Diagrama UML	42
Figura 16: Diagrama UML TFG_Galàxia_01	43
Figura 17: Esbós TFG_Galàxia_01	44
Figura 18: Esbós TFG_Mandala_02	45
Figura 19: Esbós TFG_Vòrtex_03	46
Figura 20: Esbós TFG_Plexus_04	47
Figura 21: Esbós TFG_Nervis_05	48
Figura 22: Exemple TFG_Galàxia_01	49
Figura 23: Exemple Web OpenProcessing	50
Figura 24: Exemple TFG_Mandala_02	53
Figura 25: Exemple test TFG_Mandala_02	54
Figura 26: Exemple producció TFG_Mandala_02	55
Figura 27: Exemple postproducció TFG_Mandala_02	56
Figura 28: Càrrega script	58
Figura 29: <Head>	59
Figura 30: <Body>	59
Figura 31: Instruccions	62
Figura 32: Imatge heart/work	66
Figura 33: Imatge Silk2 Generative Art	67
Figura 34: Imatge màrqueting TFG_Plexus_04	69

Índex de taules

<u>Taula 1: Llenguatges de programació creativa</u>	<u>13</u>
<u>Taula 2: Software</u>	<u>25</u>
<u>Taula 3: Hardware</u>	<u>26</u>
<u>Taula 4: Dates claus del projecte</u>	<u>27</u>
<u>Taula 5: Tasques del projecte</u>	<u>28</u>
<u>Taula 6: Metodologia Scrum</u>	<u>29</u>
<u>Taula 7: Pressupost del projecte</u>	<u>65</u>
<u>Taula 8: DAFO</u>	<u>67</u>
<u>Taula 9: Mapa de posicionament</u>	<u>68</u>

1. Introducció

Durant els últims anys he tingut la sort de poder estudiar el Grau Multimèdia a la UOC, un propòsit que arrossegava durant molts anys. L'any 2012 vaig deixar els meus estudis d'Enginyeria Informàtica a la UPV perquè vaig perdre de les ganes d'estudiar i vaig decidir apuntar-me a un Màster d'animació 3D a l'escola Primer Frame de València. Una vegada acabat el màster vaig trobar feina molt ràpidament dintre d'aquest sector i m'he dedicat aquest fins a l'actualitat.

Dins del cinema d'animació 3D, he estat professionalitzat amb la programació d'eines per als artistes i amb la creació d'esquelets per a poder animar als personatges. Aquests dos camps de coneixement són les parts més tècniques existents d'aquest sector, però sempre hi ha conceptes artístics que intervenen en cadascun de les tasques que he desenvolupat.

Gràcies a algunes assignatures del Grau Multimèdia vaig descobrir el llenguatge de codi obert de "processing" i amb ell tot l'art existent creat amb programació. Des de ben petit sempre he tingut un vessant artístic fort, però sempre lligat amb processos tècnics importants que normalment suposen problemes a resoldre. Aquesta resolució dels problemes que acaben convertint-se en projectes gràfics són el que més interès intel·lectual em generen.

És per això que, per la barreja de tecnicismes i art que he tingut sempre al món professional tinc l'interès a crear un projecte de recerca on estudio la programació creativa, ja que mescla perfectament el món tècnic pur amb la programació dels algoritmes, amb el món creatiu del gràfic per ordinador.

Dins de la programació artística els aspectes que més interès em generen són els gràfics plans o 2D generats purament per codi amb essència minimalista, però amb tocs creatius originals que donen sentit als programes generats.

2. Descripció

Aquest treball de final de grau (TFG) consisteix a fer una sèrie d'obres visuals utilitzant la programació amb processing, per a mostrar diferents elements geomètrics reconeixibles amb la finalitat de generar diferents peces visuals interessants que continguin interaccions amb els usuaris. La intenció final d'aquestes obres és transmetre relaxació i que els usuaris puguin utilitzar-ho com a eina per a la meditació.

Per a dur a terme el projecte se seguirà un procés de treball iniciat de zero fins a assolir tots els objectius. La planificació establerta a l'inici del TFG serà d'utilitat per a desglossar totes les tasques i d'aquesta manera poder organitzar el calendari i assolir d'una manera senzilla els lliuraments i obtenir els resultats desitjats.

Per tal d'organitzar el projecte d'una manera lògica, el dividirem en diferents fases: Preproducció, producció, postproducció, distribució.

En la fase de preproducció s'han buscat referències gràfiques per tal de trobar la identitat gràfica de les diferents peces generades. Aquesta fase fou el punt de partida per a tot el desenvolupament del projecte buscant així elements que en l'àmbit gràfic puguin ser interessants per a la creació de cada obra.

En la fase de producció s'ha dut a terme la programació de les diferents obres. Mitjançant el llenguatge de processing i totes les seves funcionalitats s'han codificat les peces tenint en compte afegir interactivitat amb l'usuari i buscant un resultat atractiu visualment per a l'usuari. Les característiques tècniques principals, com les dimensions i la qualitat final han estat condicionades per la plataforma de distribució final, <https://www.openprocessing.org/>.

A la fase de postproducció, s'ha buscat l'aspecte final de les peces visuals, ajustant paràmetres com el color, els materials dels elements i la velocitat de les animacions dels diferents gràfics que apareixen a les obres. Un cop escollits aquests aspectes, s'ha exportat el conjunt d'arxius necessaris, amb el format natiu de processing (.pde).

A la fase de distribució s'han importat els arxius generats a l'anterior fase a la plataforma d'Openprocessing.org, per tal que l'usuari final pugui interactuar amb les obres sense tenir instal·lat cap software addicional al seu dispositiu personal.

El resultat del TFG seria una sèrie de peces interactives generades amb codi, d'estil minimalista amb la intenció de què siguin utilitzades com a petites eines per a la relaxació i la meditació.

3. Objectius

L'objectiu principal d'aquest TFG és el desenvolupament d'un projecte on representem la barreja entre coneixements tècnics i la seva representació gràfica.

3.1 Principals

Els objectius clau del TFG són:

- Utilitzar les bases teòriques per portar a la pràctica un projecte amb resultats palpables.
- Crear un conjunt d'obres visuals amb una temàtica definida i coherent.
- Generar programes amb interactivitat mitjançant els perifèrics per a mantenir l'atenció dels usuaris.
- Aplicar els coneixements adquirits durant els estudis del Grau Multimèdia, amb un resultat professional i que apliqui moltes de les àrees formatives del Grau.
- Distribuir les peces generades a la plataforma d'openprocessing².

3.2 Secundaris

Els objectius addicionals següents enriqueixen el TFG i poden patir variacions:

- Buscar que les aplicacions programades definides siguin interessants en l'àmbit artístic.
- Difusió de les peces generades a les xarxes socials.
- Transmetre mitjançant les obres generades exposar conceptes com el disseny minimalista i la simplicitat aplicada als conceptes tècnics o teòrics.

² <https://www.openprocessing.org/>

4. Marc teòric

- Antecedents:

L'art generatiu fa referència a l'art que completament o en part ha sigut generat per un sistema autònom³. Un sistema autònom en aquest context és generalment un que no és humà i pot determinar de forma independent les característiques d'una obra d'art que d'una altra manera requeririen decisions preses per l'artista.

La utilització de la paraula generatiu en l'àmbit d'art s'ha desenvolupat amb el temps. És en 1960 quan el terme "art generatiu" comença a utilitzar-se gràcies a Margaret Boden i Ernest Edmonds, en l'ampli context dels gràfics generats per ordinador automatitzats. La primera exposició d'aquest tipus fou de Georg Nees en febrer de 1965 amb el títol de "Generative Computergrafik".

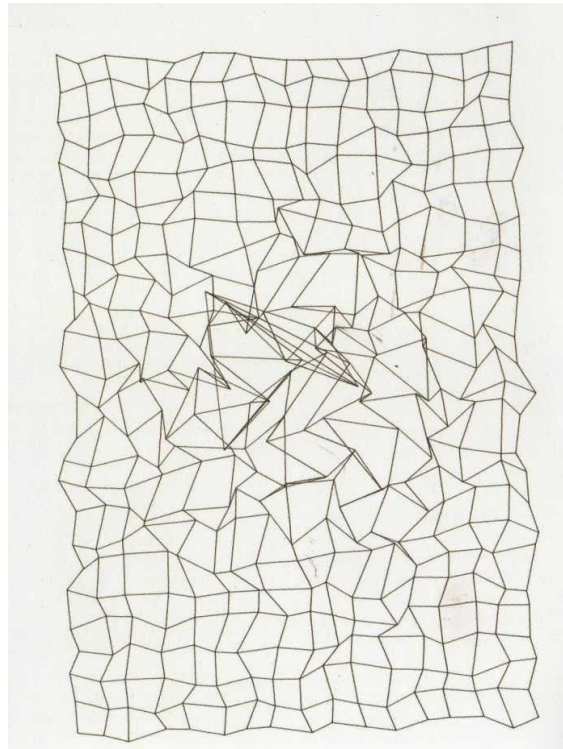


Figura 1: George Nees 1965-1968⁴

L'art generatiu utilitza la programació creativa ("creative coding"⁵) com a eina que és un tipus de programació informàtica en la qual l'objectiu és crear programacions expressives en compte de funcionals. L'aplicació d'aquesta té diferents àrees d'acció com per exemple, generar imatges en viu, crear dissenys i art visual, entreteniment, instal·lacions d'art, projeccions i "mapping", art amb so, publicitat, prototipats de productes i moltes més aplicacions.

La utilització d'aquest tipus de programació per a generar art va començar en la dècada del 1960. Posteriorment, grups com Compos 68⁶ exploraren amb èxit la programació amb finalitats artístiques, creant exposicions on mostraven el seu treball internacionalment.

³ Boden, Margaret A. 2009. What is Generative Art?. In: Digital Creativity. London:Routledge

⁴ <https://elartedigital.wordpress.com/artistas/george-nees/>

⁵ https://en.wikipedia.org/wiki/Creative_coding

⁶ https://monoskop.org/Compos_68

A partir dels vuitanta, els programadors experts es van unir a la cultura del “demoscene”⁷, utilitzant les seves habilitats per a crear obres visuals amb gran competència tècnica. Podem definir el terme “demoscene” com una subcultura internacional d’art per ordinador centrada en la producció de demostracions, és a dir, programes d’ordinador autònoms que produeixen presentacions audiovisuals. Els propòsits principals d’aquestes demostracions és ensenyar la programació, l’art visual i les habilitats musicals.

- Escenari:

Actualment podem trobar diferents exposicions i llibres com per exemple, A Philosophy of Computer Art (2009) de Dominic Lopes, on tracta d’examinar la importància de la programació integral en l’art contemporani més enllà de les interfícies que relacionen l’ésser humà amb les màquines. A banda, hi ha altres tipus de moviments com per exemple el de Toby Juliff i Travis Cox els quals critiquen el privilegi de les interfícies i l’usuari enfront de la condició integral del codi com a part important de l’art per ordinador⁸. La teoria que defenen, plantegen la programació creativa contemporània com a l’examen de codi i la seva intencionalitat com a una part integral de la comprensió del treball per part dels usuaris.

A més, ara mateix podem observar com existeix un interès renovat en la qüestió de perquè la programació creativa no ha tingut un gran impacte en la producció d’art. Google ha tractat de renovar aquest interès amb la seva iniciativa de Dev Art⁹, però aquest fet ha provocat diferents reaccions en desenvolupadors creatius que afirmen que canviar el nom per a descriure la programació creativa pot ser contraproductiu.

Encara que qualsevol tecnologia o llenguatge de programació es pot utilitzar amb finalitats creatives, existeixen certs llenguatges i llibreries de funcions que s’han dissenyat específicament per a ajudar a la creació ràpida de prototips i desenvolupaments de treballs creatius. Les eines de programació que s'utilitzen amb més freqüència en aquest context són:

Nom	Descripció	Sistema Operatiu	Llenguatge de programació	Llicència
Cinder	Llibreria per a programació amb intenció estètica, inclou funcions per a processament de gràfics, àudio, vídeo i geometria computacional.	Multiplataforma	C++	BSD de dues clàusules
generativepy	Llibreria per a la creació d’art generatiu i diagrames matemàtics com imatges o vídeos.	Multiplataforma	Python	MIT
Max MSP	Llenguatge de programació visual per a música i	Windows, Mac OS	Visual programming	Propietary

⁷ ["About the Demoscene"](#). Demoscene - The Art of Coding. Retrieved 22 May 2020.

⁸ <http://hdl.handle.net/11343/52793>

⁹ <https://devart.withgoogle.com/>

	multimèdia		language	
openFrameworks	Conjunt d'eines per ajudar al procés creatiu que proporcionen simplicitat i intuitivitat per a l'experimentació.	Multiplataforma	C++	MIT
openRNDR	Marc de codi creatiu dissenyat i desenvolupat per a la creació de prototips i desenvolupament d'aplicacions interactives i visuals d'alt rendiment.	Multiplataforma	Kotlin	BSD de dues clàusules
p5.js	Plataforma que permet als artistes, dissenyadors, estudiants o qualsevol persona aprendre a programar mitjançant entorns web. Basada en processing.	Qualsevol amb un navegador web.	JavaScript	LGPL
Processing	Llenguatge flexible i fàcil d'aprendre per a programar en el context de l'art per ordinador.	Multiplataforma	Java o Python	GPL, LGPL
Pure Data	Permet als usuaris a desenvolupar programes gràficament sense escriure línies de codi.	Multiplataforma	Visual programming language	BSD modificada
SuperCollider	Entorn i llenguatge de programació per a la síntesi d'àudio en temps real i composició algorítmica.	Multiplataforma	SuperCollider	GPLv3
Vvvv	Entorn híbrid de programació visual i textual per a un fàcil desenvolupament i creació de prototips.	Windows	Visual programming language	Proprietary

Taula 1: Llenguatges de programació creativa¹⁰

A banda del software, la programació creativa ocasionalment pot involucrar components de hardware per tal d'ingressar dades de l'entorn, produir resultats o interactuar amb els participants. Els exemples més comuns d'utilització de hardware inclouen micròfons, càmeres web, controladors de moviment, controladors MIDI, projectors, llums LED, impressores, etc.

Actualment la utilització de recursos gràfics a la web està en ple creixement, ja que les xarxes socials són el producte estrella del nostre temps i amb aquestes els reclams gràfics són d'una importància vital.

¹⁰ https://en.wikipedia.org/wiki/Creative_coding

El format més utilitzat per a la distribució d'aquest tipus d'obres és el format natiu de processing (.pde¹¹), i és per això que utilitzaré la web de <https://www.openprocessing.org/> per a distribuir les diferents peces interactives creades.

- Bases teòriques de referència:

Per tal de poder allotjar els formats finals del projecte utilitzaré la següent plataforma:

- <https://www.openprocessing.org/>. Aquesta plataforma ens permet executar arxius .pde (format natiu de processing) al núvol perquè l'usuari pugui interactuar amb el programa sense tenir cap llibreria instal·lada al seu ordinador.
- Altres projectes/estudis/productes similars o relacionats:
 - <https://artsandculture.google.com/>
 - <https://www.saatchiart.com/>
 - <https://www.pacegallery.com/>
 - <https://generativeartistry.com/>
 - <https://tholman.com/>
 - <https://apps.apple.com/app/heart-work/id1380764163>
 - <https://apps.apple.com/es/app/silk-2-generative-art/id1050339928>

¹¹ <https://processing.org/reference/environment/>

5. Continguts

El projecte “Programació Creativa” s’ha portat a terme per a tractar de generar diferents peces, mitjançant el codi, que permetin a l’usuari utilitzar-les com a eines per a la relaxació i la meditació. Per tal de demostrar-ho, he generat diferents programes interactius amb l’usuari.

L’objectiu principal del projecte és arribar a crear diferents obres visuals que es puguin atribuir com a eines de meditació o relaxació per a períodes curts de temps mitjançant la programació amb processing. Les peces que he creat combinen el desenvolupament de software amb un estil visual definit basant en la simplicitat i el minimalisme. És a dir, he representat les peces que hi ha a continuació, seguint una identitat gràfica definida.

Peces d’art generatiu: <https://www.openprocessing.org/user/252547#sketches>

1. Galàxia: <https://openprocessing.org/sketch/1046332>

Aquesta peça interpreta la creació d’una galàxia on els punts de naixements representen els planetes i les òrbites que generen. L’usuari interactua amb ella fent clic amb el ratolí per a generar una nova versió on els colors, el nombre de línies i les rotacions canvien amb uns valors aleatoris controlats per paràmetres.



Figura 2: TFG_Galàxia_01

2. **Mandala:** <https://openprocessing.org/sketch/1046414>

Aquesta peça representa un mandala, símbol ritual i espiritual del budisme i de l'hinduisme, que fa referència a l'univers. L'usuari pot interactuar amb l'obra mitjançant el ratolí i les fletxes per tal d'interactuar amb els diferents paràmetres modificables.

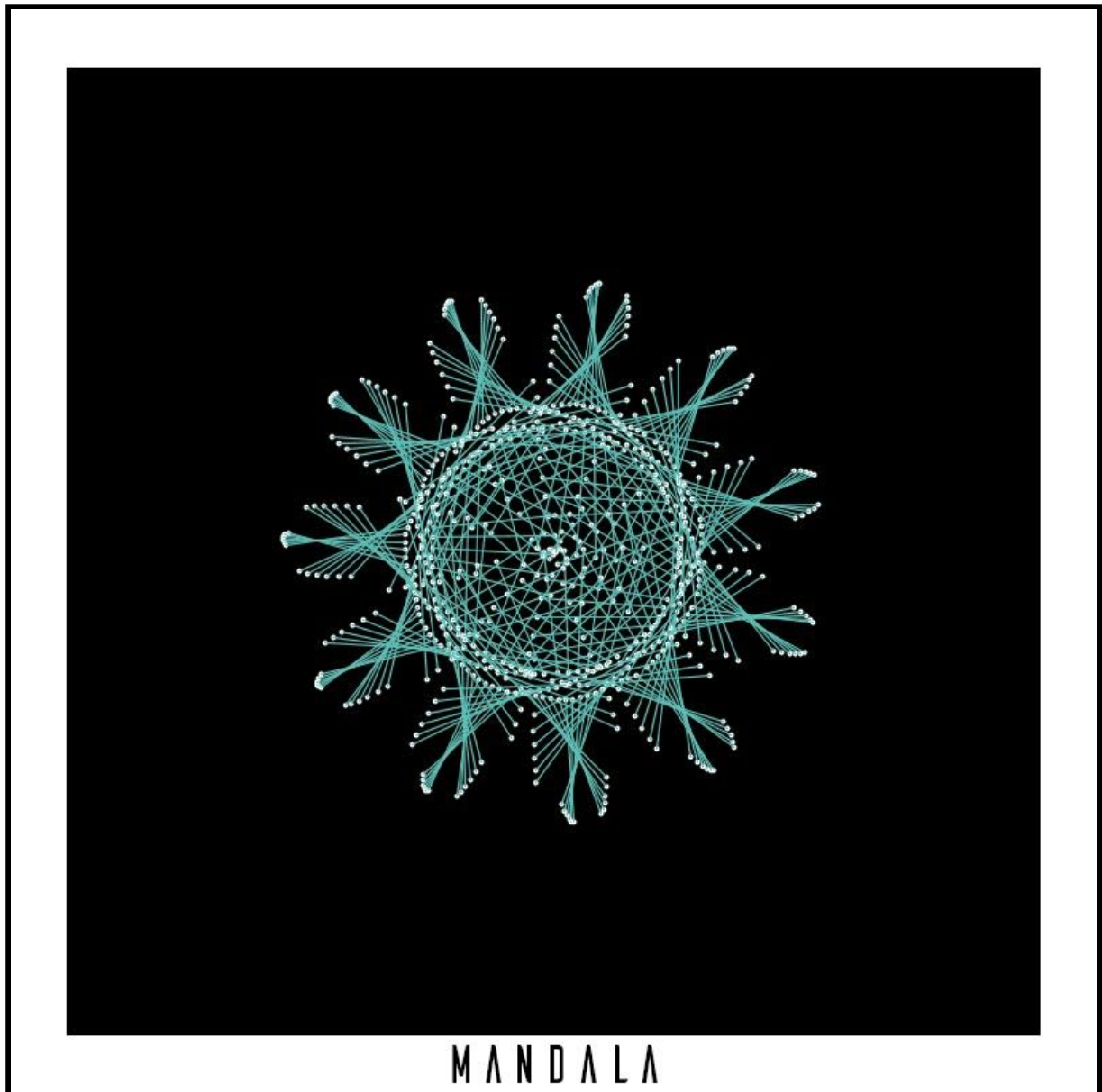


Figura 3: TFG_Mandala_02

3. **Vòrtex:** <https://openprocessing.org/sketch/1058312>

Aquesta peça representa un vòrtex, un flux giratori turbulent amb forma d'espiral. L'usuari pot utilitzar les fletxes del teclat per a interactuar amb l'obra. Tecla amunt o avall per a canviar el nombre de figures, tecla esquerra o dreta per variar el nombre de costats de la figura geomètrica.



Figura 4: TFG_Vòrtex_03

4. **Plexus:** <https://openprocessing.org/sketch/1058316>

Aquesta peça representa diferents connexions entre elements geomètrics de manera asimètrica. L'usuari pot interactuar amb ella fent clic amb el ratolí per a poder canviar els paràmetres com el color i les connexions entre els elements.

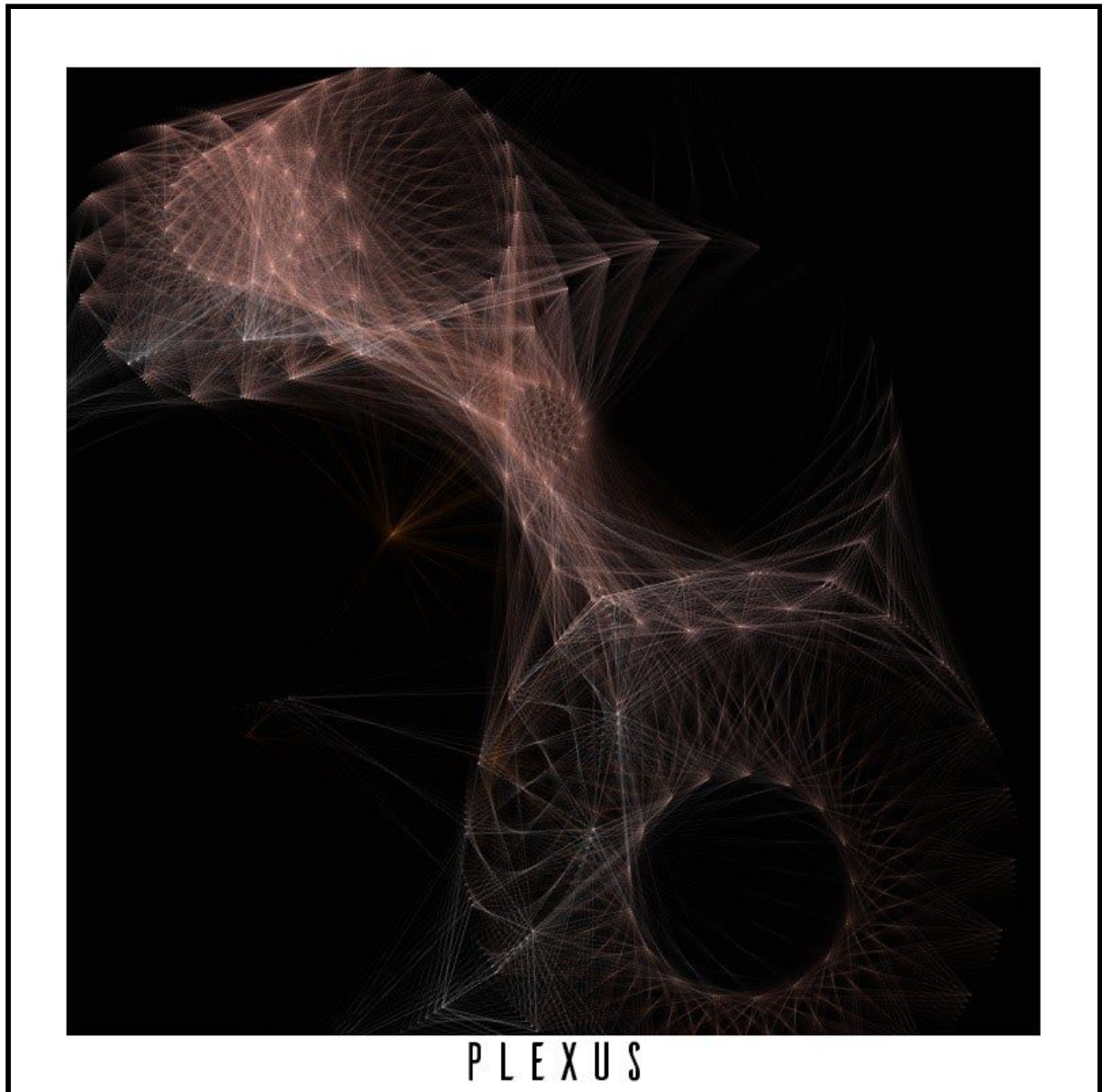


Figura 5: TFG_Plexus_04

5. Nervis: <https://openprocessing.org/sketch/1058318>

Aquesta peça interpreta les connexions nervioses, simulant fluxos entre els diferents sistemes que es generen. L'usuari interacciona amb ella fent clic amb el ratolí per a crear noves connexions nervioses o amb qualsevol tecla per a canviar el patró principal.



Figura 6: TFG_Nervis_05

Per tal de poder apropar a l'usuari a les obres, aquestes presenten diferents paràmetres amb interactivitat, amb les quals els usuaris poden variar mitjançant el teclat i/o el ratolí els valors escollits per a canviar les formes, colors i nombres d'objectes de cada peça. Com podem veure a cada peça existeixen les instruccions d'interactivitat on s'indica quins paràmetres varien en funció de les accions dels usuaris.

Es va decidir afegir interactivitat a les peces per a tractar de dinamitzar l'experiència de l'usuari. A banda de veure la demostració de la peça de manera automatitzada l'usuari podrà interactuar amb la

representació de l'obra tenint accés a canviar diferents paràmetres amb events assignats als diferents perifèrics. La intenció de mantenir aquesta interacció és crear un impacte positiu a l'usuari per tal de mantenir l'atenció centrada en l'evolució de la peça representada i que les accions que impliquin canvis a les obres, siguin simples per a buscar l'assoliment de l'objectiu principal de les peces, que és que ajudar als usuaris a relaxar-se i poder utilitzar aquests petits programes com a eina per a meditar durant un període de temps curt.

6. Metodologia

Durant tot el projecte s'aplicarà la metodologia Scrum¹², un marc de treball àgil que es basa en la iteració i lliuraments incrementals de desenvolupament d'un projecte.

El terme de metodologia àgil descriu una corrent alternativa a les metodologies de treball tradicionals, considerades com excessivament pesades i rígides, pel seu caràcter normatiu i les fortes dependències de planificacions prèvies detallades.

Els objectius principals d'utilitzar la metodologia Scrum consisteixen a valorar:

- Els individus i la seva interacció per sobre dels processos i les eines.
- El programari que funciona per sobre de la documentació exhaustiva.
- La col·laboració amb l'usuari.
- La resposta dinàmica davant dels canvis o problemes que puguin sorgir.

A l'apartat de Planificació podrem veure amb més detall els Sprints¹³ plantejats per a dur a terme el projecte. Aquesta planificació s'ha elaborat pensant en els terminis dels lliuraments plantejats a l'assignatura, dividint aquests en Sprints per a executar el projecte en la duració del semestre.

Pel que fa a la metodologia pròpia del treball, el model amb el qual intentarem analitzar i observar les diferents peces proposades, consistirà a realitzar les següents fases:

- Preproducció:

Aquesta primera fase tractarà de fer una cerca d'imatges o peces existents de referència per a utilitzar-les com a inspiració per tal d'intentar representar-les amb la programació en la fase vinent. En aquesta fase també escollirem els colors principals finals del nostre producte. Com el resultat de les peces han d'intentar transmetre serenitat i relaxació per tal d'utilitzar-les com a eina de meditació, aquesta primera fase és de vital importància per a la correcta execució de les següents.

- Producció:

Una vegada acabem amb la cerca de les imatges d'inspiració, haurem de convertir aquests elements gràfics de referència en valors programables. És a dir, utilitzar les diferents eines que ens ofereix la programació per tal de representar l'obra que busquem.

A banda, en aquesta fase, identificarem aquells paràmetres que siguin rellevants de les nostres peces com perquè l'usuari pugui interactuar amb aquests valors. Una vegada identificats aquests factors implementarem el codi necessari perquè l'usuari mitjançant diferents events amb els perifèrics (ratolí, teclat) pugui canviar valors que afectin directament al programa.

Per a la correcta realització de les diferents peces s'ha utilitzat la programació orientada a objectes¹⁴ com a metodologia principal, per tal de facilitar la representació d'un estil visual basat en la simplicitat i el minimalisme l'obra creada en qüestió.

¹² <https://www.scrumguides.org/scrum-guide.html>

¹³ <https://www.scrumguides.org/scrum-guide.html#the-sprint>

¹⁴ <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

- Postproducció:

En aquesta fase es realitzaran ajustos de correcció del codi i ajustos creatius per a finalitzar cada peça. Una vegada tinguem tota la programació desenvolupada, com la idea que volem transmetre és de tranquil·litat i relaxació les tonalitats dels colors escollits utilitzen saturacions baixes per a no crear un impacte visual massa gran a la vista dels usuaris. La velocitat de l'animació de les obres generades també està dissenyada acord en la finalitat última del projecte, intentant que les transicions dels elements que apareixen siguin suaus.

Per últim, amb tots aquests paràmetres establerts dins del codi es realitzarà l'exportació dels fitxers necessaris generats per a poder executar-los en la plataforma de distribució escollida.

- Distribució:

En aquesta fase, el codi que he generat i s'ha exportat a la fase de postproducció de les peces serà importat a la plataforma esmentada a l'apartat del marc teòric (<https://www.openprocessing.org/>) per tal que qualsevol usuari pugui interactuar amb les demostracions sense haver d'instal·lar cap classe de programa al seu ordinador.

7. Arquitectura de l'aplicació/sistema/servei

El llenguatge de programació que s'utilitza per a tractar de representar les demostracions és Processing¹⁵. He decidit escollir aquest llenguatge perquè compta amb una comunitat de programadors i artistes molt ampla, la qual cosa ajuda a resoldre dubtes d'una manera molt fàcil. A més, aquest llenguatge gaudeix d'una simplicitat clau a l'hora de codificar els programes, ja que amb l'estructura bàsica de funcions com són `setup()` i `draw()` ja ens permet generar programes amb resultats palpables. Per altra banda, aquest llenguatge compta amb una llibreria de funcions predefinides que ens ajuden a simplificar qualsevol mètode que necessitem programar, podem veure les diferents funcions existents al vinent [enllaç](https://processing.org/reference/)¹⁶. Aquesta llibreria es presenta com a recurs principal de l'API del llenguatge, però també ens ofereixen [llibries de caràcter més avançades](https://processing.org/reference/libraries/)¹⁷ per a poder treballar amb les imatges, l'àudio, el vídeo i la comunicació amb altres dispositius. També, com podem veure a la web pròpia del llenguatge existeixen una quantitat molt elevada de [tutorials](https://processing.org/tutorials/)¹⁸ gratuïts per a aprendre a dominar el llenguatge.

El paradigma de programació que he utilitzat per a totes les demostracions basades en codi és la programació orientada a objectes¹⁹.

Aquest model es basa en la idea de crear unitats independents, que continguin valors i dades emmagatzemades per a treballar amb elles i que les funcions puguin manipular aquestes dades.

Aquestes entitats no poden ser manipulades directament des de fora, és a dir, des d'una altra part del codi, sols podrem utilitzar-les aplicant els mètodes de comunicació que tinguem definits.

Els objectes seran per tant independents uns d'altres i sols necessitarem conèixer el protocol de comunicació amb el que podrem actuar sobre ells. Aquests protocols de comunicació estan establerts pel mateix llenguatge i per accedir a qualsevol atribut dels objectes generats utilitzarem la següent sintaxi "nomObjecte.nomPropietat" per tal de modificar qualsevol dels paràmetres que tinguem definit a la creació dels objectes. Aquests són reutilitzables, ja que són elements complets i autònoms i ens permetran programar d'una manera modular.

Aquesta modularitat ens permetrà treballar d'una manera molt simplificada els paràmetres a utilitzar per a la interactivitat amb l'usuari i mantenir una organització del codi definida amb classes, funcions i atributs.

Com podem veure, per a dur a terme aquest tipus de paradigma és essencial realitzar una fase de planificació inicial en la qual definim les classes, els objectes i els seus atributs segons les necessitats del nostre programa, ja que aquesta planificació ens ajudarà a resoldre el nostre programa més fàcilment i ens permetrà fer ús de la modularitat que ens aporta la programació orientada a objectes.

¹⁵ <https://processing.org/>

¹⁶ <https://processing.org/reference/>

¹⁷ <https://processing.org/reference/libraries/>

¹⁸ <https://processing.org/tutorials/>

¹⁹ Coad, P; Yourdon, E. (1991). Prentice-Hall International editions, ed. [Object-oriented Design](#). ISBN 9780136300700.

8. Plataforma de desenvolupament

A les taules següents es mostra la informació detallada dels recursos tecnològics que s'han utilitzat per desenvolupar el projecte "Programació Creativa".

8.1 Software

Nom	Especificacions Tècniques	Ús
Windows 10 Pro	Programari en local	Sistema operatiu
Google Chrome	Programari en local	Navegador web
Google Drive	Programari al núvol	Emmagatzematge dels arxius del projecte. Còpies de seguretat
Google Docs	Programari al núvol	Redacció del projecte
Google Sheets	Programari al núvol	Planificació i creació del diagrama de Gantt
*Processing 3.5.4	Programari en local	Programació de les obres
OpenProcessing	Programari al núvol	Eina utilitzada per a l'allotjament de les demostracions. https://www.openprocessing.org/
Adobe Acrobat DC	Llicència d'estudiant UOC	Edició de la memòria Exportació PDF
Adobe InDesign CC	Llicència d'estudiant UOC	Maquetació de la memòria
Adobe Media Encoder CC	Llicència d'estudiant UOC	Configuració i processament dels arxius d'exportació

Taula 2: Software

- **Característiques de la plataforma de desenvolupament principal, Processing 3.5.4:**
 - PDE (Processing development enviroment), és a dir, l'entorn de desenvolupament. És l'eina utilitzada per a programar en local i compta amb funcions bàsiques que conformen un entorn perfecte per a introduir-se en la programació i poder realitzar proves.
 - Col·lecció de funcions, ordres i mètodes que sumades a les llibreries existents permeten desenvolupar programes d'índole més avançada, com per exemple, l'enviament de dades en línia, lectura d'imatges des d'una webcam o guardar imatges complexes en format .PDF.
 - El llenguatge consta d'una sintaxi molt pareguda al Java però amb diferents modificacions. Podem entendre processing com a un llenguatge de scripting ubicat dins de l'entorn de Java, però sense el mateix nivell d'abstracció que presenta aquest.
 - Comunitat en línia activa al mateix lloc web del llenguatge, on es comparteixen coneixements, es resolen dubtes i es poden consultar les principals funcions del programa.

8.2 Hardware

Nom	Especificacions Tècniques	Ús
Ordinador de taula: Custom	Processador AMD Ryzen 7 1700 Eight-Core Processor CPU 3.00 GHz 32,0 GB de RAM Windows 10 (64 bits) Targeta gràfica NVIDIA GeForce GTX 1070 8 GB Samsung SSD 960 EVO M2 250 GB Samsung SSD 850 EVO 500 GB Seagate BarraCuda HDD 2 TB	Ordinador des del qual s'ha realitzat la gestió i el desenvolupament de totes les fases del projecte del TFG
Monitor 1: DELL U2515H 25"	Pantalla principal de treball	Escriptori expandit
Monitor 2: BENQ Zowie XL LCD 24"	Pantalla auxiliar de treball	Escriptori expandit
Teclat: Logitech G610 Mecànic	Cable USB	Eina utilitzada per a l'escriptura del TFG i la programació de les demostracions
Ratolí: Logitech G Pro	Cable USB	Eina per a la interacció amb el sistema operatiu
Auriculars: Corsair HS50	Cable Jack	Eina utilitzada per a veure diferent documentació

Taula 3: Hardware

9. Planificació

Informació detallada sobre la planificació de treball.

- Dates clau:

ACTIVITAT	INICI	FI
Pluja d'idees	16/09/20	23/10/20
PAC1	16/09/20	05/10/20
Cerca de referències	06/10/20	25/10/20
PAC2	06/10/20	02/11/20
Escollir peces	03/11/20	13/11/20
PAC3	03/11/20	07/12/20
Execució peces	08/12/20	11/01/21
Distribució	08/12/20	11/01/20
Lliurament Final	08/12/20	11/01/21

Taula 4: Dates claus del projecte.

- Detall de les tasques del projecte:

TASCA	INICI	FI	DURACIÓ (dies)
0.0 Pluja d'idees	16/09/20	23/9/20	7
1.0 PAC1	16/09/20	05/10/20	19
1.1 Títol, introducció, descripció, objectius, marc teòric, continguts, planificació i pressupost.	23/09/20	27/09/20	4
1.2 Definir el projecte i justificar-lo a nivell teòric i/o pràctic.	28/09/20	30/09/20	2
1.3 Planificar el desenvolupament del projecte, les fases i les tasques a realitzar.	01/10/20	03/10/20	2
1.4 Establir estructura principal de la memòria amb el consultor.	04/10/20	08/10/20	4
Lliurament PAC1	09/10/20	09/10/20	1
2.0 PAC2	06/10/20	02/11/20	27
2.1 Preproducció: desenvolupament del projecte i aconseguir les fites de la planificació.	06/10/20	9/10/20	4
2.2 Cerca de referències i documentació amb profunditat.	10/10/20	15/10/20	6
2.3 Revisar objectius i planificació per tal d'aplicar canvis si fou necessari.	16/10/20	21/10/20	6
2.4 Escollir temàtica i peces a desenvolupar.	22/10/20	27/10/20	6
2.5 Desenvolupament memòria.	28/10/20	30/10/20	3
2.6 Idees i conceptes	31/10/20	02/11/20	3
Lliurament PAC2	02/11/20	02/11/20	1
3.0 PAC3	03/11/20	07/12/20	35
3.1 Continuar el desenvolupament del projecte i assolir un estat avançat de cara a la consecució dels objectius.	03/11/20	07/12/20	35
3.2 Avançar la documentació de la memòria	04/11/20	09/11/20	6
3.3 Revisar els punts de la memòria per a fer canvis si apareixen errades.	10/11/20	13/11/20	4
3.4 Producció de les peces.	14/11/20	07/12/20	24
3.5 Postproducció de les peces.	14/11/20	07/12/20	24
3.6 Exportacions de imatges, màrqueting	14/11/20	07/12/20	24
3.7 Importar peces a openprocessing.	01/12/20	07/12/20	7
Lliurament PAC3	07/12/20	07/12/20	1
4.0 Lliurament Final	08/12/20	11/01/21	35
4.1 Finalitzar projecte. (Últims retocs)	08/12/20	11/01/21	35
4.2 Finalitzar memòria. (Maquetació)	01/01/21	11/01/21	11
4.3 Presentació del projecte per al públic general.	19/12/20	23/12/20	5
4.4 Presentació en vídeo per al tribunal d'avaluació.	24/12/20	29/12/20	5
Lliurament Final	11/01/21	11/01/21	1
5.0 Presentació divulgativa.	11/01/21	20/01/21	10

Taula 5: Tasques del projecte

- Diagrama de Gantt:

Per a realitzar la planificació del treball i poder veure totes les tasques que s'han realitzat i executat al projecte s'ha utilitzat aquesta eina. Aquesta ens dóna una idea de com avança el projecte i ens permet modificar les tasques si és necessari. Adjunto el Diagrama de Gantt del projecte a l'Annex 7 d'aquesta memòria.

- Metodologia Scrum:

Seguint el marc de treball Scrum, s'han planificat els següents Sprints a implementar durant el desenvolupament del projecte. La durada de cada Sprint està vinculada als terminis dels lliuraments de les PACs, que el seu total són els quatre mesos que dura el semestre.

SPRINT	ID	DESCRIPCIÓ	ESFORÇ
PRIMER	1	Títol, introducció, descripció, objectius, marc teòric, continguts, planificació i pressupost.	5
	2	Definir el projecte i justificar-lo a nivell teòric i/o pràctic.	3
	3	Planificar el desenvolupament del projecte, les fases i les tasques a realitzar.	4
	4	Establir estructura principal de la memòria amb el consultor.	3
SEGON	5	Preproducció: desenvolupament del projecte i aconseguir les fites de la planificació.	4
	6	Cerca de referències i documentació amb profunditat.	5
	7	Revisar objectius i planificació per tal d'aplicar canvis si fou necessari.	5
	8	Escollir temàtica i peces a desenvolupar.	2
	9	Desenvolupament memòria.	4
	10	Idees i conceptes	1
TERCER	11	Continuar el desenvolupament del projecte i assolir un estat avançat de cara a la consecució dels objectius.	5
	12	Avançar la documentació de la memòria	3
	13	Revisar els punts de la memòria per a fer canvis si apareixen errades.	2
	14	Producció de les peces.	5
	15	Postproducció de les peces.	5
	16	Exportacions de imatges, màrqueting	3
	17	Importar peces a openprocessing.	2
QUART	18	Finalitzar projecte. (Últims retocs)	5
	19	Finalitzar memòria. (Maquetació)	5
	20	Presentació del projecte per al públic general.	4
	21	Presentació en vídeo per al tribunal d'avaluació.	4
	22	Presentació divulgativa.	5

Taula 6: Metodologia Scrum

10. Procés de treball/desenvolupament

Es documenten tots els processos de treball que han estat necessaris per al desenvolupament del projecte. S'han seguit les fases definides en la metodologia scrum i el calendari establert en la planificació. Les principals fites han estat:

- **1r Sprint - PAC 1, 05/10/20**

Resum de les fites:

- S'han elaborat els apartats: introducció, descripció, objectius, marc teòric, continguts, planificació i pressupost.
- S'ha realitzat la planificació amb realisme i amb visió global de les diferents fases i tasques a realitzar, elaborant un diagrama de Gantt que ens ha servit de guia durant tot el desenvolupament del projecte.
- S'ha establert l'estructura inicial de la memòria en col·laboració amb el consultor de l'assignatura escollint les diferents temàtiques possibles.
- S'ha lliurat tota la documentació treballada i prevista en l'enunciat de la Pràctica d'Avaluació Continuada 1.

- **2n Sprint - PAC 2, 02/11/20**

Resum de les fites:

- S'han elaborat els apartats: metodologia, arquitectura, plataforma, procés de treball, diagrames UML, prototips, guions, usabilitat / ux, annex 4.
- S'ha rebut el feedback del lliurament de la PAC1 per part del consultor i s'ha procedit a corregir i adaptar la planificació i els textos de la memòria.
- S'ha començat el procés de documentació del projecte, s'han escollit els fenòmens a estudiar i la recerca d'aquests.
- Aplicant algunes de les pautes del consultor s'han pres decisions sobre l'execució de les demostracions.
- S'ha lliurat tota la documentació treballada i prevista en l'enunciat de la Pràctica d'Avaluació Continuada 2.

- **3r Sprint - PAC 3, 07/12/20**

Nota: S'ha decidit juntament amb el consultor, canviar l'enfocament del projecte. El projecte inicialment estava concebut com a un projecte de recerca amb la temàtica de representació de fenòmens físics mitjançant processing, la direcció del projecte ha virat cap a un projecte de tipus professionalitzador on recreem peces artístiques utilitzant el mateix llenguatge de programació, però enfocant els resultats a unes visuals perquè els usuaris les puguin utilitzar com a mètode de relaxació.

Resum de les fites:

- S'han elaborat els apartats: procés de treball, APIS utilitzades, perfils d'usuari, seguretat, tests, versions de l'aplicació/servei, requisits d'instal·lació/implantació/ús, instruccions d'instal·lació / implantació, instruccions d'ús.

- S'ha realitzat, aplicant la metodologia establerta per fases utilitzant la metodologia Scrum com a eina per a la planificació i gestió dels terminis de lliurament de les activitats.
- S'han elaborat les 3 primeres peces del projecte.
- S'ha realitzat la importació del codi a la plataforma de distribució.
- S'ha aplicat el feedback rebut per part del consultor a la PAC2.
- S'ha lliurat tota la documentació treballada i prevista en l'enunciat de la Pràctica d'Avaluació Continuada 3.

Procés de creació de les peces:

- **Genèrics:**

Per a dur a terme cada peça s'ha creat una plantilla per tal d'estalviar temps en la codificació dels grafismes. La plantilla base conté les següents línies:

```
/*
Nom:
Descripció:
Controls:
Autor:
    Rafel Bernabeu Aznar
*/
// Variables
boolean x;
float x;
int x;

//Classes
class x{}

//Funcions
void setup(){}

void draw(){}

void funcio(){}

//Interacció amb l'usuari
void mousePressed(){}

void keyPressed(){}

void infoUsuari(){}

```


Com podem veure al codi exposat, la plantilla conté les funcions bàsiques per a la creació de qualsevol programa.

- Primerament i utilitzant els caràcters `/*` i `*/`, tot el text que estigui inclòs entre aquests paràmetres es considera com a un comentari i no és processat per processing com a instrucció. Aquest encapçalament s'utilitza per a definir el nom de l'obra, la descripció, els controls i l'autoria.
- Seguidament a la plantilla podem veure les variables utilitzades, en aquest cas els tipus que hi ha definits són a mode demostració cada obra té les seves pròpies i necessàries per a l'execució correcta de l'aplicació.
- A continuació observem la definició d'una classe que s'utilitza en diferents obres per a crear objectes amb diferents atributs i poder accedir a ells de manera genèrica, sense haver de definir cada vegada l'objecte quan s'utilitza.
- La funció `setup`, obligatòria en qualsevol codi de processing conté els paràmetres inicials de l'aplicació i alguns atributs com poden ser la mida de la pantalla creada, el color del fons i crides a les funcions necessàries per a l'execució del programa.
- La funció `draw`, present a tots els codis de processing, és l'encarregada d'executar el codi relacionat amb la mostra visual, és a dir, qualsevol element gràfic que es vulgui mostrar per pantalla s'haurà de fer la crida en aquesta funció, tant sigui una funció en si o un element propi.
- `mousePressed` i `keyPressed`, són un exemple de funcions relacionades amb events provinents dels perifèrics ratolí i teclat, s'utilitzen per a programar la interacció de l'usuari amb el codi.
- La funció `infoUsuari` s'utilitza durant totes les obres per a mostrar a la pantalla principal les instruccions a seguir per a interactuar amb la peça.

Els arxius de codi de les peces contenen comentaris de les parts més importants del codi per tal de poder modificar o solucionar problemes d'una manera més ràpida, ja que amb els comentaris podem identificar ràpidament les instruccions escrites. A més, els noms propis utilitzats com a variables, funcions i classes identifiquen completament el tipus, valor o execució que representen.

Durant tota la creació de les peces s'ha utilitzat l'API de processing per a consultar qualsevol dubte relacionat amb el codi d'instruccions generat. Podem trobar aquesta informació als enllaços:

- <https://processing.org/reference/>
- <https://processing.org/reference/libraries/>

- **TFG_Galàxia_01:**
 - **Preproducció:**

Aquesta primera obra està basada en un sketch de Bárbara Almeida²⁰, que podem veure al següent [enllaç](#). La temàtica escollida intenta plantejar la immensitat de l'univers i reflexionar sobre el poc que sabem d'aquest. A banda d'utilitzar aquest sketch per a la part més tècnica s'ha fet ús d'imatges de referència publicades a la xarxa com:



Figura 7: Referència Galàxia²¹

- **Producció:**

En aquesta fase el primer que s'ha fet és planificar tots els elements necessaris per a codificar l'obra. Com podem veure al codi font s'ha generat una classe anomenada Planeta per tal d'afegir els atributs necessaris com a les característiques principals d'aquest (posició, mida, angle, color, etc) i poder modificar-les com a objectes individuals. A més s'ha declarat un Array amb la llista dels planetes a generar per a poder controlar la creació d'aquest. Per últim s'ha creat una funció anomenada "crearGalaxia" on podem controlar la quantitat mínima i màxima de planetes que apareixen quan executem la peça. La crida a aquesta funció es realitza a la funció reset() i podem decidir quants planetes podem generar. Finalment després de tenir la peça pràcticament codificada s'han decidit els paràmetres interactius amb l'usuari i s'han generat les instruccions pertinents. En aquest cas, amb un clic de ratolí podem generar una nova galàxia i prement qualsevol tecla podem aturar l'execució i reprendre-la.

- **Postproducció:**

En aquesta fase s'ha revisat el codi per a controlar que no hi havia cap mena d'error i s'han ajustat els colors dels planetes que apareixen quan generem una nova galàxia. Tot i que els colors són aleatoris mitjançant el codi controlem que aquests comptin amb tonalitats paregudes. Aquest factor el controlem mitjançant la variable float "tonalitat". Per últim s'ha importat el codi dins de la plataforma de distribució per a comprovar que funciona correctament.

²⁰ <https://www.openprocessing.org/user/45107>

²¹ <https://www.astrobitacora.com/fusion-de-14-galaxias-inicio-universo/>

- **TFG_Mandala_02:**

- **Preproducció:**

En aquesta fase s'ha realitzat una recerca d'imatges inspirades amb aquests símbols per tal de representar-les mitjançant el llenguatge de processing. La imatge principal d'inspiració ha sigut:

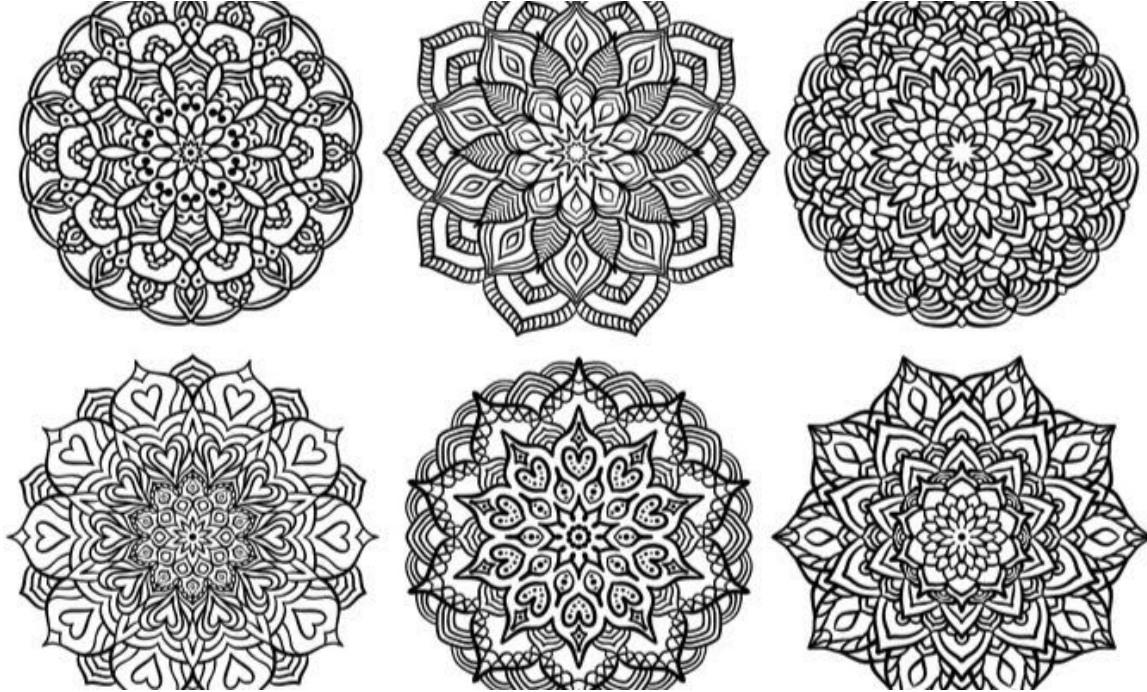


Figura 8: Referència Mandala²²

El plantejament seguit després de la recerca d'imatges era poder replicar símbols pareguts dintre d'una mateixa peça. A la peça, mitjançant la creació d'un nou mandala i els paràmetres interactius amb l'usuari podem generar un nombre infinit de diferents "mandales".

- **Producció:**

En aquesta fase s'ha codificat els elements principals de l'obra, sempre planificant desde un inici els elements necessaris per a poder abordar-la. Com podem observar al codi font s'han utilitzat diferents funcions trigonomètriques com el sinus i el cosinus per a poder controlar la representació visual de les línies a l'espai. A banda, s'han generat variables per a controlar els angles de rotació, els nombres de línies, etc. Pel que fa a la interacció amb l'usuari, com s'indica a les instruccions, aquests poden variar el símbol mostrat per pantalla mitjançant diferents accions com fer clic amb el ratolí per a generar un nou símbol aleatori, utilitzar les fletxes per a canviar el nombre de línies i variar les rotacions d'aquestes.

- **Postproducció:**

Una vegada finalitzada la fase de producció, on obtenim el resultat presentat a l'apartat de prototips, s'han generat nous elements i s'han canviat el color de les línies. Concretament, en aquesta fase, es va decidir finalitzar cada línia amb un punt blanc i donar una tonalitat blava a les línies que es representen. A més, es va comprovar que el codi no contenia cap error per a poder realitzar la importació a la plataforma de distribució correctament.

²² <https://www.aprendemas.com/co/blog/mundo-educativo/por-que-colorear-mandalas-ayuda-al-aprendizaje-75572>

- **TFG_Vòrtex_03:**

- **Preproducció:**

A aquesta fase s'ha realitzat la recerca d'imatges o elements inspiradors per a la reproducció d'un vòrtex. La imatge utilitzada com a font d'inspiració és la següent:



Figura 9: Referència Vòrtex²³

- **Producció:**

A la producció de la peça vòrtex i com podem veure al codi font, la part més representativa del codi és la funció anomenada dibuixarFigura, la qual s'encarrega de representar la figura geomètrica inicial. La interactivitat d'aquesta peça té un pes molt important, ja que ens permet canviar el nombre de figures que mostrem per pantalla i a més podem modificar els costats de la figura geomètrica inicial. Perquè el bucle gràfic no s'aturés s'ha generat un codi específic per a què la direcció que segueix la representació, canvi automàticament de manera inversa, així aconseguim que l'execució de la peça no s'aturi mai. En aquest cas, i com juguem també amb l'opacitat de les figures segons la seva posició a l'espai s'ha decidit deixar el fons negre i les figures blanques per a no intervenir en la visualització correcta dels elements.

- **Postproducció:**

En aquesta fase, s'ha revisat el codi per a confirmar que no hi ha cap error de representació i s'han realitzat ajustos pel que fa a la interacció amb l'usuari. Finalment s'ha importat el codi a la plataforma de distribució canviant solament el paràmetre de la funció "size" per a poder representar la peça a pantalla completa.

²³ <https://ca.wikipedia.org/wiki/V%C3%B2rtex>

- **4t Sprint - Lliurament final, 11/01/21**

Resum de les fites:

- S'han elaborat els apartats: Abstract, abstract (english), projecció a futur, conclusió, annex - lliurable del projecte, annex - bibliografia, agraïments, notacions i convencions, perfils d'usuari, anàlisi de mercat, màrqueting i vendes, annex - vita.
- S'han elaborat les 2 últimes peces del projecte.
- S'ha realitzat la importació del codi a la plataforma de distribució.
- S'ha aplicat el feedback rebut per part del consultor a la PAC3.
- S'ha lliurat tota la documentació treballada i prevista en l'enunciat del lliurament final del TFG

- **TFG_Plexus_04:**

- **Preproducció:**

En aquesta fase s'han buscat les referències principals per a l'execució d'aquesta peça. En aquest cas s'ha buscat inspiració en un plugin utilitzat a l'Adobe After Effects que rep el nom de Plexus. Podem trobar la web d'autoria d'aquest efecte al següent [enllaç](#). A més, s'han utilitzat els següents sketches a tall d'informació per arribar al resultat obtingut:

- <https://www.openprocessing.org/sketch/650668>
- <https://www.openprocessing.org/sketch/772742>

El vídeo que he fet servir com a referència principal és:

- <https://vimeo.com/19947737>



Figura 10: Referència Plexus²⁴

²⁴ <https://motionarray.com/after-effects-templates/plexus-explosion-logo-85943>

- **Producció:**

Pel que fa a la producció d'aquesta peça, la planificació dels elements necessaris ha sigut un factor clau. Com podem veure al codi font s'han programat diferents mòduls com:

- Classe Cercle, on s'han afegit com atributs valors com el radi, la posició a l'espai, el nombre de línies i les velocitats.
- Classe Color, que conté atributs com R,G,B i en funció de les velocitats de cadascun d'aquest creem la variació de color.
- Funció connectar, utilitzada per a unir les línies entre elles tenint en compte els punts d'unió i les rotacions dels elements.
- Funció iniciarRandom, funció encarregada d'executar les ordres principals de la peça per a les generacions aleatòries de l'obra.

- **Postproducció:**

En aquesta fase s'han realitzat ajustos sobretot en respecte als colors. Com a quasi totes les peces, els colors s'elegeixen de forma aleatòria mitjançant la funció random de processing, però sobre aquesta funció s'executen instruccions per tal de controlar els colors dels elements. Amb això podem assegurar que els colors representats són compatibles entre ells i no generen un efecte de saturació impactant a l'usuari. En aquesta peça els colors es representen amb funció de les velocitats que es generen quan creem un nou esbós. Les velocitats també són generades aleatòriament, però de forma controlada i com podem veure al codi font la velocitat mínima està declarada amb un valor de 0,2 i la velocitat màxima amb un valor de 0,8.

Finalment s'ha revisat el codi exhaustivament per a comprovar que no hi ha cap error ni en l'execució local ni en línia.

- **TFG_Nervis_05:**
 - **Preproducció:**

Aquesta obra està inspirada en les connexions nervioses que existeixen als nostres cossos. En aquest cas en concret la imatge escollida com a inspiració fou la següent:

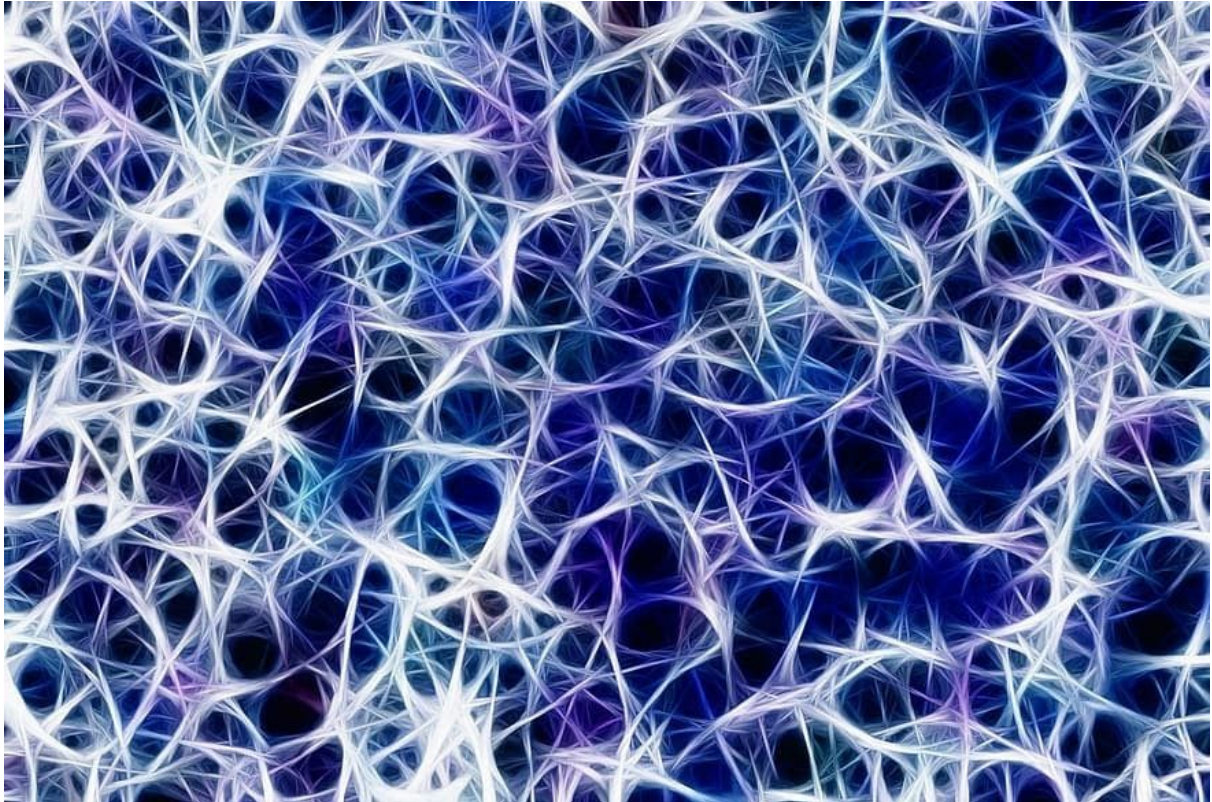


Figura 11: Referència Nervis²⁵

A la peça s'intenta representar el flux de les cèl·lules al sistema nerviós combinant aquestes amb colors agradables visualment.

- **Producció:**

A la fase de producció d'aquesta peça es va planificar amb cura tot el necessari per a representar uns gràfics complexos com els que presenta l'obra. Intuïtivament sabia que per a generar un moviment orgànic necessitava utilitzar la funció noise que ens ofereix processing. Aquesta funció és el factor clau d'aquesta peça tant per al moviment de les partícules com per als colors. Com podem veure al codi font s'han programat elements com ara:

- Classe Vector, que conté atributs com la posició a l'espai i el color. També conté funcions internes com `iniciarParticules()` i `update()` que són les encarregades d'iniciar el sistema de partícules i actualitzar segons la ubicació a l'espai d'aquestes.
- Funció inicialitzar, encarregada de definir els paràmetres com ara els colors, la posició, la velocitat i les partícules.
- Funcions per a la interactivitat amb l'usuari com `mousePressed`, `keyPressed` i `infoUsuari`.

²⁵ <https://www.pikist.com/free-photo-snmitt/es>

- **Postproducció:**

En aquesta fase es van realitzar els últims ajustos en respecte als colors representats utilitzant com a mètode principal el control de l'aleatorietat d'aquests per a trobar combinacions visuals atractives. Finalment s'ha revisat el codi per a la correcta exportació a la plataforma de distribució i el seu funcionament en local.

- **Creació imatges de màrqueting:**

Per a generar les imatges de màrqueting s'ha creat un document d'Adobe Photoshop a manera de plantilla per a reutilitzar el marc i l'objecte de text canviant entre les diferents peces. Per a poder extreure les imatges a la màxima qualitat possible s'ha utilitzat la funció keyPressed:

```
void keyPressed()  
{  
    save("imatge.jpg");  
}
```

A banda, s'ha comentat la funció d'infoUsuari per a no mostrar les instruccions per pantalla i poder guardar els frames sense cap text entrellaçat.

El format de les imatges guardades és de 800 x 800 píxels i .JPG. Una vegada exportades localment totes les imatges s'han importat a la plantilla de Photoshop per afegir el requadre i el text pertinent.

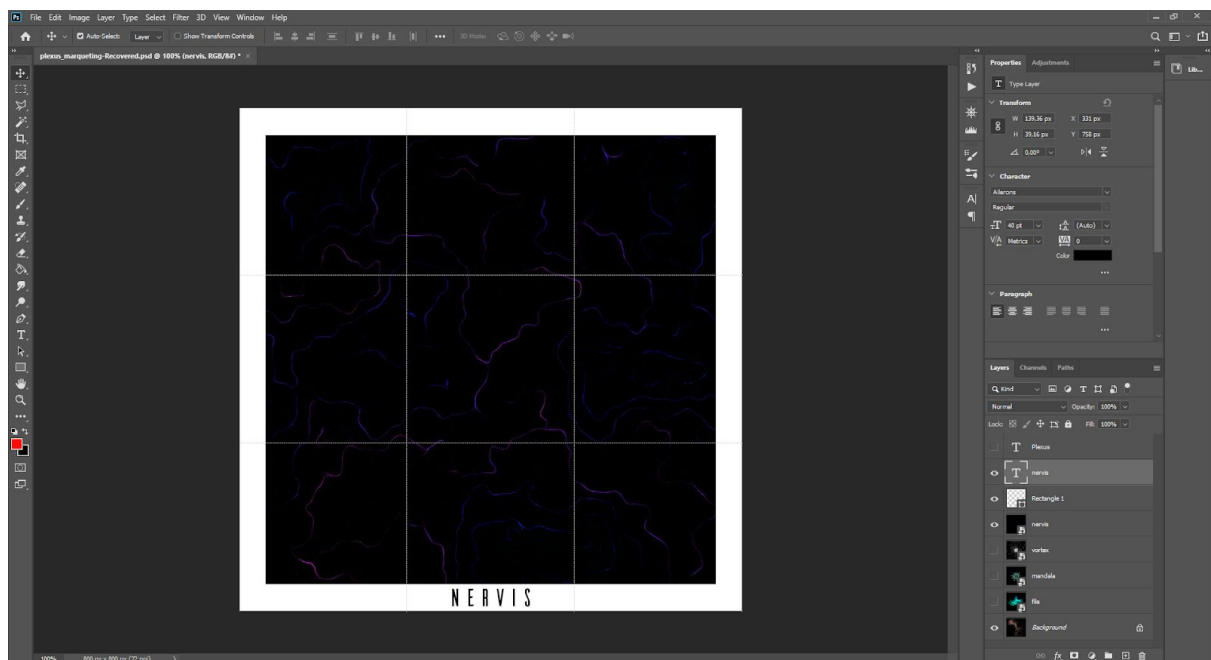


Figura 12: Imatges Màrqueting

Pel que fa a la tipografia escollida del text que apareix a les imatges, es tracta d'Ailerons. S'ha decidit utilitzar aquesta tipografia pel seu estil modern i de pal sec buscant la cohesió dels elements representats a les obres. Podem trobar-la a l'enllaç:

<https://www.behance.net/gallery/25541553/Ailerons-Typeface>

Com podem veure a l'enllaç aquesta compta amb la llicència gratuïta per a projectes d'ús personal.

11. APIS utilitzades

- **PROCESSING:**

Processing²⁶ és un llenguatge de programació que es va crear per a facilitar el desenvolupament de programes visuals donant importància a l'animació i la interacció dels usuaris amb les aplicacions creades. Els desenvolupadors van crear aquest llenguatge per a poder esbossar elements visuals mitjançant el codi. Actualment processing com a llenguatge i la seva comunitat ha crescut de manera exponencial i el llenguatge s'ha utilitzat per a treballs més avançats en l'àmbit de producció a banda de l'opció de generar esbossos.

Originalment el llenguatge estava concebut com a una extensió del llenguatge Java dirigida per a artistes i dissenyadors, a hores d'ara l'evolució d'aquest llenguatge l'ha portat a convertir-se en una eina de disseny i de prototipatge completa que s'utilitza per a molts tipus de projectes, com per exemple, gràfics amb moviment, visualització de dades complexes, visuals vinculades amb el so, etc.

L'objectiu més important d'aquesta API és fer accessible la programació per a un públic molt ample, és per això que l'eina es pot descarregar gratuïtament, utilitzar-se de la mateixa forma i està basada en software lliure.



Figura 13: Web Processing

- **OPENPROCESSING:**

Openprocessing²⁷ és una comunitat de programadors creatius que busca l'experimentació amb codi, el disseny, l'art i qualsevol cosa que envolti aquestes. Aquesta plataforma permet programar directament al navegador i en el cas concret d'aquest projecte s'ha utilitzat per a comprovar el correcte funcionament del codi i essencialment per a canviar el paràmetre `size(800,800)`; utilitzat en local per a crear la finestra de l'aplicació per `size(window.innerWidth, window.innerHeight)`; perquè la peça ocupi la totalitat de la pantalla sense tenir en compte la resolució concreta d'aquesta.

Aquesta plataforma compta amb diferents opcions d'interactuació, entre les quals destaquen:

- **Discover:**

Aquesta pàgina ens permet buscar entre tots els projectes de la comunitat existents filtrant els resultats per nom del creador, likes rebuts, hashtags utilitzats i a més per la seva temporalitat, és a dir, filtrar els resultats per mes, any o qualsevol moment.

- **Learn:**

²⁶ <https://processing.org>

²⁷ <https://www.openprocessing.org/>

En aquesta pàgina podem trobar diferents tutorials, des de guies d'iniciació a tutorials més complexos per a generar projectes d'alt nivell. Aquests tutorials estan creats per usuaris destacats de la plataforma i són una eina perfecta per aprendre.

- Teach:

En aquesta secció la plataforma ofereix un entorn per al professorat, on aquests poden impartir classes als alumnes basats en aquest llenguatge. Compta amb opcions per a generar tutorials i controlar l'accessibilitat del contingut creat per a les classes.

A més openprocessing ofereix l'opció de subscripció per tenir avantatges com:

- Més protecció de visualització del codi generat a la plataforma.
- Opcions personalitzables per a transportar les creacions a altres plataformes.
- Utilitzar llibreries externes a les existents.
- Emmagatzematge al núvol per a recursos utilitzats en els programes.
- Creació il·limitada d'esbossos.

Amb el pla gratuït de la plataforma podem consultar les diferents pàgines destacades, utilitzar l'entorn de programació en línia per als nostres esbossos i interactuar mitjançant comentaris o likes amb els altres usuaris de la comunitat.

En aquest projecte s'ha utilitzat la plataforma com a inspiració per alguna de les obres generades i per a la distribució de les peces finals.

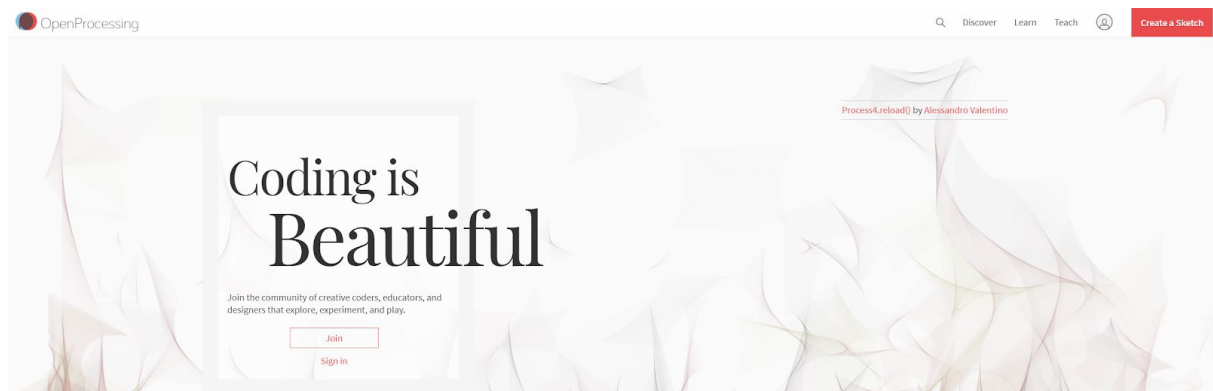


Figura 14: Web OpenProcessing

12. Diagrames UML

A la següent imatge podem veure el diagrama genèric que s'ha utilitzat per a la realització de les diferents peces del projecte:

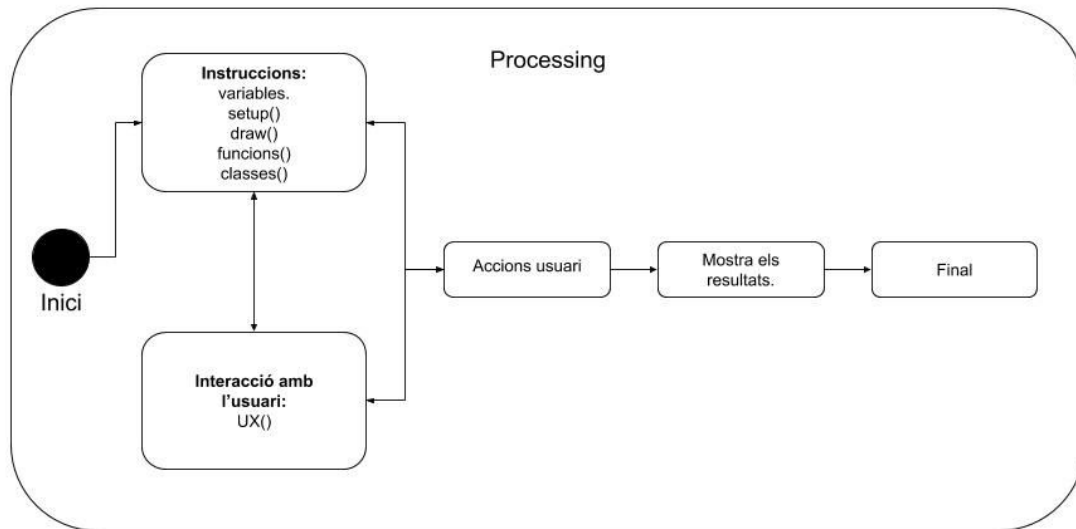


Figura 15: Diagrama UML

Primerament en iniciar el programa s'executen les diferents instruccions programades, entre aquestes trobem:

- Declaració de les variables globals i locals utilitzades en les diferents funcions.
- Funció setup en la qual s'estableixen els paràmetres de llançament de l'aplicació.
- Funció draw en la qual s'inclou el codi per a poder mostrar per pantalla els diferents gràfics creats.
- Funcions addicionals per a intervenir amb les classes i funcions.
- Classes per a definir els atributs i els mètodes dels objectes.

Al requadre d'interacció podem veure les funcions relacionades amb les accions de l'usuari davant l'aplicació. En les obres generades podem veure diferents tipus d'interacció, principalment centrades en els events provinents del ratolí o el teclat. Finalment quan l'usuari realitza l'acció podem veure el resultat per pantalla.

A la següent imatge podem veure el diagrama UML de l'exemple concret de la peça TFG_Galàxia_01:

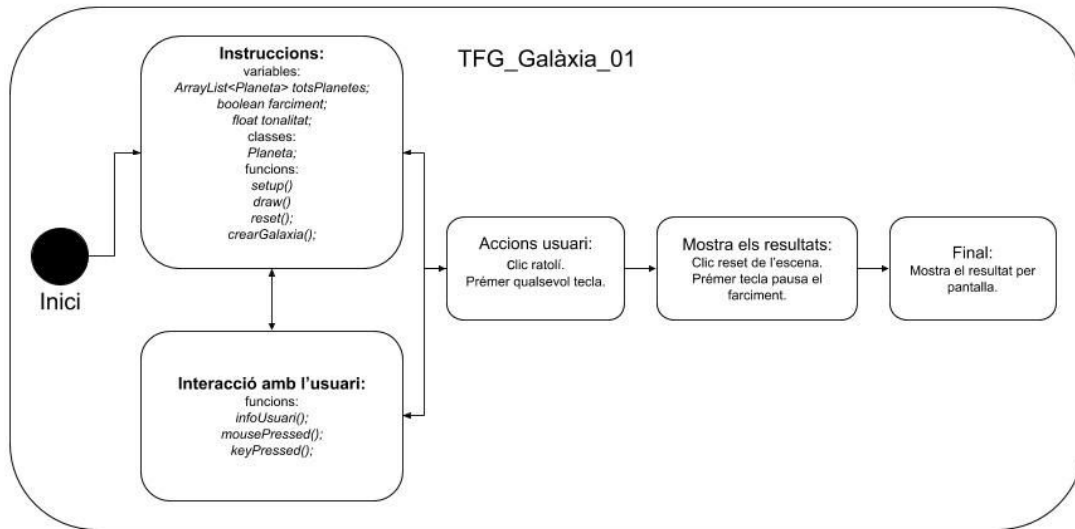


Figura 16: Diagrama UML TFG_Galàxia_01

Com podem veure a la imatge anterior, totes les peces exposades segueixen el diagrama presentat com a genèric. Aquests diagrames faciliten la planificació dels elements necessaris per a assolir la complexitat dels programes generats.

En el requadre d'instruccions podem trobar les variables, les classes i les funcions necessàries per al funcionament de la peça. Al requadre d'interacció amb l'usuari trobem tant la informació de les instruccions per a l'usuari com les funcions amb els events d'interactivitat amb la peça.

13. Prototips

A continuació adjunto els primers esbossos generats per ordinador de les peces creades.

- **SKETCHES:**
 - TFG_Galàxia_01:

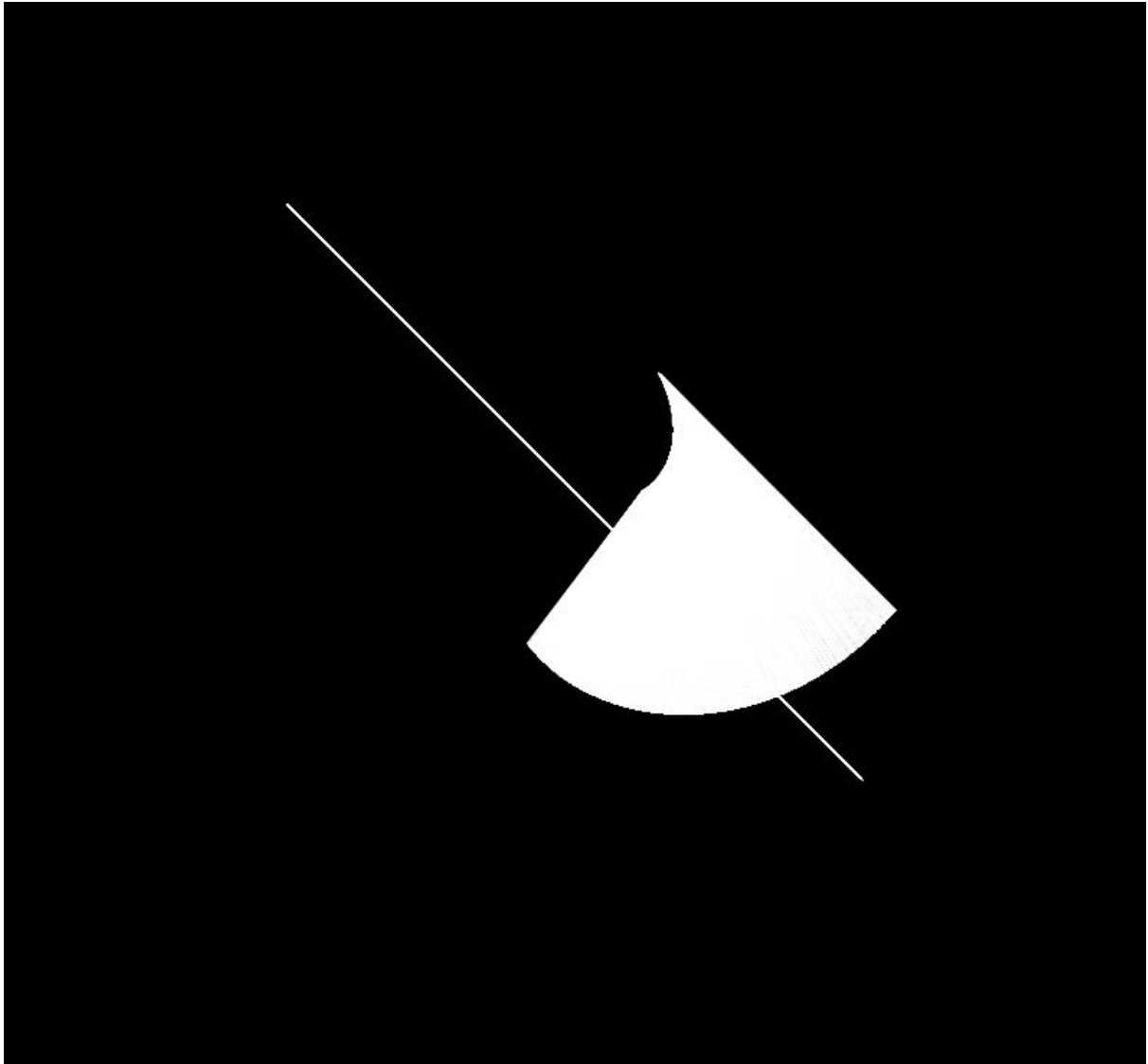


Figura 17: Esbós TFG_Galàxia_01

- TFG_Mandala_02:

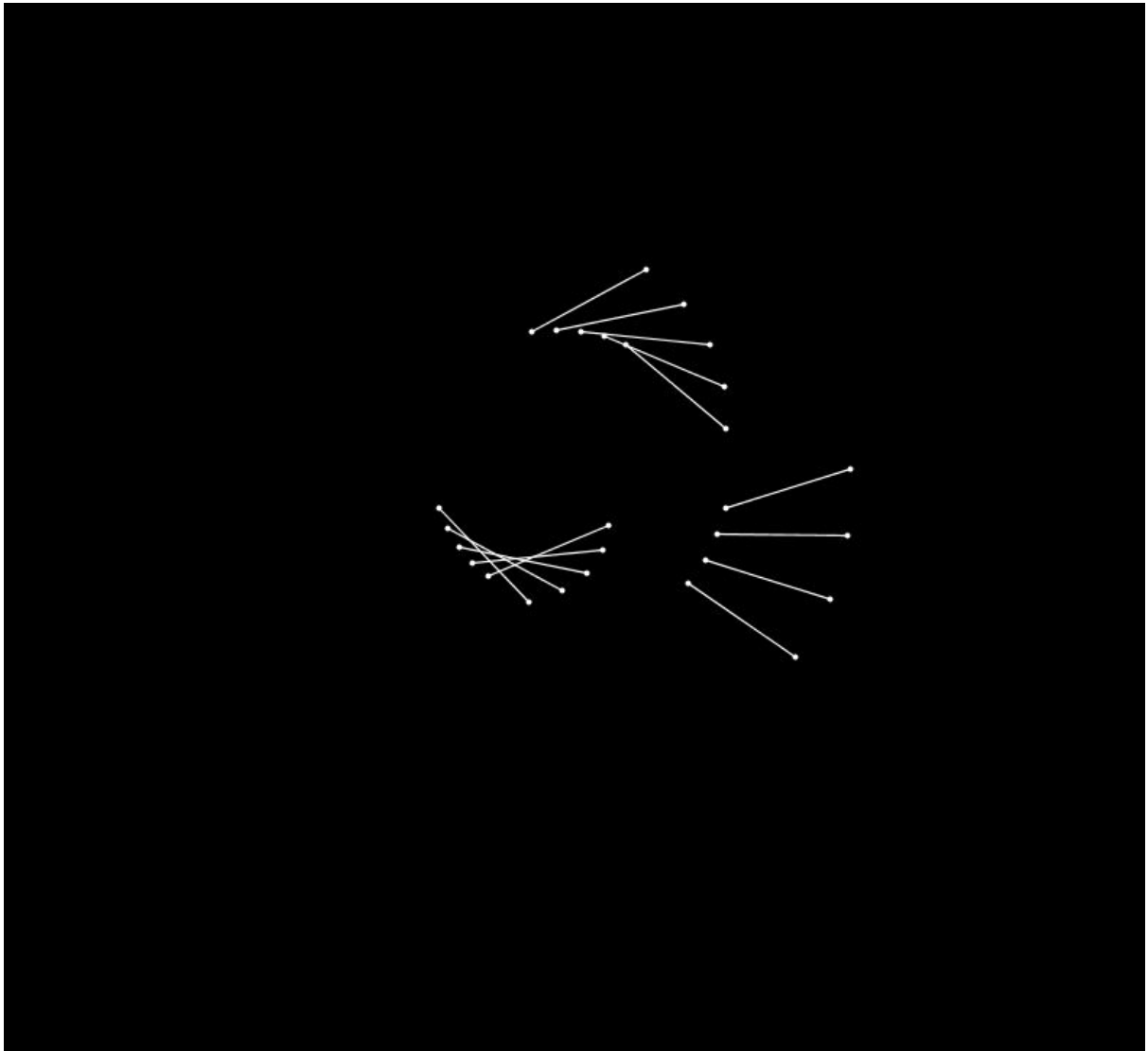


Figura 18: Esbós TFG_Mandala_02

- TFG_Vòrtex_03:

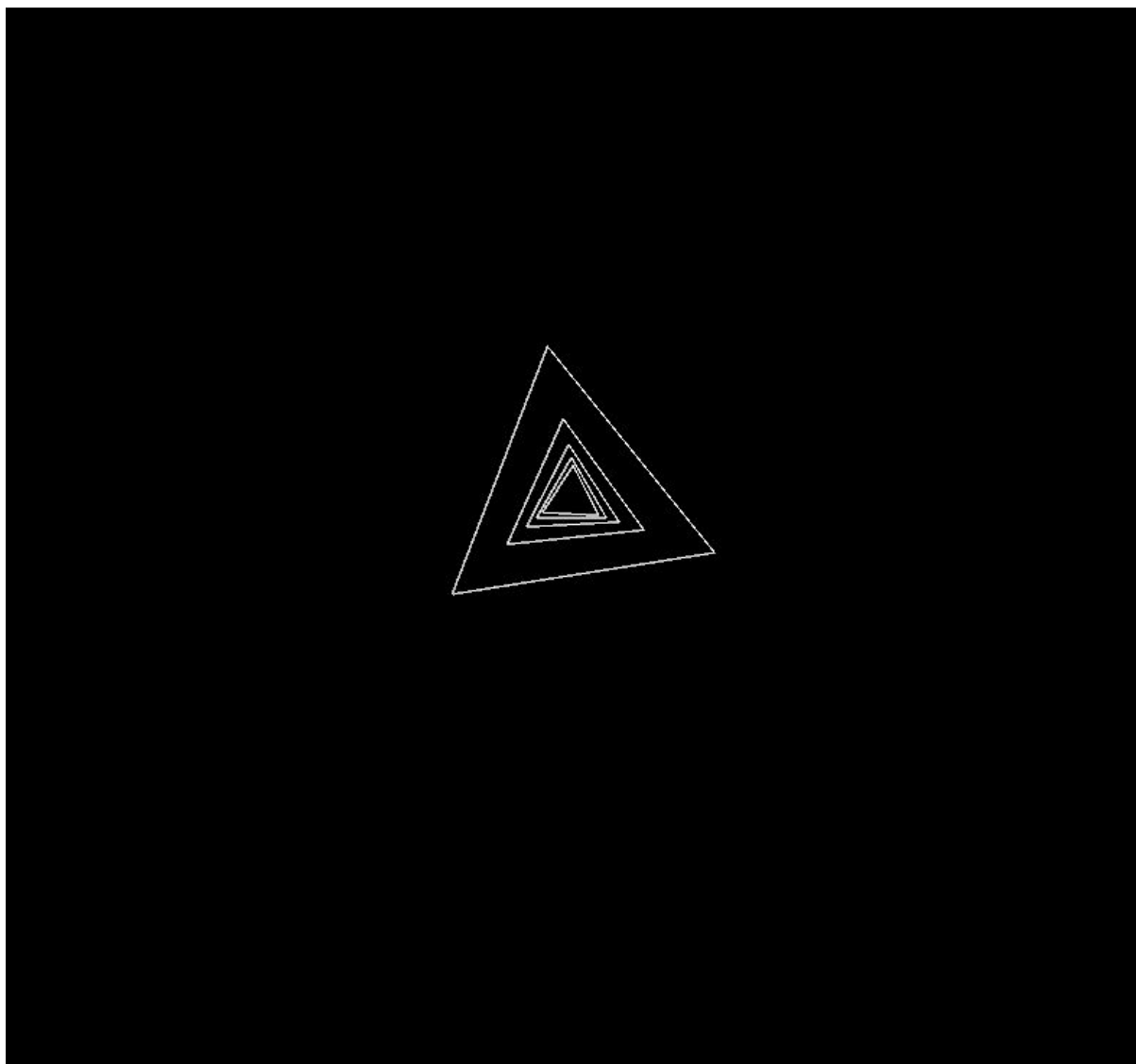


Figura 19: Esbós TFG_Vòrtex_03

- TFG_Plexus_04:



Figura 20: Esbós TFG_Plexus_04

- TFG_Nervis_05:

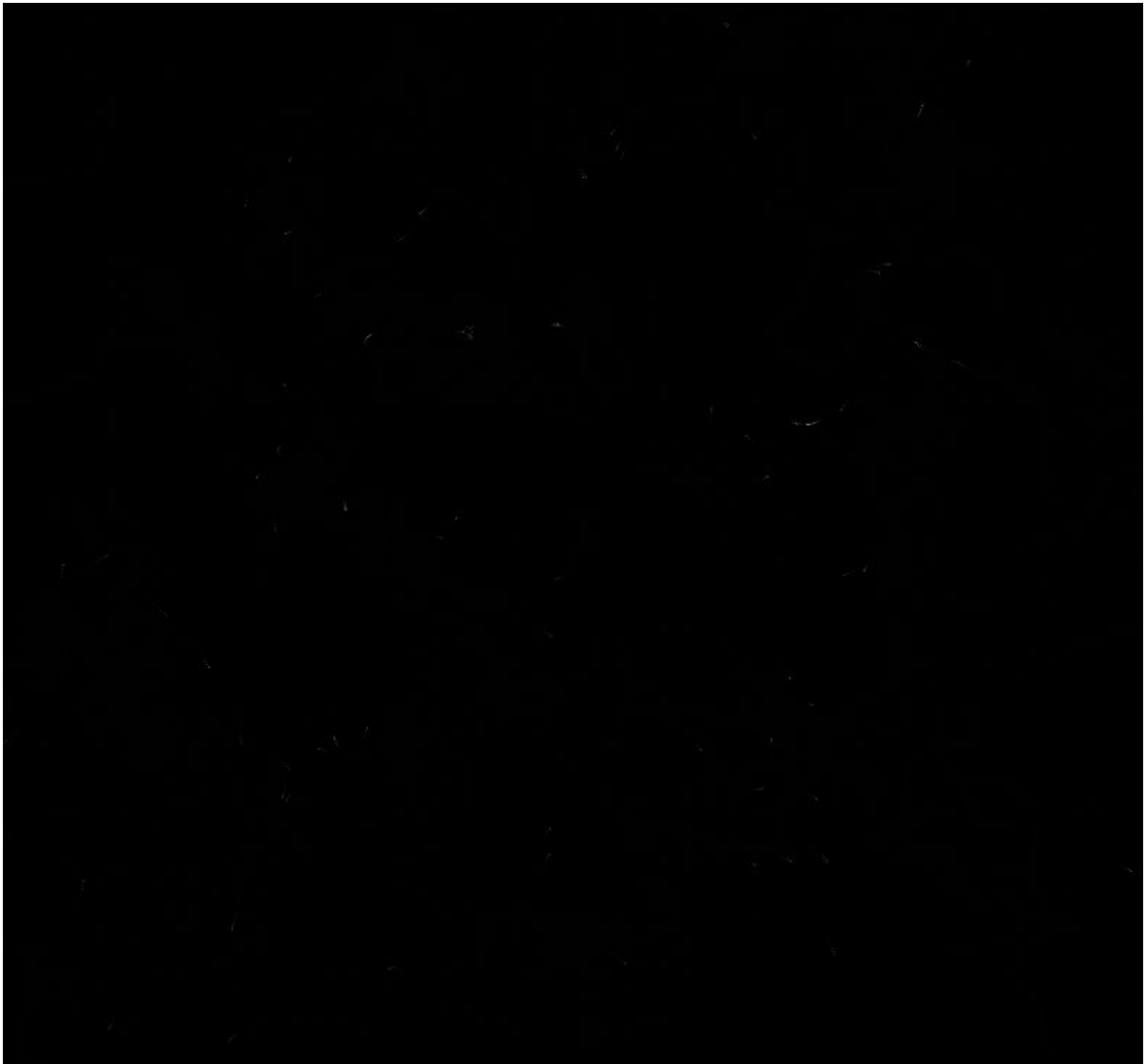


Figura 21: Esbós TFG_Nervis_05

Com hem pogut observar a les imatges anteriors els prototips presenten una idea inicial de com serà l'obra final. Normalment sempre utilitzo un fons completament negre amb els elements a representar en blanc per a poder diferenciar clarament com actuen les instruccions programades a les demostracions. Amb la metodologia de treball aplicada aquests primers esbossos s'extreuen entre el final de la fase de preproducció i l'inici de la fase de producció.

14. Usabilitat/UX

Les interfícies generades per a les peces contenen les instruccions perquè l'usuari interactuï amb aquestes. Bàsicament són text on es presenten les instruccions perquè l'usuari pugui canviar valors de manera aleatòria però controlada dels paràmetres.

Per a poder dur a terme aquesta interfície, cada peça té associada una funció, descrita a l'apartat anterior, on es mostra per pantalla un text amb les instruccions a seguir.

Exemple text:

```
void infoUsuari()  
{  
    textSize(11);  
    textAlign(CENTER);  
    fill(255);  
    String frase = "Fer clic a qualsevol lloc per a obtenir una nova  
galaxia.\n";  
    text(frase, width/2, 30);  
}
```

Exemple gràfic:

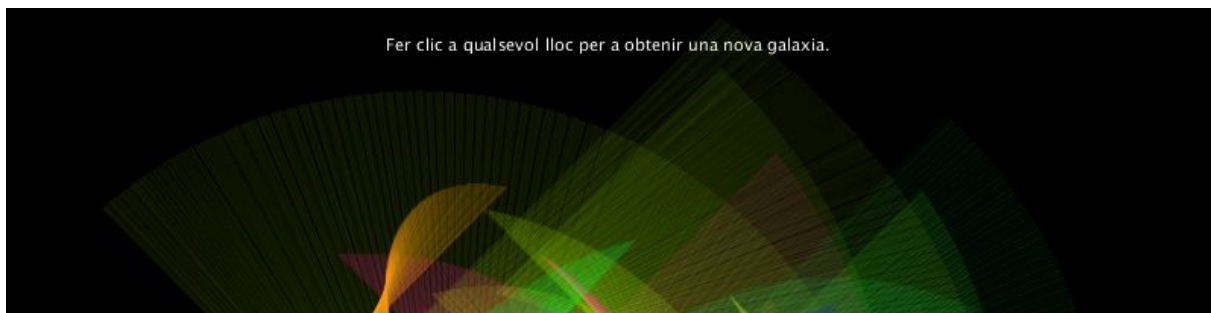


Figura 22: Exemple TFG_Galàxia_01

Com podem veure a la imatge anterior i fent referència al codi exposat:

- `textSize` s'utilitza per a definir la mida del text.
- `textAlign` s'utilitza per a definir l'alineació en respecte a la dimensió de la finestra.
- `fill()` s'utilitza per a definir el color del text.
- `String` s'utilitza per a guardar la variable frase amb el text que podem veure a la imatge.
- `text` s'utilitza per a fer la mostra per pantalla, on es passen com a atribut el text, i la seva posició a la pantalla.

La plataforma d'openprocessing ens permet de manera automàtica afegir instruccions a les aplicacions creades:

My Sketch

TITLE

DESCRIPTION

HOW TO INTERACT WITH IT | e.g. "mouse, keyboard"

TAGS

LICENSE Attribution ShareAlike learn more about [Creative Commons](#) [get Plus+](#) to change the license

DRAFT ☒

WHO CAN SEE YOUR SKETCH? Anyone Only Me [get Plus+](#) to change privacy settings

WHO CAN SEE THE CODE? Anyone Only Me [get Plus+](#) to hide the source code

WHO CAN COMMENT? Anyone Only Me

Figura 23: Exemple Web OpenProcessing

En l'apartat "How to interact with it" podem escriure el text amb les instruccions a seguir per l'usuari. S'ha decidit mostrar aquesta informació amb el codi natiu de processing per a poder controlar que la informació estigui sempre present durant l'execució del codi, ja que si utilitzem aquesta opció de la plataforma, el text desapareix pocs segons després.

Els factors per a la mostra d'aquesta informació s'han basat en:

- Simplicitat, utilitzar un mètode que no utilitzi massa recursos per a centrar la totalitat dels recursos en les representacions de les peces.
- No molestar, mida de la lletra petita i en blanc perquè no tingui un pes excessiu en la composició.
- Directe, missatges curts per a transmetre d'una manera molt simple les instruccions.
- Adaptar el format i el contingut per a la correcta execució en qualsevol pantalla d'ordinador.

15. Perfils d'usuari

Per a poder assolir l'objectiu principal del projecte cal definir el públic objectiu o target d'aquestes obres. Concretament el públic objectiu que pugui fer ús d'aquestes peces amb l'objectiu de relaxar-se o realitzar petites meditacions. Si aconseguim ajustar aquests perfils, evitarem dirigir les obres a un públic global i trobarem als futurs consumidors d'aquest producte per tal d'aconseguir l'objectiu.

- Perfil demogràfic:

Aquest projecte dirigeix les obres a persones que treballen directament a l'ordinador durant moltes hores. S'ha decidit restringir el projecte a aquest tipus de perfil, ja que són els usuaris potencials de l'objectiu principal d'aquest treball, aquest tipus de perfil podrà fer ús de les obres en un període relativament curt de temps i aconseguir realitzar una meditació curta o aclarir la ment. L'objectiu, com he comentat abans, és que amb les obres generades puguin deixar la ment en blanc i puguin relaxar-se. Per tant els podríem definir com a persones que treballen tant en l'àmbit com el tècnic utilitzant l'ordinador com a eina principal. L'edat del públic objectiu serà dels 18 als 65 anys, que són els anys en els quals se sol desenvolupar l'activitat professional.

Com les instruccions mostrades a les aplicacions estan explícitament creades per al projecte, en aquest cas, ens dirigim a públic de llengua catalana. Canviant aquestes instruccions a l'anglès podríem dirigir-nos a un públic més global.

- Perfil sociocultural:

El nivell econòmic del públic objectiu és mig/alt, però al mateix temps també podem considerar-lo com a no requerit, ja que aquest projecte inicialment s'ha creat sense ànim de lucre, més bé com a una obra social. El gènere serà tant femení com masculí, actualment l'ordinador s'utilitza arreu del món en infinitat de sectors sense importar el gènere de la persona que l'utilitza.

- Perfil digital:

El perfil digital del públic objectiu d'aquest projecte principalment seran usuaris que s'interessen per l'art digital i amb coneixements prou alts sobre la xarxa. Normalment aquest usuari serà actiu en les principals xarxes socials on també podrà trobar les obres o compartir-les.

Les paraules clau amb la que podria trobar les obres serien: generative art, art meditation, processing art, relaxing art, hypnotic animation, relaxing animation.

16. Seguretat

En aquest projecte, en tenir distribució en línia, la seguretat depèn completament de la plataforma de distribució, en aquest cas www.openprocessing.org. Com podem veure, en aquest enllaç <https://www.openprocessing.org/home/tos> la plataforma està regida per les lleis pròpies del país d'origen.

Pel que fa a les obres generades amb el pla gratuït ofert per la plataforma ens permet realitzar algunes accions relacionades amb la seguretat i la privacitat de les obres com qui pot comentar en la teva publicació o el tipus de llicència de l'obra. Les llicències que ens permet atribuir a l'obra són:

- No creative commons license.
- Attribution.
- Attribution shareAlike.
- Attribution no derivatives.
- Attribution non-commercial.
- Attribution non-commercial shareAlike
- Attribution non-commercial no derivatives.

En el cas del projecte en concret, i com estem parlant d'un projecte que de manera inicial no s'ha originat per a lucrar-se, les obres es distribuïran amb la llicència Attribution non-commercial.

Per a poder aconseguir un grau més alt de seguretat i privacitat en les obres hauríem d'afiliar-se a la membresia plus que ofereix la plataforma amb una quota de \$39,99 a l'any. Amb aquesta subscripció podríem establir diferents paràmetres com qui pot veure les obres, qui pot veure el codi, distribuir les obres de forma privada o pública, publicar obres sense límits, etc.

17. Test

Els tests que s'han realitzat s'han dut a terme a l'ordinador on s'ha acomplert el projecte. Processing permet executar el codi generat i observar a la consola si hi ha algun tipus d'error. Aquest procés és clau per a la correcta execució de les obres generades i també per a evitar problemes greus com bucles infinits o una necessitat de recursos de processament exagerada.

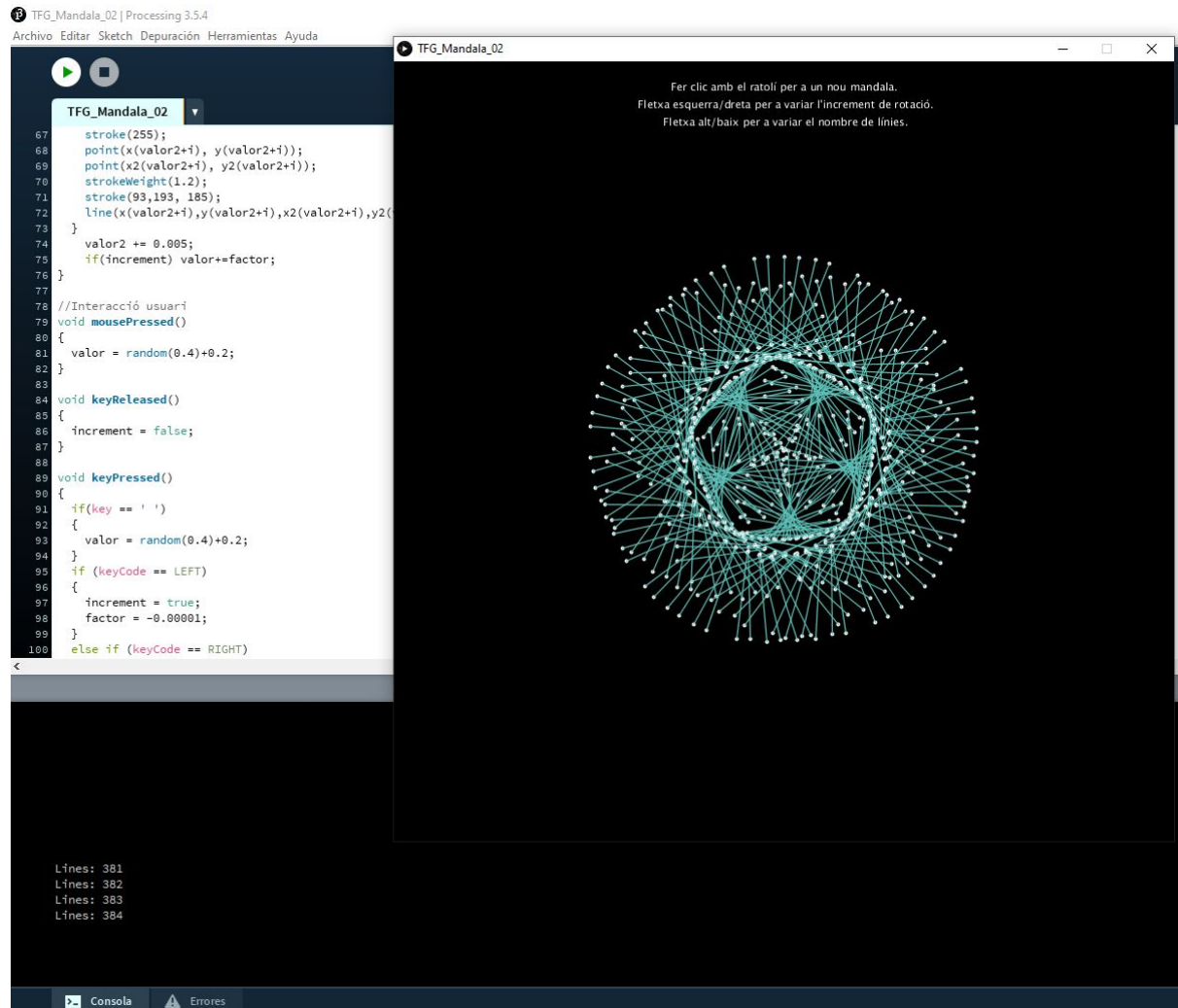


Figura 24: Exemple TFG_Mandala_02

Com podem veure a la següent captura amb el botó de play situat a la part superior esquerra executem el codi escrit i ens apareix la finestra amb l'obra. A la part inferior podem veure la consola, on en aquest cas mostra un paràmetre de la interacció de l'usuari. Si hi hagués qualsevol error de compilació tant la consola com la pestanya d'errors ens mostrarien el missatge de l'error per tal de facilitar-nos la detecció d'errors d'escriptura.

Una vegada l'obra està finalitzada en local, al meu ordinador, el següent test és publicar-ho a la plataforma de distribució d'openprocessing per a comprovar que el programa funciona correctament i la seva visualització és correcta. A diferència d'en local el paràmetre les mides de la funció size són canviades de:

```
size(800,800);  
a  
size(window.innerWidth, window.innerHeight);
```

Aquest canvi s'aplica perquè l'aplicació s'adapti a qualsevol mesura de monitor, sense tenir en compte les característiques tècniques d'aquest. Mitjançant aquest canvi, aconseguim representar l'obra a pantalla completa.

El test final es realitza a la plataforma de distribució amb la finalitat de comprovar la correcta execució del codi:

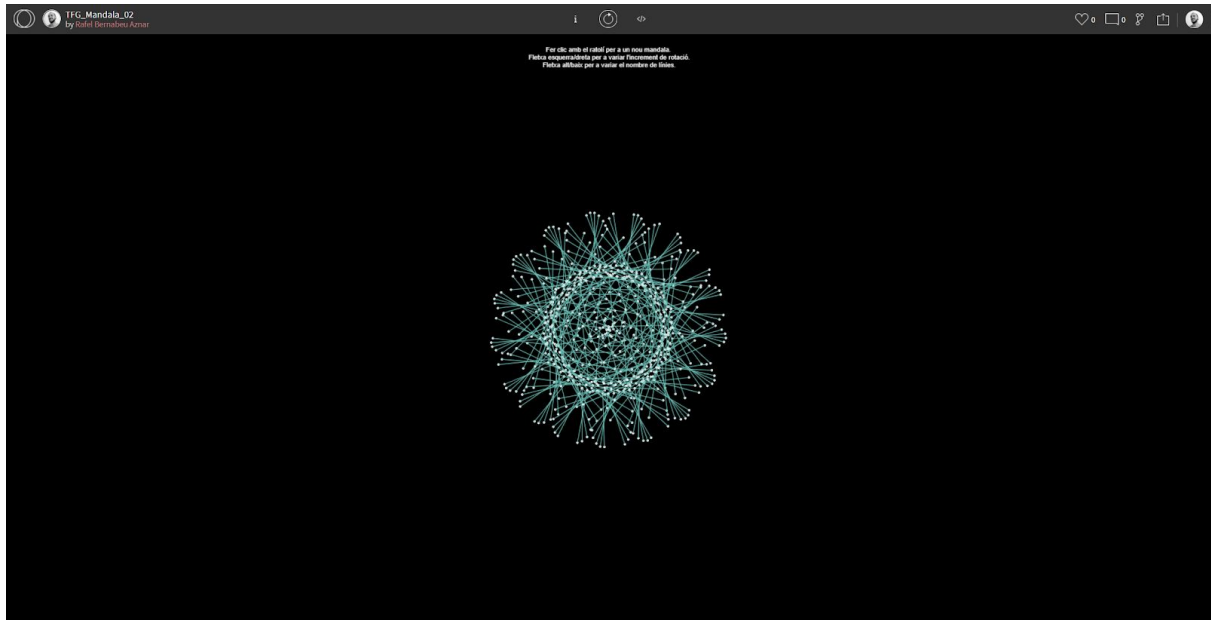


Figura 25: Exemple test TFG_Mandala_02

18. Versions de l'aplicació/Servei

Com hem vist a la metodologia de treball, en aquest projecte podem extreure almenys dues versions de cada obra generada. Per una banda, podem obtenir la versió de producció on estan definits quasi tots els paràmetres de l'àmbit visual de la peça. Per altra banda, també obtenim la versió de postproducció de l'obra on es realitzen ajustos finals creatius i de codificació. Un exemple d'aquestes diferents versions és:

- Exemple producció:

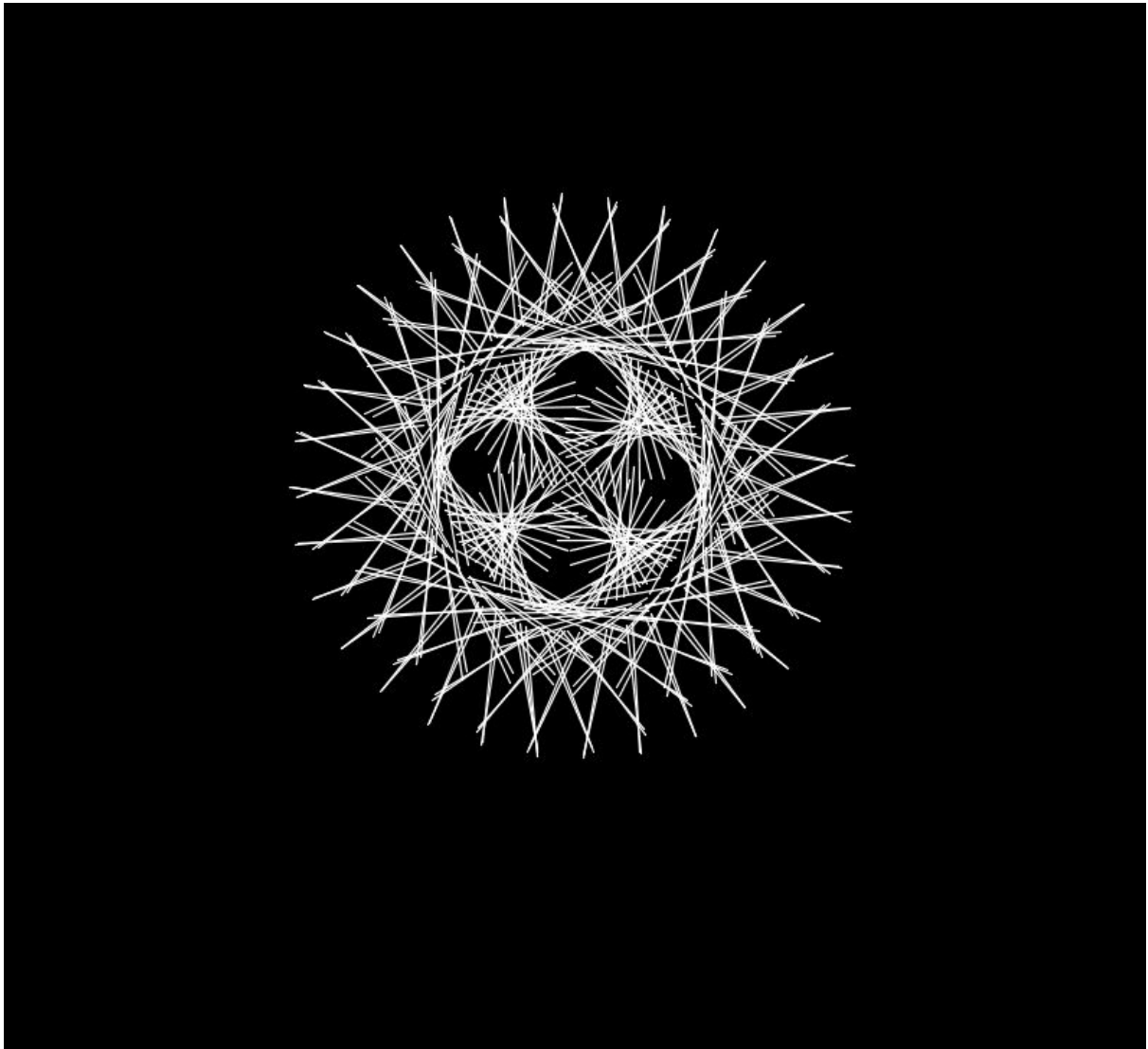


Figura 26: Exemple producció TFG_Mandala_02

En aquest punt, com podem veure a la imatge anterior, la peça generada ja contempla totes les instruccions al codi per al seu funcionament. A la següent fase, postproducció, afegirem les tonalitats de color, els punts al final de les línies per a buscar més riquesa visual i per últim les instruccions d'interactivitat per a l'usuari.

- Exemple postproducció:

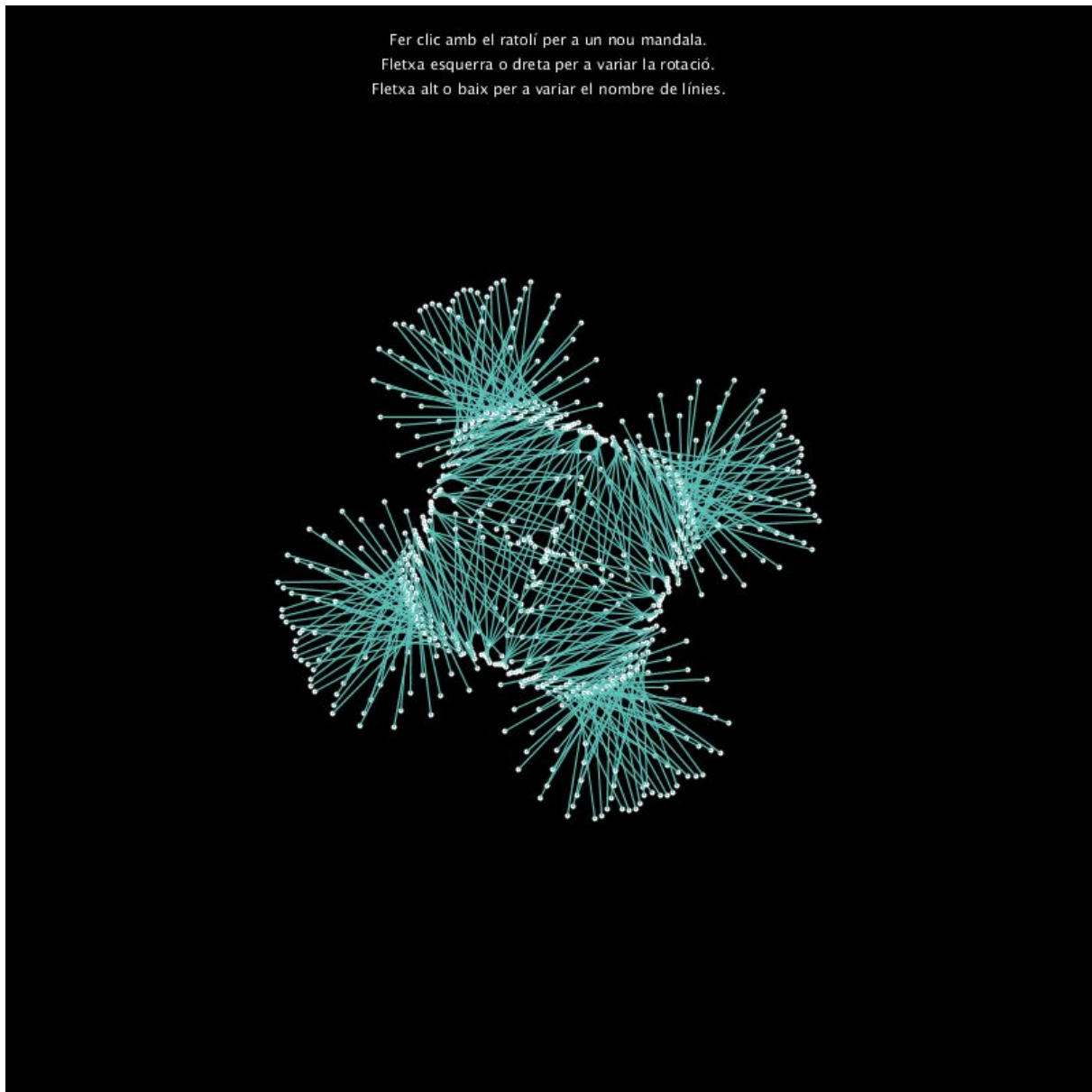


Figura 27: Exemple postproducció TFG_Mandala_02

Com podem observar a la imatge anterior, en aquest punt podem donar per finalitzada aquesta fase i sols quedaria la tasca d'importació a la plataforma de distribució.

Els elements que sofreixen canvis, com el color, els ajustos de codi, les velocitats, les rotacions, el nombre de partícules, el nombre de línies i altres elements, varien segons la peça específica.

Aquestes dues fases, més la fase de preproducció i la fase de distribució, han sigut executades per a la totalitat de les obres, metodologia la qual m'ha permès mantenir un ordre durant el desenvolupament del projecte i conèixer els estats de cada una de les peces.

19. Requisits d'ús

La plataforma distribuïdora (www.openprocessing.org) ens marca els requisits principals per a l'ús de les obres. En aquest cas, com es tracta de compilacions mitjançant el navegador els requisits mínims dels usuaris, a banda de disposar d'un equip, serien:

Sistemes operatius compatibles:

- Microsoft Windows.
- Mac OS
- Linux
- Android
- iOS

Navegadors compatibles:

- Internet Explorer
- Microsoft Edge
- Firefox
- Chrome
- Safari
- iOS Safari
- Opera
- Opera Mini
- Navegador Android

20. Instruccions d'instal·lació/implantació

Per aquest projecte no és necessari realitzar cap instal·lació, ja que la plataforma de distribució s'encarrega de poder executar el codi generat mitjançant el navegador. Per a poder executar el codi generat en local a la web la plataforma <https://www.openprocessing.org/> utilitza JavaScript combinat amb HTML per a interpretar les ordres creades amb processing.

L'esquema general d'una pàgina web és un document HTML on trobem les diferents etiquetes d'aquesta. L'estructura bàsica d'una web HTML compta amb les etiquetes de `<head>` i `<body>` és en aquestes on podem trobar l'execució dels scripts per a la representació visual d'aquests. Hi ha diferents maneres d'afegir el codi al document HTML. Per una banda podem escriure el codi directament al document HTML, però no és la manera òptima, ja que per a una millor organització es recomana separar el codi HTML del codi JavaScript. Per altra banda, podem incloure codi JavaScript de manera externa, és a dir, utilitzar el llenguatge per a fer referència a l'arxiu que volem executar.

La ubicació de l'etiqueta script pot estar situada en diferents etiquetes dintre del codi HTML, per exemple pot estar ubicada dins de l'etiqueta `<head>` la qual cosa provocarà que la pàgina web descarregarà l'arxiu abans de començar a dibuixar el contingut de la pàgina. També podem trobar l'etiqueta script dins de l'etiqueta `<body>`, on l'execució del codi es realitzarà després de carregar la pàgina.

A banda de l'expressat abans també podem tenir en compte el mode de càrrega d'un script, ja que aquest pot fer-se de manera asíncrona o diferida. De manera asíncrona el nostre script s'executarà tan prompte com sigui possible i de manera diferida ajornarà l'execució del script quan la pàgina estigui carregada.

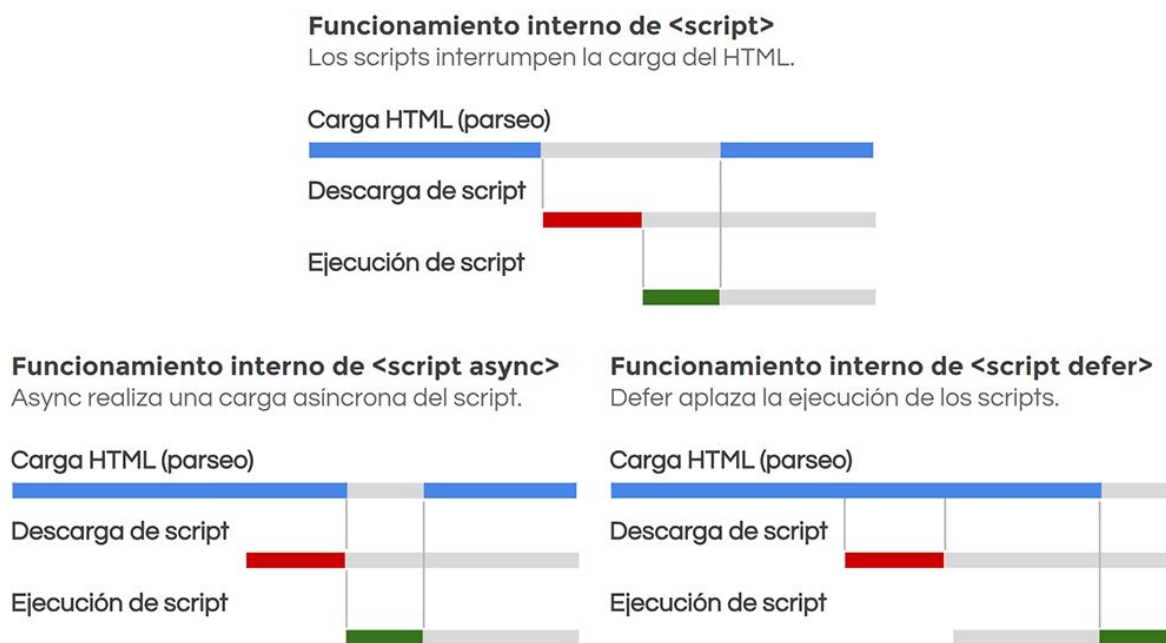


Figura 28: Càrrega script²⁸

En el cas de la web d'openprocessing, si fem una inspecció del codi font, podem veure com l'etiqueta script apareix tant a `<head>` com a `<body>`, factor que ens indica que segons el tipus de codi a executar els programadors web decideixen quan es realitza la càrrega d'aquest.

²⁸ <https://lenguajehtml.com/html/scripting/etiquetas-html-scripts/>

```

<!DOCTYPE html>
<html class="layout_default owner existingSketch js flexbox canvas canvastext webgl no-touch geolocation postmessage websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers no-applicationcache svg inlinesvg smil svgclippaths nonMember" data-1t-installed="true">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <script type="text/javascript" src="https://bam-cell.nr-data.net/1/48adb45b7a?&a=427819410&v=1184.ab39b52&t_gation%22:%7B%7D%70&fp=708&at=5xZRF1sY5E0&30&jsonp=NREUM.setToken"></script>
    <script src="https://js-agent.newrelic.com/nr-1184.min.js"></script>
    <script type="text/javascript" async src="https://www.google-analytics.com/gtm/js?id=GTM-WX8BJLV&cid=1533528003.1602172723"></script>
    <script async src="https://www.google-analytics.com/analytics.js"></script>
    <script type="text/javascript"></script>
    <!-- META TAGS -->
    <meta name="description" content">
    <meta property="fb:admins" content="844361">
    <meta property="fb:app_id" content="113458952018100">
    <meta property="og:url" content="https://www.openprocessing.org/sketch/1046332/">
    <meta property="og:title" content="TFG_Galàxia_01 - OpenProcessing">
    <meta property="og:description" content">
    <meta property="og:type" content="video.other">
    <meta property="og:image" content="https://www.openprocessing.org/sketch/1046332/thumbnail">
    <meta property="og:image:secure_url" content="https://www.openprocessing.org/sketch/1046332/thumbnail">
    <meta property="og:image:width" content="200">
    <meta property="og:image:height" content="200">
    <!-- TWITTER -->
    <meta name="twitter:card" content="summary">
    <meta name="twitter:site" content="@openprocessing">
    <meta name="twitter:title" content="TFG_Galàxia_01">
    <meta name="twitter:description" content">
    <meta name="twitter:image" content="https://openprocessing-usercontent.s3.amazonaws.com/thumbnails/visualThumbnail1046332@2x.jpg">
    <!--

```

Figura 29: <Head>

```

<!DOCTYPE html>
<html class="layout_default owner existingSketch js flexbox canvas canvastext webgl no-touch geolocation postmessage websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers no-applicationcache svg inlinesvg smil svgclippaths nonMember" data-1t-installed="true">
  <head></head>
  <body class="">
    <div id="app" class="playMode showInstructions"></div>
    <!-- <div class="icon icon_fullscreen"></div> -->
    <div class="modals"></div>
    <!-- empty and fake div to allow hack sketches work -->
    <div id="appletAjaxContainer"></div>
    <!-- commented to load local
    script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script -->
    <script></script>
    <!-- VUE -->
    <script src="/assets/js/vendor/node_modules/vue/dist/vue.min.js"></script>
    <script src="/assets/js/vendor/node_modules/vue-tippy/dist/vue-tippy.min.js"></script>
    <script src="/assets/js/op_components.js"></script>
    <!-- Sentry -->
    <script src="/assets/js/vendor/node_modules/@sentry/browser/build/bundle.min.js" crossorigin="anonymous"></script>
    <script src="/assets/js/vendor/node_modules/@sentry/integrations/build/vue.min.js" crossorigin="anonymous"></script>
    <script src="/assets/js/vendor/node_modules/@sentry/integrations/build/extraerrordata.min.js" crossorigin="anonymous"></script>
    <script></script>
    <script src="/assets/js/vendor/jquery-1.11.1.min.js"></script>
    <script src="/assets/js/vendor/modernizr-2.6.2.min.js"></script>
    <script src="/assets/js/vendor/bootstrap.min.js"></script>
    <script src="/assets/js/vendor/node_modules/moment/min/moment.min.js"></script>
    <script src="/assets/js/vendor/node_modules/moment-timezone/builds/moment-timezone-with-data.min.js"></script>
    <script src="/assets/js/vendor/node_modules/underscore/underscore-min.js"></script>
    <script src="/assets/js/vendor/jquery.toTextArea.js-master/jquery.toTextArea.js"></script>
    <script src="/assets/js/vendor/jquery.unveil.js"></script>
    <script src="/assets/js/vendor/node_modules/d3/dist/d3.min.js"></script>
    <script src="/assets/js/vendor/jquery.form.min.js"></script>
    <script src="/assets/js/vendor/node_modules/popper.js/dist/umd/popper.min.js"></script>
    <script src="/assets/js/vendor/node_modules/tippy.js/umd/index.all.min.js"></script>
    <script src="/assets/js/openprocessing.js?version=11.13"></script>

```

Figura 30: <Body>

Analitzant el codi HTML podem veure com l'execució del nostre script es produeix quan la pàgina s'ha acabat de carregar, per tant el codi s'executa a l'etiqueta <body>. A més podem trobar aquesta instrucció:

```
<script defer="" src="/assets/js/sketch/sketch_view.js?version=11.13"></script>
```

La qual és l'encarregada de mostrar visualment l'execució per pantalla del nostre codi. L'atribut `defer`²⁹, s'encarrega d'ajornar l'execució del script quan el navegador hagi descarregat tot el necessari.

Gràcies a la plataforma de distribució tots aquests aspectes tècnics on es mesclen llenguatges de desenvolupament web i el codi natiu de les nostres aplicacions queden en un segon pla i ens faciliten la representació de les obres creades, ja que sols s'ha hagut d'importar el codi en una finestra de creació de sketch, emmagatzemar-lo a la plataforma i executar-lo sense necessitar cap mena d'instal·lació ni de coneixements avançats de llenguatges com HTML i JavaScript.

²⁹ <https://lenguajehtml.com/html/scripting/etiquetas-html-scripts/>

21. Instruccions d'ús

Les instruccions d'ús de cada peça, estan representades mitjançant una funció dintre del codi. Com he comentat abans, la plataforma de distribució ens permet afegir informació per a l'usuari amb les indicacions corresponents per a la interacció d'aquest amb el codi. En el cas de les obres plantejades al projecte s'ha decidit afegir les instruccions en mode de text dintre de l'obra perquè si utilitzem l'opció que ens permet la plataforma, aquestes desapareixen en un període de temps molt curt.

La funció que s'ha creat per a les instruccions dels usuaris es presenta de la següent manera:

```
/*
    Mostrem els controls de la interacció de l'usuari a la part inferior de la
    pantalla.
*/
void infoUsuari() {
    textSize(11);
    textAlign(CENTER);
    fill(255);
    String frase = "Fer clic a qualsevol lloc per a obtenir una nova
galaxia.\n";
    text(frase, width/2, 30);
}
```

En totes les obres generades la funció es denomina de la mateixa manera i la crida a aquesta es realitza en la funció principal `setup()`;

Com podem veure a la vinent imatge, les instruccions per a l'usuari quedarien impreses damunt dels gràfics generats mitjançant el codi i a la part superior de l'obra.

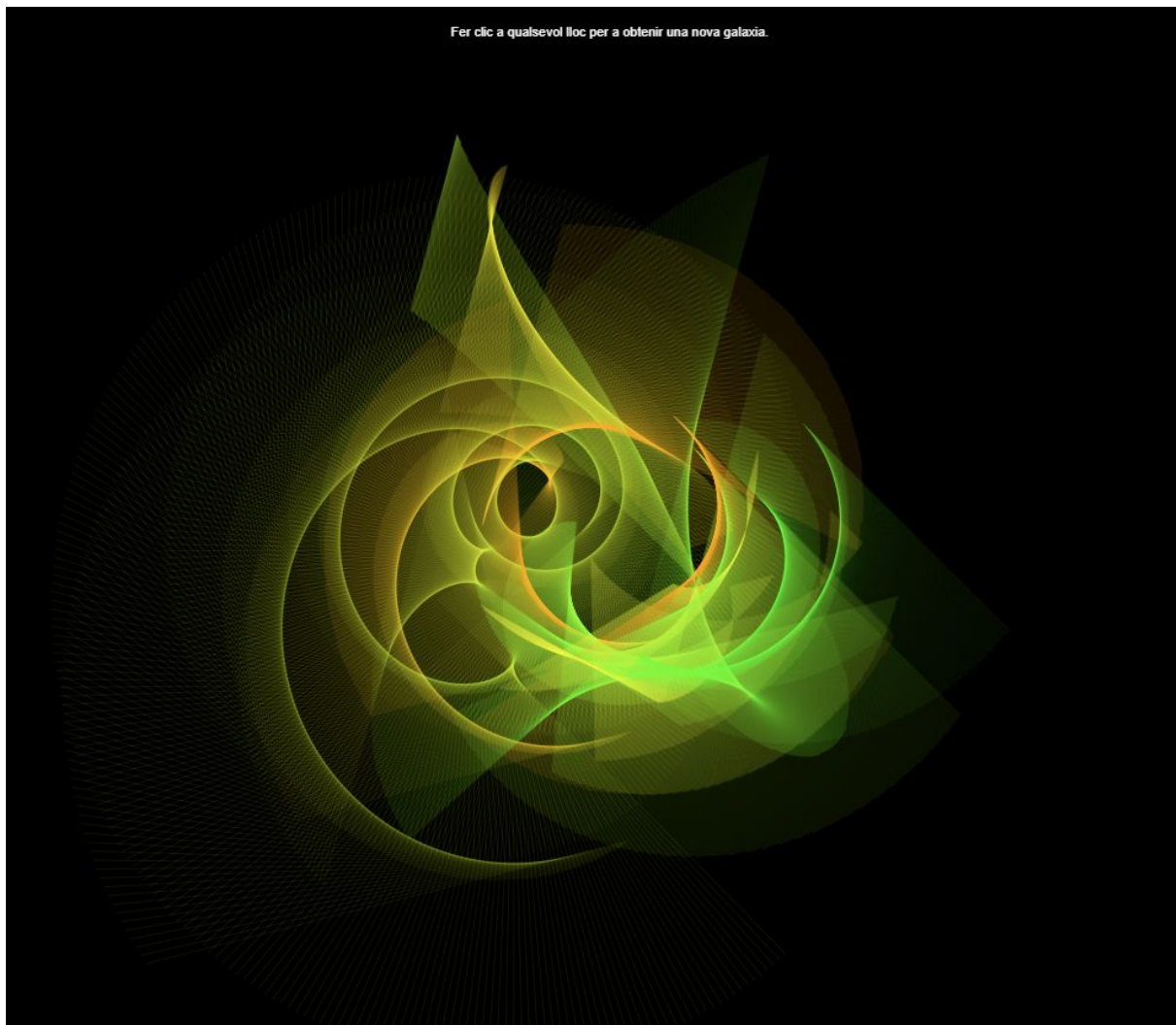


Figura 31: Instruccions

Aquest patró es repeteix a la totalitat de les obres per tal d'unificar les instruccions als usuaris tenint en compte que aquestes sempre apareguin al mateix lloc i amb la mateixa tipografia, per a donar cohesió al projecte.

22. Projectió a futur

L'objectiu d'aquest TFG ha estat dissenyar i crear diferents peces amb programació creativa per a buscar que els consumidors o usuaris d'aquestes puguin relaxar-se o utilitzar-les per a meditar. Inicialment l'enfocament del projecte està pensat sense ànim de lucre, intentant que aquest servei s'entengui com a una obra social i així poder crear a la volta una comunitat d'usuaris.

A escala general podríem esmentar diferents canvis per a maximitzar la seva difusió a la xarxa, com per exemple:

- Canviar l'idioma de les instruccions a l'anglès per a dirigir-nos a un públic més global.
- Canviar el format de les peces (actualment en .PDE) a format de vídeo i/o imatge per a poder difondre les obres generades a les xarxes socials.
- Crear més obres pensant en temàtiques concretes per a crear cohesió en els conjunts de les peces.
- Millorar els aspectes visuals de les existents buscant un estil propi perquè els usuaris ens puguin identificar fàcilment.

Pel que fa a la projecció a futur d'aquest projecte centrant-nos en poder extreure una rendibilitat d'aquest es podria enfocar de diferents maneres:

- Web pròpia:

Actualment la distribució de les obres es realitza mitjançant la plataforma d'openprocessing, amb la creació d'una web pròpia on exposarem les plataformes podríem buscar extreure una rendibilitat del projecte. Primerament utilitzant eines de publicitat existents a la xarxa podríem afegir anuncis a les pàgines referenciant productes relacionats amb la meditació o la relaxació. També s'hauria d'afegir una subscripció per a usuaris prèmium on l'usuari tingués més contingut i alguns avantatges d'avant dels usuaris convencionals.

Les peces a desenvolupar per a la plataforma web estarien agrupades per diferents temàtiques a desenvolupar com per exemple, l'univers, símbols, fluids, etc. D'aquesta manera creant diferents obres sobre una temàtica concreta l'usuari podria escollir quina li apeteix consumir segons les seves preferències. Una altra característica que aportaria molt al projecte seria afegir meditacions guiades mitjançant el so, tant música de fons com narració afegida per a guiar als clients entre diferents relaxacions o meditacions.

- Conversió de la web en aplicació per a dispositius mòbils:

Un altre format que ens ajudaria a buscar una rendibilitat al projecte podria ser la creació d'una aplicació per als dispositius mòbils, que en els temps actuals són el producte on les persones consumeixen més contingut diàriament. Aquesta aplicació tindria les mateixes característiques que la web exposada abans, però afegint el gran grau de mobilitat que ens permeten els dispositius mòbils.

- Web d'exposicions artístiques i aprenentatge:

Aquest format es basaria en crear una comunitat d'artistes on cadascun pugues mostrar les seves obres amb un estàndard de qualitat definit i una temàtica marcada, centrada en la relaxació o la meditació. Aquesta plataforma permetria poder oferir diferents serveis com ara:

- Venda de còpies físiques dels frames escollits pels usuaris:

Com estem treballant en productes gràfics per ordinador resultaria molt fàcil implementar mitjançant el codi que l'usuari pugues seleccionar un frame escollit de la peça que més li agradi per tal d'imprimir-la i poder obtenir una còpia física d'aquesta per a utilitzar com a decoració.

- Tutorials d'aprenentatge per a la creació d'aquest tipus de peces:

Pot ser, alguns dels consumidors s'interessin per la generació d'aquest tipus de peces, és per això que la plataforma podria oferir unes guies d'aprenentatge ben definides explicat el pas a pas del codi perquè l'usuari pugui generar les seves pròpies peces.

Actualment podríem entendre el projecte com a una prova inicial per a veure l'acceptació del públic. Si aconseguirem una atenció prou elevada, hauríem de plantejar quina opció de les exposades abans tindria més possibilitats per tal d'aconseguir un producte viable. L'evolució natural per a poder validar les idees per a fer la prova del producte serien:

- Creació d'una marca pròpia per a poder identificar els nostres productes.
- Creació de perfils a les xarxes socials més importants (Facebook, Twitter, Instagram, tiktok, etc.)
- Distribució de les peces a la xarxa.
- Invertir en publicitat per a la mostra de les peces.
- Interactuar amb comunitats d'usuaris que mediten o s'interessen en mètodes de relaxació.

Una vegada realitzades aquestes proves s'extraurien les dades de l'impacte generat i es valoraria quina de les possibles solucions proposades seria la més viable. Basant-se en la interacció dels usuaris a les xarxes socials i definint aquest tipus d'usuaris per a poder perfilar el tipus de client que s'interessa amb el producte.

23. Pressupost

Elaboració del projecte "Programació Creativa" per a distribuir a través del lloc web.

PREU TOTAL: 7.550,00 € + IVA

UNITATS	DESCRIPCIÓ	PREU HORA	HORES	PREU TOTAL
1	Documentació del projecte: Recerca, materials, documentació necessària, catalogació.	24,00€	50	1.200,00€
1	Escollir peces, desenvolupament d'idees.	24,00€	50	1.200,00€
1	Correcció del projecte i textos.	24,00€	20	480,00€
10	Realització de les peces d'art.	35,00€	100	3.500,00€
1	Canvis dels materials (imatges, gràfics, vídeo).	24,00€	10	240,00€
1	Exportacions codi.	24,00€	5	120,00€
1	Importacions codi a la plataforma.	24,00€	5	120,00€
1	Exportació imatges per màrqueting.	30,00€	15	450,00€
1	Optimització materials. (Debug)	24,00€	10	240,00€
TOTAL				7.550,00€

Taula 7: Pressupost del projecte.

*Preu sense IVA.

24. Anàlisi de mercat

S'ha dut a terme una cerca a la xarxa d'altres plataformes relacionades amb l'objectiu principal del TFG on comparteixen alguna de les idees que presentem en aquest projecte.

- <https://heartwork.app/>:
 - Apple Store: <https://apps.apple.com/app/heart-work/id1380764163>

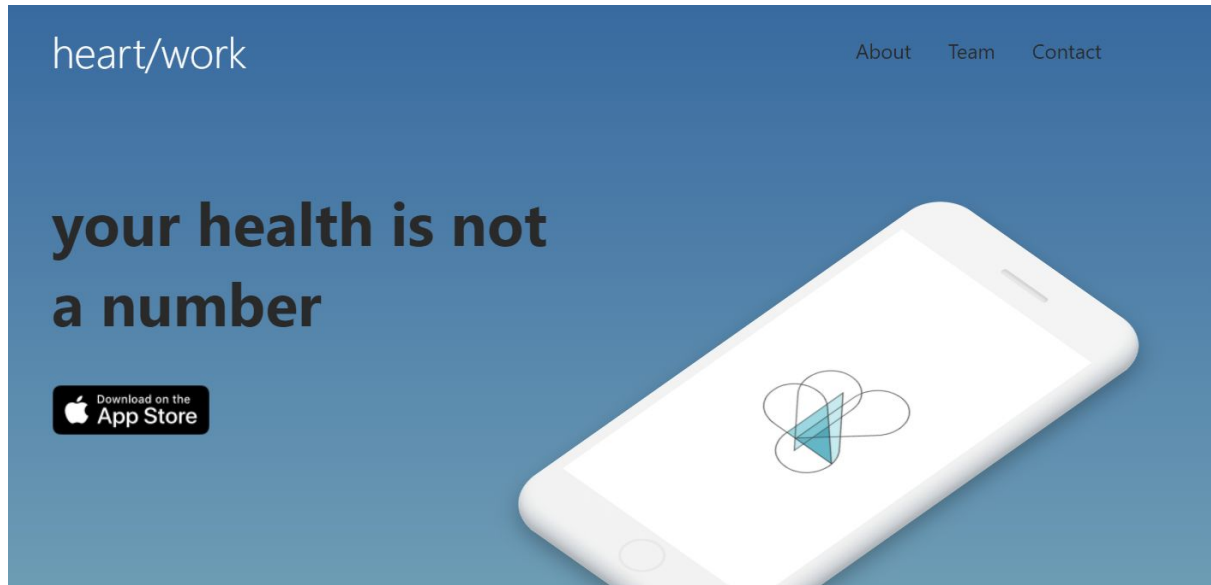


Figura 32: Imatge heart/work

Aquesta aplicació per a dispositius mòbils genera obres d'art basades en la informació de salut de l'usuari. Permet generar peces d'art generatiu enllaçant els batecs del cor quan realitzem diferent tipus d'activitats, com per exemple la meditació. Utilitza paràmetres com ara els batecs del cor en l'activitat o en repòs, la variabilitat de la freqüència cardíaca, nombre de passos realitzats, distància per a córrer o caminar, escales pujades, energia utilitzada, clima, ubicació, estat d'ànim i preferències de l'usuari com el color favorit per a generar una obra artística molt personalitzada.

Com podem veure tant a la web pròpia com a l'aplicació l'estil de disseny que segueix és minimalista i molt simplificat. Utilitza els colors amb poca saturació i basats en les tonalitats blanques per a representar tota la interfície.

- <http://weavesilk.com/>:
 - Apple Store: <https://apps.apple.com/es/app/silk-2-generative-art/id1050339928?mt=8&ign-mpt=uo%3D4>

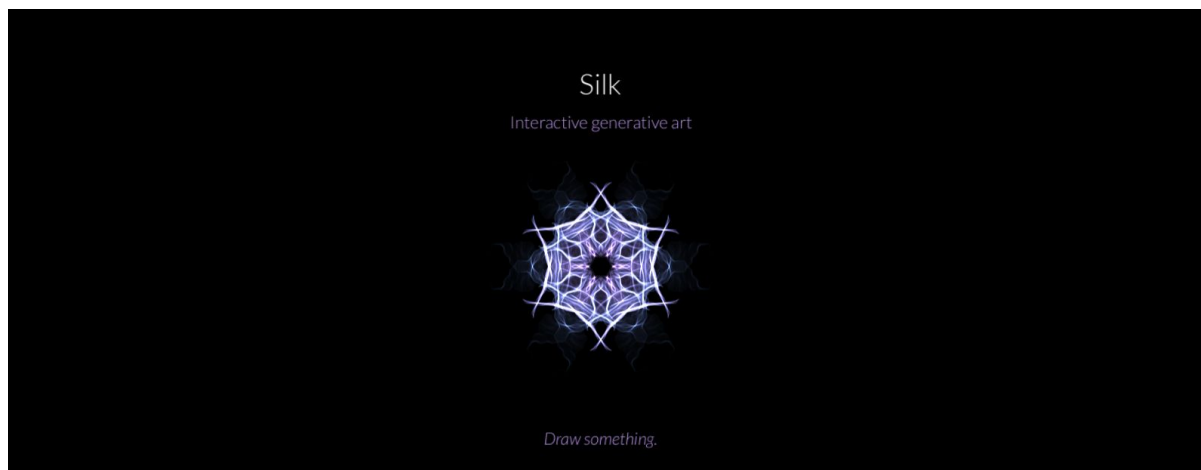


Figura 33: Imatge Silk2 Generative Art

Aquesta aplicació / web permet als usuaris generar peces d'art a qualsevol usuari sense tenir cap mena de coneixement artístic. Com podem veure a les obres d'exemple els pinzells que utilitza i les metodologies de dibuix estan basats en símbols com el mandala, ja que quan utilitzem l'eina el pinzell presenta simetries en diferents eixos que ens permeten generar obres perfectament simètriques en qualsevol pla.

Com podem veure a la descripció de l'aplicació els desenvolupadors inciten a la relaxació i a l'expressió per a la creació de fons de pantalla. A banda de l'aplicació també podem utilitzar-la mitjançant qualsevol navegador, permet exportar les peces creades en alta qualitat i compartir-les en qualsevol xarxa social.

Com podem observar tant la web com l'aplicació mostren una interfície molt simple per tal de centrar l'atenció en l'obra creada sense distreure a l'usuari en moltes opcions. Mantenen en ambdues plataformes un disseny minimalista i simplificat fins a l'extrem.

Anàlisi DAFO

S'ha aplicat aquest mètode de planificació estratègica per tractar d'avaluar les debilitats, les amenaces, les fortaleeses i les oportunitats d'aquest projecte.


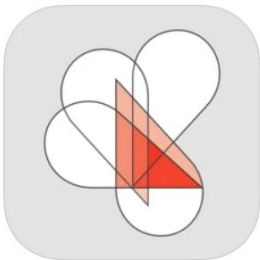
	Positius	Negatius
Origen Intern	Fortaleeses <ul style="list-style-type: none"> • Missatges visuals amb impacte positiu. • Transmetre propostes visuals amb valor afegit (relaxació, meditació). • Extreure de les peces generades imatges com a obres d'art. • Sentiment de marca potent vinculat amb els usuaris. 	Debilitats <ul style="list-style-type: none"> • Estat inicial del projecte. • Complexitat dels processos de desenvolupament. • Adaptació als diferents mètodes de reproducció (dispositius o navegadors).
Origen	Oportunitats	Amenaces

Extern	<ul style="list-style-type: none"> • Mercat amb poca competència • Vincle fort amb els usuaris • Innovació en les tècniques de meditació existents 	<ul style="list-style-type: none"> • Temps de càrrega de peces molt complexes • Disseny amigable per a l'usuari • Manca d'àudio de les peces.
--------	---	--

Taula 8: DAFO

Mapa de posicionament

En el mapa de posicionament amb la competència existent s'ha avaluat l'ús del disseny i les opcions de configuració de les peces creades permeses als usuaris.

+Disseny			
- Opcions usuari	 <p>Silk 2 - Generative art</p>	 <p>Heart/work</p>	+Opcions usuari
	Programació Creativa		
-Disseny			

Taula 9: Mapa de posicionament

Com podem veure al mapa de posicionament, actualment i per culpa de l'estat inicial del projecte davant d'aquestes aplicacions el nostre projecte estaria posicionat amb molt desavantatge davant de les aplicacions existents. Com s'ha comentat a l'apartat anterior, el projecte s'utilitzarà com a prova per a veure si és capaç de generar interès suficient en els usuaris per a abordar les propostes de projecció a futur. A més, la projecció de futur d'aquest projecte té diferents vessants les quals comparteixen alguns elements amb les aplicacions analitzades però amb moltes diferències de conceptes i visualització de la idea de negoci.

25. Màrqueting i Vendes

El projecte de Programació Creativa consta d'una sèrie d'obres visuals amb la finalitat de què els usuaris les utilitzin per a poder relaxar-se. Com s'ha comentat durant tota la memòria, aquest projecte inicialment està enfocat sense cap ànim de lucre, simplement com a una prova per a veure l'interès que poden arribar a generar aquest tipus de peces i si la seva finalitat s'assoleix per part dels usuaris. De totes maneres es podrien utilitzar els materials generats fins ara, amb un canvi de format, com a part d'una primera versió de recursos gràfics bàsics per a utilitzar com a màrqueting.

Aquest canvi de format comptaria amb dos variants diferents, primerament per tal de distribuir les peces a les xarxes socials, s'han escollit frames amb gran impacte visual per a la seva distribució a la xarxa, a les publicacions anirà adjunt l'enllaç perquè l'usuari pugui interactuar amb aquestes. Una altra forma de distribució seria generar demostracions d'aquestes peces en format vídeo a màxima qualitat, però es perdria la interacció de l'usuari i tal vegada no aconseguiríem que aquest entrés directament a l'enllaç de les obres.

La pauta seria presentar les peces a la xarxa amb un disseny de presentació coherent entre elles per tal de buscar una identitat de marca del projecte i així poder diferenciar-se d'altres productes existents, utilitzant-les com a reclam per a accedir a la plataforma de distribució on actualment s'ubica el projecte.

S'han preparat les imatges més impactants de cada obra, generant un requadre blanc que envolta l'obra amb la paraula clau de la interpretació d'aquesta i sense afegir les instruccions de l'usuari per a no distraure l'atenció. A banda les publicacions afegiran text amb l'enllaç directe la plataforma de distribució.

Exemple de les propostes d'imatge per a la difusió de les obres generades a les xarxes socials:

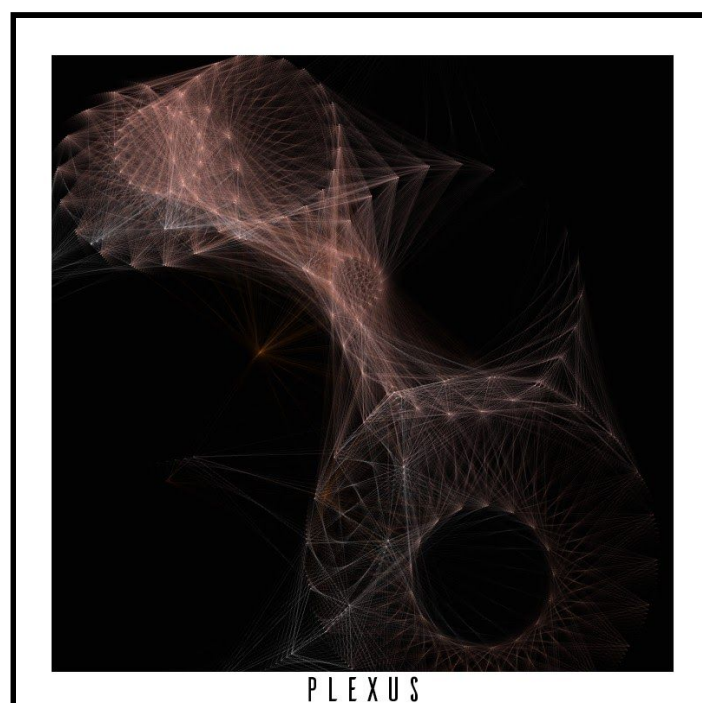


Figura 34: Imatge màrqueting TFG_Plexus_04

26. Conclusió

Durant l'execució de la memòria d'aquest projecte he explicat els objectius principals del Treball Final de Grau, bàsicament aplicar els coneixements adquirits durant l'estudi del Grau Multimèdia de manera pràctica i teòrica davant d'una temàtica que em genera molt d'interès.

La idea ha sigut relacionar els estudis del Grau amb una primera idea de projecte professional envoltat en l'entorn de la programació creativa. Principalment les obres creades tenen la intenció de generar un impacte visual fort a l'usuari i investigar la relació que pot haver-hi entre els gràfics generats per ordinador amb l'estat d'ànim de les persones.

Aquest projecte té una motivació molt definida, ja que com a practicant de la meditació he utilitzat moltes vegades recursos d'aquest tipus per a poder deixar la ment en calma. A banda, de fer ús d'aquests recursos gràfics el fet de codificar-les també aconsegueix relaxar-me i poder utilitzar aquests elements com a ruta d'escapament a l'estrès quotidià del dia a dia.

El resultat del projecte han sigut una col·lecció d'obres visuals creades des de zero que barregen el món més tècnic de la programació amb el món més artístic, barreja que ha marcat la meua vida des de sempre, tant en l'àmbit familiar com al professional. Per a poder assolir els resultats he hagut d'investigar, recordar i aprendre nous coneixements aplicables al projecte, no sols en l'àmbit més tècnic o l'artístic sino també pel que fa a la planificació i la gestió d'un projecte d'aquestes dimensions i durada.

Ara que he finalitzat el projecte estic prou satisfet amb els resultats visuals, però he tingut diferents complicacions pel que fa a la planificació del projecte. He assolit alguns objectius amb retard perquè a l'inici del projecte la temàtica i l'enfocament del projecte era diferent de la final. Aquest canvi en el projecte va generar un retard en els lliuraments que he anat acumulant durant totes les entregues programades, però finalment he pogut efectuar el lliurament final en el termini establert.

Pel que fa a la visió més artística del projecte, qualsevol procés creatiu pot ser millorable, però sempre s'ha d'establir un punt final perquè sino mai podríem donar el projecte com a acabat. Espero que tots els coneixements aplicats a la creació d'aquestes obres m'ajudin a millorar per a les següents creacions i així obtenir uns resultats millors.

De tots els moments difícils o d'incertesa es poden extreure conclusions bones i en aquest cas i com a mètode recordatori per a mi mateix diré **“Escull d'acord amb la motivació i l'interès propi!”**

Annex 1. Lliurables del projecte

- Memòria del projecte.

Aquest document.

- Codi font de les peces.

Arxiu comprimit que inclou tots els fitxers necessaris per al funcionament de les obres en local. És el codi font de totes les peces i el que s'ha importat dins de la plataforma de distribució. Per a poder reproduir-les localment l'usuari haurà d'instal·lar processing desde el següent [enllaç](#).

- Imatges de màrqueting.

Imatges generades per la difusió a les xarxes de les obres generades.

Annex 2. Codi font (extractes)

- TFG_Galàxia_01:

```
// Classe planeta
class Planeta {
    ArrayList<Planeta> satellits = new ArrayList<Planeta>();
    Planeta pare;
    PVector posicio = new PVector();
    float mida = 0;
    float midaRatio = 0;
    float angle = 0;
    float angleRatio = 0.1;
    boolean inversAngleRatio = false;
    color colorPrincipal = color(0, 0, 0, 0);

    Planeta(float midaEntrada) {
        this.mida = midaEntrada;
    }
}

void crearGalaxia(int nivellsCompte, int minSatCompte, int maxSatCompte) {
    totsPlanetes.clear();

    // Creació del primer planeta al centre de la pantalla.
    Planeta masterPlaneta = new Planeta(200);
    masterPlaneta.posicio.set(width/2, height/2);
    masterPlaneta.midaRatio = random(0.1, 0.25);
    masterPlaneta.colorPrincipal = color(255, 255, 255, 0);

    ArrayList<Planeta> nousPlanetes = new ArrayList<Planeta>();
    nousPlanetes.add(masterPlaneta);

    for (int i = 0; i < nivellsCompte; i++) {
        ArrayList<Planeta> tsatellits = new ArrayList<Planeta>();

        for (Planeta p : nousPlanetes) {
            int satelliteCompte = int( random(minSatCompte, maxSatCompte) );
            if (satelliteCompte == 0) {
                continue;
            }

            ArrayList<Planeta> satellits = p.addsatellits(satelliteCompte);
            tsatellits.addAll(satellits);
        }

        totsPlanetes.addAll(nousPlanetes);
    }
}
```

```

        nousPlanetes.clear();
        nousPlanetes = tsatellitits;
    }
}

```

- **TFG_Mandala_02:**

//Calculs/valors per a les línies.

```

float x(float valor2)
{
    return sin(valor2/10)*100 + cos(valor2/valor)*100;
}

```

```

float y(float valor2)
{
    return cos(valor2/10)*100 + sin(valor2/valor)*100;
}

```

```

float x2(float valor2)
{
    return sin(valor2/10)*10 + cos(valor2/valor)*100;
}

```

```

float y2(float valor2)
{
    return cos(valor2/10)*10 + sin(valor2/valor)*100;
}

```

- **TFG_Vòrtex_03:**

//Funció per a generar la figura principal. Utilitzo beginShape i endShape per a poder generar una figura més complexa.

```

void dibuixarFigura() {
    beginShape();
    for (int i = 0; i < nombreVertex; i++) {
        x = figuraRadi * cos(TWO_PI/nombreVertex*i);
        y = figuraRadi * sin(TWO_PI/nombreVertex*i);
        vertex(x, y);
    }
    endShape(CLOSE);
}

```

```

void keyPressed() {
    if (keyCode == UP) {
        figuraSeparacio /= 1.5;
        numFigures *= 1.5;
    } else if (keyCode == DOWN) {

```

```

        figuraSeparacio *= 1.5;
        numFigures /= 1.5;
    } else if (keyCode == LEFT) {
        if (nombreVertex > 3)
            nombreVertex -= 1;
    } else if (keyCode == RIGHT) {
        nombreVertex += 1;
    }
}

```

- TFG_Plexus_04:

```

class Cercle
{
    float x, y, radi, theta = 0;
    int nbLinies = (int)random(3, 25);
    float velRot = (random(1) < .5 ? 1 : -1) * random(.005, .034);
    float velRad = (random(1) < .5 ? 1 : -1) * random(.3, 1.4);

    Cercle(float p_radi, float p_x, float p_y)
    {
        radi = p_radi;
        x = p_x;
        y = p_y;
    }

    void actualitzar()
    {
        theta += velRot;
        velRad *= abs(radi += velRad) > rMax ? -1 : 1;
    }
}

//Classe Color on definim els seus atributs.
class Color
{
    float R, G, B, velR, velG, velB;
    final static float velMin = .2;
    final static float velMax = .8;
    Color()
    {
        R = random(20, 255);
        G = random(20, 255);
        B = random(20, 255);
        velR = (random(1) > .5 ? 1 : -1) * random(velMin, velMax);
        velG = (random(1) > .5 ? 1 : -1) * random(velMin, velMax);
    }
}

```

```

        velB = (random(1) > .5 ? 1 : -1) * random(velMin, velMax);
    }
    //Actualitzem els colors
    public void actualitzar()
    {
        velR = ((R += velR) > 255 || (R < 20)) ? -velR : velR;
        velG = ((G += velG) > 255 || (G < 20)) ? -velG : velG;
        velB = ((B += velB) > 255 || (B < 20)) ? -velB : velB;
    }
}
//Funció per a "connectar" les línies generades dels cercles.
void connectar(Cercle c1, Cercle c2)
{
    float d, x1, y1, x2, y2, r1 = c1.radi, r2 = c2.radi;
    float rCoeff = map(min(abs(r1), abs(r2)), 0, rMax, .08, 1);
    int n1 = c1.nbLinies, n2 = c2.nbLinies;
    for (int i = 0; i < n1; i++)
    {
        x1 = c1.x + r1 * cos(i * TWO_PI / n1 + c1.theta);
        y1 = c1.y + r1 * sin(i * TWO_PI / n1 + c1.theta);
        for (int j = 0; j < n2; j++)
        {
            x2 = c2.x + r2 * cos(j * TWO_PI / n2 + c2.theta);
            y2 = c2.y + r2 * sin(j * TWO_PI / n2 + c2.theta);

            d = dist(x1, y1, x2, y2);
            if (d < dMin)
            {
                stroke(mColor.R + r2/1.5, mColor.G + r2/2.2, mColor.B + r2/1.5,
map(d, 0, dMin, 140, 0) * rCoeff);
                line(x1, y1, x2, y2);
            }
        }
    }
}

```

- **TFG_Nervis_05:**

```

//Classe Vector
class vector {
    float x, y;
    color mColor;

    vector () {
        iniciarParticules();
    }
}

```

```

//Iniciar particules
void iniciarParticules(){
    x = random(width);
    y = random(height);

    float n = noise(x * aleatorietatXY, y * aleatorietatXY, aleatorietatZ);
    mColor = color(color2 + color1 * n, 255, 255);
}

//Actualitzar la posició
void actualitzar(){
    float n = (noise(x*aleatorietatXY, y*aleatorietatXY, aleatorietatZ));
    n = map(n, .1, .9, 0, TP);

    x += n*cos(n * TP) * velMaximaParticules;
    y += n*sin((1-n) * TP) * velMaximaParticules;

    if ((x < 0) || (x > width) ||
        (y < 0) || (y > height)) {
        iniciarParticules();
    }
    stroke(mColor, 80);
    point(x, y);
}
}

void inicialitzar()
{
    color1 = random(85, 110);
    color2 = random(255 - color1);
    aleatorietatZ = random(123456);
    velAleatorietatZ = random(.005, .015) * (random(1) < .5 ? 1 : -1);
    aleatorietatXY = random(.001, .01);
    Particules = new vector[nombreParticules];
    for (int i = 0; i < nombreParticules; i++) {
        Particules[i] = new vector();
    }
    background(0);
}

```

Annex 3. Llibreries/Codi extern utilitzat

Per aquest projecte s'ha utilitzat la web pròpia de processing per a realitzar les consultes generals de les funcions pròpies del llenguatge. Podem trobar aquesta informació a:

- <https://processing.org/reference/>
- <https://processing.org/reference/libraries/>

Per a la creació de les obres s'han utilitzat per a consultar el codi els següents recursos:

- TFG_Galàxia_01:

<https://www.openprocessing.org/sketch/211167>

- TFG_Plexus_04:

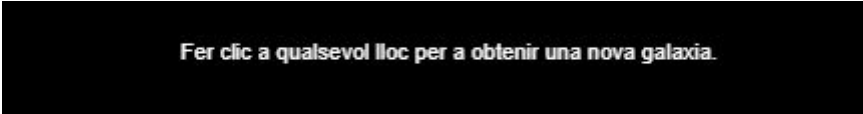
<https://www.openprocessing.org/sketch/650668>

<https://www.openprocessing.org/sketch/772742>

Annex 4. Guia d'usuari

Per a poder interactuar amb les peces generades creades explícitament per aquest projecte s'han de seguir els vinents passos:

- Fer clic a l'enllaç de la plataforma de distribució:
<https://www.openprocessing.org/sketch/1046332>
- Llegir les instruccions per a interactuar amb la peça que apareixen a la part superior de la pantalla.



Fer clic a qualsevol lloc per a obtenir una nova galaxia.

- Interactuar si es desitja mitjançant les ordres habilitades.
- Consultar la informació pròpia de cada peça:
 - Botó 1: Podrem consultar la informació general de la peça.
 - Botó 2: Podrem reiniciar l'execució de l'obra.
 - Botó 3: Podrem consultar el codi font generat per a la peça.



- Altres opcions disponibles:



- Botó 1: Donar "Like" a la peça oberta.
- Botó 2: Afegir un comentari a la peça.
- Botó 3: Crear una versió de la peça al perfil de l'usuari que l'està consultant per a poder variar el codi.
- Botó 4:
 - Compartir la peça a Facebook, Twitter.
 - Descarregar la peça per a poder obrir-la en local.
 - Copiar codi per a inserir la peça a una web.
 - Atribuir les credencials necessàries estipulades per al tipus de llicència establert.
- Botó 5: Accés al perfil d'usuari.

Per a poder reproduir-les localment l'usuari haurà d'instal·lar processing al seu ordinador desde el següent [enllaç](#).

Annex 5. Bibliografia

- Processing. <www.processing.org> - Accés: Durant tota l'execució del projecte.
- openProcessing. <www.openprocessing.org> - Accés: Durant tota l'execució del projecte.
- Boden, Margaret A. 2009. What is Generative Art?. In: Digital Creativity. London:Routledge - Accés: 25 / 09 / 2020.
- Entorn Processing. <<https://processing.org/reference/environment/>> - Accés: 27 / 09 / 2020.
- API Processing. <<https://processing.org/reference/>> - Accés: Durant tota l'execució del projecte.
- Llibries Processing. <<https://processing.org/reference/libraries/>> - Accés: Durant tota l'execució del projecte.
- Tutorials processing. <<https://processing.org/tutorials/>> - Accés: 27 / 09 / 2020.
- Referència George Nees. <<https://elartedigital.wordpress.com/artistas/george-nees/>> - Accés: 25 / 09 / 2020.
- Programació creativa. <https://en.wikipedia.org/wiki/Creative_coding> - Accés: 30 / 09 / 2020.
- Compos 69. <https://monoskop.org/Compos_68> - Accés: 01 / 10 / 2020.
- Demoscene. "[About the Demoscene](#)". Demoscene - The Art of Coding. Retrieved 22 May 2020. - Accés: 01 / 10 / 2020.
- Contemporary Computer Art. Toby Juliff, Travis Cox (April 2015). "[The Post-display condition of contemporary computer art](#)". eMaj. 8. Archived from [the original](#) on 29 June 2016. Retrieved 31 May 2016. - Accés: 01 / 10 / 2020.
- DevArt . <<https://devart.withgoogle.com/>> - Accés: 30 / 09 / 2020.
- The post-display condition of contemporary art. <<http://hdl.handle.net/11343/52793>> - Accés: 01 / 10 / 2020.
- Metodologia scrum. <<https://www.scrum.org/>> - Accés: 15 / 10 / 2020.
- Guia Scrum. <<https://www.scrumguides.org/scrum-guide.html>> - Accés: 15 / 10 / 2020.
- Sprint Scrum. <<https://www.scrumguides.org/scrum-guide.html#the-sprint>> - Accés: 15 / 10 / 2020.
- POO. Coad, P; Yourdon, E. (1991). Prentice-Hall International editions, ed. [Object-oriented Design](#). ISBN 9780136300700. - Accés: 17 / 10 / 2020.
- Programació orientada a objectes. <<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>> - Accés: 18 / 10 / 2020.
- Google Arts & Culture. <<https://artsandculture.google.com/>> - Accés: 28 / 09 / 2020.
- Saatchi Art. <<https://www.saatchiart.com/>> - Accés: 28 / 09 / 2020.
- Galeria Pace. <<https://www.pacegallery.com/>> - Accés: 28 / 09 / 2020.
- Generative Artist. <<https://generativeartistry.com/>> - Accés: 29 / 09 / 2020.
- Tholman. <<https://tholman.com/>> - Accés: 29 / 09 / 2020.
- Apple Store - Heart Work. <<https://apps.apple.com/app/heart-work/id1380764163>> - Accés: 12 / 11 / 2020.
- Apple Store - Silk 2. <<https://apps.apple.com/es/app/silk-2-generative-art/id1050339928>> - Accés: 12 / 11 / 2020.
- Web Heart Work. <<https://heartwork.app/>> - Accés: 12 / 11 / 2020.
- Web Silk. <<http://weavesilk.com/>> - Accés: 12 / 11 / 2020.

- Càrrega JavaScript.<<https://lenguajejs.com/javascript/introduccion/como-funciona/>> - Accés: 03 / 01 / 2021.
- Etiquetes HTML. <<https://lenguajehtml.com/html/scripting/etiquetas-html-scripts/>> - Accés: 03 / 01 / 2021.
- Bárbara Almeida Sketch. <<https://www.openprocessing.org/user/45107>> - Accés: 03 / 11 / 2020
- Referència galàxia. <<https://www.astrobitacora.com/fusion-de-14-galaxias-inicio-universo/>> - Accés: 03 / 11 / 2020
- Referència Mandala.
<<https://www.aprendemas.com/co/blog/mundo-educativo/por-que-colorear-mandalas-ayuda-a-l-aprendizaje-75572>> - Accés: 03 / 11 / 2020
- Referència Vòrtex.<<https://ca.wikipedia.org/wiki/V%C3%B2rtex>> - Accés: 03 / 11 / 2020
- Referència
Plexus.<<https://motionarray.com/after-effects-templates/plexus-explosion-logo-85943>> - Accés: 13 / 11 / 2020
- Referència Nervis.<<https://www.pikist.com/free-photo-snmitt/es>> - Accés: 14 / 11 / 2020
- Tipografia imatges màrqueting.
<<https://www.behance.net/gallery/25541553/Ailerons-Typeface>> - Accés: 07 / 01 / 2021.

Annex 6. Vita

Em dic Rafel Bernabeu Aznar, treballo com a rigger en la indústria d'animació 3D. He tingut la sort de participar en molts projectes en aquest sector, treballant en films com Atrapa la Bandera, Tadeo Jones 2 o Animal Crackers.

Des que era petit tinc interès en tot el que envolta a la tecnologia i la creació de continguts mitjançant l'ordinador. Quan vaig començar els meus estudis universitaris no existia una enginyeria que barregés el món tècnic amb el món artístic, és per això que vaig decidir matricular-me a l'Enginyeria Tècnica Informàtica que oferia la Universitat de València. Amb el temps em vaig donar de què estava deixant de costat el que més m'agradava, els gràfics per ordinador, ja que a la Politècnica la carrera escollida estava enfocada completament a l'àmbit tècnic. Aquesta frustració va derivar en la presa d'una de les decisions més importants a la meua vida, començar un curs d'animació 3D. Aquest factor va ser un canvi determinant per a la meua motivació i per a poder començar una carrera professional en aquest sector.

En la meua temporada a Madrid vaig descobrir el Grau Multimèdia i com sempre he tingut l'espina de acabar els estudis universitaris vaig decidir traslladar el meu expedient del politècnic a la UOC. Aquest Grau és el que hauria d'haver escollit quan tenia divuit anys, però no existia. Actualment em trobo a l'etapa final d'aquests estudis i estic completament satisfet d'haver escollit aquesta variant.

La intenció final de realitzar aquest Grau és formalitzar els coneixements perquè en el sector on treballo no hi ha massa estabilitat i per tant és una forma d'obrir-se noves possibilitats en altres sectors existents com l'educació.

Les meves característiques més destacables són l'ordre, la constància i la rapidesa en la resolució de problemes derivants de la tecnologia.

Annex 7. Diagrama de Gantt

Diagrama de Gantt - Programació Creativa

