

# TFG

---

## PIEZAS JUGABLES

Presentado por Marina Portela Ruiz

Tutora: Sara Vilar

Facultat de Belles Arts de Sant Carles

Grado en Bellas Artes

Curso 2013-2014



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
FACULTAT DE BELLES ARTS DE SANT CARLES

## RESUMEN Y PALABRAS CLAVE

Los elementos lúdicos en las obras de arte son algo que siempre me había interesado, mi propósito era el de generar, a lo largo del grado, una relación de obras coherentes y cohesionadas que dialoguen sobre el juego dentro del arte. Piezas en todo caso interactivas y jugables, que supongan una suerte de tarjeta de visita que me confiera una identidad como artista. No pretendía en ningún caso generar una serie, o varias versiones de una misma obra. Si no un modo de pensar y operar que podría verse como trasfondo en todas ellas.

En este trabajo describo esta relación de obras y el trabajo realizado hasta llegar a ellas.

Interactivos - Juego - Instalaciones - Arduino -Processing

## AGRADECIMIENTOS

A mis padres por cumplir con su palabra, y por su apoyo incondicional y su cariño. A Jano por permanecer siempre a mi lado apoyándome durante estos cuatro años. A mis abuelos por servirme tantas veces de inspiración. A Paula, María y Marta porque cuando empecé todo esto ya sabían que iban a vivirlo como si ellas mismas hicieran el grado. Y a las nuevas incorporaciones que he hecho en mi vida al o largo de estos años, especialmente a Ana, Miriam y Bea. A Mar sin la que varios de estos proyectos ni habrían sido tan emocionantes ni habrían tenido sentido. A los compañeros que he tenido en cada uno de los trabajos, por permitir siempre que metiera mano y los gamificara un poco. A Paco Giner por la gran ayuda que me ofreció cuando se me ocurrió que el arte y el videojuego sería un buen tema para una Tesis (que nunca llegué a hacer, pero cuya bibliografía ha sido muy útil en todo esto) Y por supuesto a Sara Vilar, porque abrió un mundo de posibilidades ante mis ojos.

Y a MiguelH, Yol, Alba, Nayara, Iris, Natalie, Ana, Lore, Miriam, Patri, Bego, Po, Alejo, Martita, Ali, Javi, Trini, Vic, Gels, Gema, Estevan, Rosa, Ö, Jaime, Bondi, Luch, Ana, Laura, Dema, Alberto, Noe, Andrés, Carmen, Esther, Laura, Miguel, Virginia, Cris, Vero, MJ, y MariaR.

# ÍNDICE

1. INTRODUCCIÓN	5
2. OBJETIVOS Y METODOLOGÍA	7
3. CUERPO DEL PROYECTO	9
3.1. DESDE LAS ADIVINANZAS HASTA EL JUEGO	13
3.2. “JUGABLE” COMO PREMISA	17
3.3. PIEZAS ENMARCADAS EN EL PROYECTO	21
3.3.1. Perfiles, 2012	21
3.3.2. Hormigas, 2012	26
3.3.3. Atravesar, 2013	31
3.3.4. Preparados, Listos... Luz!	33
3.3.5. Sambori, 2014	37
3.3.6. Quiz, 2014	40
4. CONCLUSIONES	43
5. BIBLIOGRAFÍA	44
6. ANEXOS	46
6.1. Código de <i>Hormigas</i>	46
6.2. Código de <i>Preparados, listos... Luz!</i>	97
6.3. Código de Código de <i>Quiz</i>	108

# 1. INTRODUCCIÓN

“El hombre solo juega cuando es hombre en pleno sentido de la palabra, y solo es enteramente hombre cuando juega”<sup>1</sup>

Friedrich Schiller

En este proyecto he hecho un ejercicio de reflexión sobre cómo he llegado a hacer lo que hago, y qué tipo de trabajos son los que más me interesan. Para explicarlo correctamente no se me ocurre otra manera que la de presentarme y explicar lo que he venido haciendo durante los últimos diez años.

Siempre he sentido un especial interés por el mundo del arte, pero cuando llegó el momento de decidirme por una carrera me asusté. Era buena en matemáticas y física y me gustaban mucho los ordenadores, decidí hacer algo con trabajo asegurado como era, en aquel momento, la Ingeniería de Informática de Sistemas, de la UPV. No me arrepiento de haberla hecho, mi trabajo no sería lo mismo sin ese bagaje, pero a los pocos años de trabajar como informática me di cuenta de que no quería que mi vida pasara sin darme una oportunidad en Bellas Artes. Nada más terminar la carrera, me matriculé en el Máster Oficial en Producción Artística, pero me supo a poco. Tres años después decidí dejar el trabajo y cursar el grado en Bellas Artes.

Mientras realizaba la carrera de informática hacía prácticas en el IVAM, como guía didáctico. Pasé muchas horas allí, lo que me permitió dedicarle un tiempo a ver exposiciones y leer algunos libros sobre arte. Tuve la enormísima suerte de coincidir en el tiempo con una exposición espectacular de James Turrell que cambió del todo mi visión del arte. Habían personas que no veían más que una bandera en los cuadros de Jasper John, o que veían inacabadas las esculturas de Matisse o no se maravillaban con las contorsiones del metal de las figuras de Gargallo. Pero esta exposición tenía algo que hacía que todo el mundo la disfrutase, de uno u otro modo.

Desde mis primeros trabajos, cuando aun no analizaba del todo, o nada, el porqué de lo que hacía, empecé a interesarme por hacer piezas con las que de alguna manera se pudiera jugar.

Creo que el razonamiento que usaba se basaba en que mucha de la gente que me rodea, cuya opinión me importaba, no siente ningún interés por el arte, especialmente por el arte moderno, en el que pretendía sumergirme. Creo que al principio quería llegar a ellos de alguna manera, buscar una especie de treta que les suscitara la curiosidad suficiente para acercarse a ver mi trabajo, y que de alguna manera pudiera surgir algún tipo de empatía, aunque el trasfondo de las piezas les siguiera dando completamente igual.

---

1 SHILLER, F. *Cartas sobre la educación estética del hombre*: Anthropos, 1990.

El recurso que me pareció más conveniente en aquel momento era el de la interacción. Si les hacía manipular, modificar, dialogar... En definitiva jugar, con mi trabajo, quizás llegara a conseguir aquel acercamiento que tanto echaba en falta. Resultó que, aunque no funcionó tanto como me habría gustado, a mí sí me funcionó, con aquello generaba trabajos que me interesaban mucho más. Al mismo tiempo podía explotar mi parte informática, y con ello no me refiero a programar, si no a generar un diálogo “pieza-usuario” que funcionara análogo al “ordenador-usuario”. Podía utilizar mis nociones sobre usabilidad, público objetivo, interacción con el usuario, etc., Sobre los que, aunque para otro ámbito, ya estaba formada.

Me habría gustado añadir un capítulo sobre referentes, en el que hablar de la exposición *Play or bit* de 1966, donde se podían ver piezas como *Concave Convexe* de Bruno Munari. O hablar de obras mágicas como *Rain Room* de Random International, o *Waves* de Daniel Palacios, *Theaters* de Sugimoto Hiroshi. O tan divertidas como *My little piece of privacy* de Niklas Roy. Explicar lo mucho que me han influenciado los laberintos de Pistoletto, las Burbujas de WKA Finetsern, los móviles de Calder, los juegos visuales de Zilvinas Kempinas en *Tube*, los gifs animados de Daniel Barreto o el contador de Che-Wei Wang *Counting to a billion*. Hablar de ejemplos de espacios lúdicos tan claros como son *On space foam* de Tomas Saraceno, o *Tambor* de Ernesto Neto, o una de mis preferidas, *SandBox* de Rafael Lozano Hemmer. Además de todas las que sí han encontrado sitio a lo largo del proyecto. Pero debido a la extensión del mismo me ha resultado imposible. Aun así no podía terminar esta introducción sin ni siquiera nombrarlas.

## 2. OBJETIVOS Y METODOLOGÍA

Como ya he señalado, los elementos lúdicos en las obras de arte son algo que siempre me había interesado, mi propósito era el de generar, a lo largo del grado, una relación de obras coherentes y cohesionadas que dialoguen sobre el juego dentro del arte. Piezas en todo caso interactivas y jugables, que supongan una suerte de tarjeta de visita que me confiera una identidad como artista. No pretendía en ningún caso generar una serie, o varias versiones de una misma obra. Si no un modo de pensar y operar que podría verse como trasfondo en todas ellas.

Siempre que veo una pieza que me interesa, por cualquier motivo, la archivo. Mantengo siempre esa actividad de coleccionar trabajos de otros que llaman mi atención, sobre todo aquello que me seduce de forma inmediata, y aquellas piezas que me suscitan un “ojala lo hubiera hecho yo”. Me sirve cualquier tipo de contenido, desde un dibujo a lápiz, una frase elocuente, hasta el vídeo de una complicada pieza expuesta en el MOMA.

Reviso esta colección, que va creciendo día a día, analizando que elementos pueden tener en común, que es lo que considero que funciona, intentando destilar de ellas lo que pueda serme útil. Para alimentar esta colección consulto de forma regular varias páginas de enlaces, tipo Pinterest. Sigo en varias redes sociales varios artistas que han ido entrando en mi colección, y eso me permite ver que les interesa a ellos y que siga creciendo. Es una fuente de información que considero muy valiosa.

Además llevo siempre conmigo, bien por los dispositivos móviles de última generación, una aplicación de gestión de RSS, que me permite ver de forma rápida un sinnúmero de blogs sobre el tema. Páginas de artistas, de galerías, de difusión de arte. Y en cualquier rato libre leo, gestiono, colecciono. No es algo que haga cuando me pongo en el reto de crear una pieza, pero lo considero parte elemental de mi metodología de trabajo. No es cuestión de disciplina, sinceramente me interesa.

Normalmente, hasta la fecha, mis ideas parten de alguna premisa, un ejercicio de clase, la propuesta de un amigo... Una especie de invitación. Como si para crear la pieza necesitara de un sitio donde debiera encajar. De todas formas si se me ocurre una idea siempre la anoto por si encajara en algún momento. Aunque normalmente no son las que llevo a término.

Lo primero que hago cuando me llega una propuesta es pensar en el juego ¿Qué tipo de juego encaja con la propuesta? En ocasiones ninguno lo hace, pero normalmente encuentro algo que me sirva. En esta fase la batería de imágenes y conceptos que reviso son las de juego. Genero una lista de ideas y las voy analizando poco a poco, descartando lo que no me sirve, enredando lo que parece que sí. Y cuando creo que he llegado a la buena la depuro, la simplifico al máximo y la termino de definir a partir de ahí.

Habitualmente mis primeros impulsos están un poco desajustados, o son imposibles, o no sirven para el lugar, no transmiten lo que deben, o se me van del presupuesto. Me toma un tiempo definir el trabajo que voy a hacer. Siempre tengo que hacer una depuración de ideas, meditarlas y dormir las hasta que cuajan del todo. Suele llegar un momento en el que la idea, que inicialmente me parecía interesante, de momento me entusiasma, y sé que es ahí cuando tengo que empezar a trabajar.

Una vez definida empiezo a hacer pruebas. Si necesito sensores, por ejemplo, trabajo un poco con ellos hasta que entiendo del todo su funcionamiento y veo que los puedo controlar. O en el caso de materiales como el alginato, hago algunas pruebas para ver qué tal registra, en qué dimensiones funcionan mejor... Hago muchos bocetos, esquemas, mapas, presupuestos, incluso me cuento por escrito varias veces lo que quiero de la pieza para ver si tiene sentido.

Cuando tengo claros todos los materiales, el lenguaje de programación en el que quiero trabajar, etc. compro todo lo que necesito y construyo la pieza. Y la pongo al alcance de los que tengo más cerca para que la prueben, También hago vídeos y fotos durante el proceso de construcción y se las hago ver a mis padres y a mis amigos más cercanos para que me den su opinión.

La mayoría de las piezas que hago no están del todo completadas hasta que están en el lugar en el que las voy a exponer. Así que el momento de montar suele ser decisivo. Procuro minimizar lo máximo posible el tiempo de montaje porque soy consciente de que es allí donde más problemas puedo tener, y intento tener tiempo para maniobrar y cambiar cualquier cosa si fuera necesario.

Por supuesto cada proyecto es un mundo, pero en términos generales todos ellos han pasado por todas estas fases para ser concretados.



### 3. CUERPO DEL PROYECTO

Antes que nada explicaré brevemente la relación que se da entre el arte y el juego. Cómo el uno y el otro se buscan de manera natural, y comparten diversas características que hacen de su uso en conjunto un beneficio. Uno de los impulsos fundamentales del arte moderno es el deseo de anular la distancia que media entre la audiencia y la obra, para ello trataban de implicar al espectador como participante en ella.

Los vínculos que se establecen entre el juego y el arte de nuestros días empiezan a localizarse en los movimientos de las Vanguardias históricas, que en su búsqueda de lo nuevo y llevados por su espíritu transgresor hacen que los artistas recuperen para el arte el concepto de juego. Sirviéndose de él para canalizar muchos de los grandes temas de interés, como era este del mayor acercamiento y participación del espectador.

El arte moderno está además emparentado con el juego en su carácter irónico e ingenioso. No son pocos los escritos y posturas que niegan o se burlan del arte mismo y que analizadas consecuentemente persiguen todo lo contrario, pues en realidad son una afirmación que profundiza en el valor del arte.

Creo pertinente hacer una revisión del texto *El elemento lúdico del arte*<sup>2</sup> del filósofo alemán Hans-Georg Gadamer para explicar un poco más mi posición.

Gadamer afirma que el juego es, en primer término, un movimiento de vaivén que se repite continuamente, como el “juego de luces” o el “juego de las olas”, donde se presenta un continuo ir y venir que no está vinculado a fin alguno. Pero la humanidad, capaz de burlar lo característico de la razón respecto a fines, ordena y disciplina sus propios movimientos de juego como si tuviesen fines. Por ejemplo, cuando un niño cuenta cuantas veces bota el balón en el suelo antes de escapársele. Por lo tanto para que el juego sea tal deben existir ciertas reglas auto impuestas, o aceptadas por el jugador.

Por otra parte, afirma que el jugar exige siempre un “jugar-con”, aunque el que acompañe en el juego sea un mero espectador, esto no le convierte en otra cosa que en un participante. Pensemos por ejemplo en el caso del fútbol, aquellos que de verdad lo siguen sienten que ganan o pierden con sus equipos, al verlo, participan en su juego. Y también, al igual que los jugadores, conocen las reglas del juego y las aceptan. Cuando aparece o se valora en cuanto a experiencia hay ahí algo que se ha comprendido. Es la misma identificación que constituye el sentido de la obra.

Es este un desafío que como juego sale de la “obra” y que espera ser correspondido. Exige una respuesta que sólo puede dar aquel que haya aceptado el desafío. Y esta respuesta tiene que ser propia, la que el mismo es-

pectador produce activamente. De manera que el espectador es a su vez un co-jugador de la obra.

“Sólo habrá una recepción real, una experiencia artística real de la obra de arte, para aquel que «juega-con»”<sup>3</sup>

Con esto Gadamer nos anuncia que toda obra debe de ser participada para ser vista en plenitud. En el grado que la misma obra lo permita. Esto confirma de alguna manera que el realizar piezas con las que se pueda interactuar, o jugar debe de ayudar a este acercamiento, por lo menos predispone al espectador a activarse frente a la obra.

El arte contemporáneo establece a través de su criterio lúdico la participación del espectador, buscando vías para integrar al público en el proceso creativo, lo cual no sería posible sin un planteamiento lúdico del arte. Ya que como veíamos anteriormente, para esta intervención del público debería estar convenientemente preparado a través de una enseñanza necesaria para la participación consciente en la acción artística.

Esto implica un conocimiento de los lenguajes en los diferentes movimientos artísticos, así como un cierto dominio perceptivo del espacio, la forma, el color, etc., pero este tipo de educación es minoritaria, por lo que acercarnos a la utopía del arte moderno para todos, en el que todos podemos participar depende de que abramos las puertas correctas, porque este acercamiento debe suceder de motu propio, por parte del público. Volviendo al fútbol, no es algo que se dé en clase, nadie nos hace estudiar sus normas y su historia, pero una gran mayoría de la población española tiene grandes nociones sobre el tema.

En la *Crítica del juicio estético* Kant dice así:

“Sólo cuando la imaginación, en su libertad, despierta el entendimiento, y éste, sin concepto pone la imaginación en un juego regular, entonces se comunica la representación, no como pensamiento, sino como sentimiento interior de un estado del espíritu conforme a fin”<sup>4</sup>

El juego puede ser un buen mediador para situar la mente del espectador en un estado más propicio para observar y entender una obra. Pero ¿Qué es un juego? Es necesario entender bien el concepto de juego y su relación con el concepto de arte para poder manejarnos en este terreno. Existen incontables definiciones para juego, desde un amplio abanico de puntos de vista, pero este contexto en concreto he seleccionado dos que considero más adecuadas.

Huizinga, en su clásico *Homo Ludens* lo define como paso que conduce al proceso creativo:

“En su aspecto formal, es una acción libre ejecutada como si y sentida como situada fuera de la vida corriente, pero que, a pesar de todo, puede absorber por completo al jugador, sin que haya en ella ningún interés material

3 GADAMER, H-G. *La actualidad de lo bello*: Paidós Ibérica, 1991

4 KANT, I. *La crítica del Juicio*, Immanuel Kant: Maia Editores, 2011

ni se obtenga de ella provecho alguno, que se ejecuta dentro de un determinado espacio, que se desarrolla en un orden sentido de las reglas y que da origen a asociaciones que propenden a rodearse de misterio o a disfrazarse para destacar del mundo habitual".<sup>5</sup>

El psicólogo Dr. Eric Berne por su parte nos dice que el hombre necesita del juego para mantener un equilibrio estable, ya que "es la única respuesta completamente satisfactoria al hambre de reconocimiento y la necesidad de programación"<sup>6</sup>. Esta necesidad de programación trataría sobre todo de evitar el aburrimiento que conduce al deterioro emocional y biológico de la persona. Concluye aclarando que debido a las pocas oportunidades de tener una verdadera intimidad en la vida diaria, las personas invierten la mayor parte del tiempo de la vida seria en juegos.

Encuentro en ambas ciertos paralelismos con el propio arte, ambos presentan el mismo carácter necesario, El hombre entiende el mundo a través del juego, desde pequeños, y a lo largo de toda su vida necesita momentos lúdicos en los que unas reglas insólitas le hagan olvidar las otras que le impone la vida cotidiana, familiar, profesional. El arte es también una necesidad universal y, al igual que el juego, no es una necesidad práctica, primaria, ligada a la supervivencia, como por ejemplo alimentarse, sino únicamente espiritual e intelectual.

Es esa libertad la que hace que el juego sea necesario en nuestra vida, y la obra de arte es también un espacio privilegiado donde todo es posible. Ideas preestablecidas, como el tiempo o la distancia, se prestan a una libre interpretación, y pueden ser revisadas y renovadas tanto por los jugadores como por los artistas.

El juego mantiene muchas características comunes con el arte. Tomando como base las características del juego analizadas por Huizinga en el *Homo Ludens* han sido identificadas las siguientes cualidades en común:

- Tienen su fin en sí mismo.
- En el juego y en el arte se encuentra el concepto de placer funcional debido al placer que proporciona la propia actividad, independientemente de su finalidad.
- La cualidad que determina ambas acciones es que se diferencian de la vida corriente.
- Son ante todo una actividad que se desarrolla libremente.
- Ambas actividades se practican con la mayor seriedad y que escapen de la vida productiva no supone tomarlas a broma.
- Son una de las maneras de humanizar el azar y la forma menos traumática de acceder a lo no enunciado.
- Las reglas que las articulan propician el impulso de crear de forma acorde con unas premisas aceptadas por los participantes, que no admiten dudas.

---

5 HUIZINGA, J. *Homo Ludens*: Alianza Editorial, 2000

6 BERNE, E. *Juegos en los que participamos*: RBA Libros, 2007

- La primera regla para el juego y para el arte es querer ingresar en el ámbito correspondiente, ya sea como autores o espectadores.
- Se practican dentro de determinados límites de tiempo y espacio en los que existe un orden propio y absoluto.
- Muestran interés y arraigo por lo primitivo.
- Los dos poseen una función didáctica reconocida.
- Propenden a ser bellos en el desarrollo de sus formas y pueden elevarse por encima de lo serio.
- La ironía, el humor y la sátira pueden formar parte del juego y del arte.
- Continuamente utilizan en mayor o menor medida términos como reto, confrontación, provocación, competición, riesgo.
- El individuo que infringe las reglas del grupo o se sustrae es un aguafiestas que debe de ser apartado del mimo.
- Tienen el carácter propio de las fiestas de lo excesivo y superabundante, de lo ritual y del intercambio.<sup>7</sup>

Claramente aquí se está hablando en términos muy generales, pero podemos reconocer nuestro ámbito de trabajo en todas estas premisas de una forma u otra. Viendo esta gran cantidad de parecidos no es complicado entender que me resultarían inseparables al poco de ponerme a trabajar sobre el tema. Llegada a cierto punto ya no veía coherente que alguno de mis trabajos no fuera también un juego, como soporte y de forma explícita. Lo que empezó como una decisión fortuita se fue haciendo poco a poco una manera de trabajar.

---

7 GUTIÉRREZ, T. *La significación del juego en el arte moderno y sus implicaciones en la educación artística [tesis doctoral]* Madrid: Universidad Complutense de Madrid. 2004. [consulta: 2014-06-03] <<http://biblioteca.ucm.es/tesis/bba/ucm-t28325.pdf>>

### 3.1. DESDE LAS ADIVINANZAS HASTA EL JUEGO

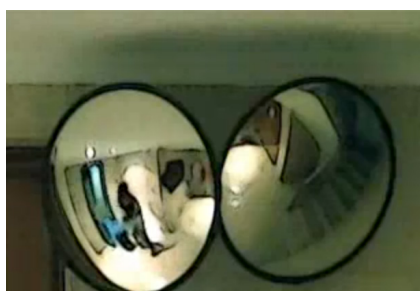
Pensamos que era necesario antes de hablar sobre las piezas seleccionadas como conjunto del proyecto, hacer una serie de apuntes sobre cómo llegamos hasta ellas, en este capítulo desarrollo brevemente los trabajos realizados durante el Máster Oficial en Producción Artística, que tuvo lugar en su primera edición, 2005-2006.

Como especialización en informática escogí multimedia, es una carrera que permite cierto coqueteo con las artes plásticas, desde el diseño, la edición de vídeo y de audio, el 3D... Al final son temas que también trabajamos aquí en el grado. Al llegar al máster ya había iniciado esa búsqueda de la que hablaba en la introducción, de una forma interactiva de tratar el arte. El primer trabajo que nos pidieron durante el máster fue el de realizar un pequeño corto, daba igual la técnica. El corto se enmarcaría en un supuesto programa de televisión creado por el conjunto de la clase.

El vídeo, realizado junto con Andrés Martín (compañero de informática que se mudó conmigo a la facultad de bellas artes) pretendía ser una historia abierta que cada uno pudiera interpretar a su manera. Nosotros le dimos una historia base, que pensamos que podía tener varias lecturas, y la llamamos *Casi 3 minutos*, su duración. La forma en que introdujimos aquí al espectador fue sin duda alguna errónea. Las respuestas que tuvimos fueron muy frías, no despertaban la curiosidad de nadie. Pensamos que se sentían decepcionados, como si hubiésemos incumplido un contrato no escrito con el que nos veía en el que la introducción, el nudo y el desenlace debían quedar nítidamente aclarados. Supongo que nuestras capacidades como guionistas dejaban que desear y no fuimos capaces en aquel momento de resolver una propuesta de ese tipo de manera convincente. Dedujimos que en este caso lo interesante habría sido construir una interacción real, en la que el espectador pudiera elegir como continuaba la historia en cada caso.

Viendo el mal resultado del anterior, cuando nos propusieron el siguiente proyecto de vídeo probé con algo mucho más tangible. En este caso me decidí por un vídeo corto que pudiera servir para un flipbook. Una narración muy breve pero transformada en objeto que pudieran manipular y pasarse de unos a otros. Aunque el contenido del trabajo era mucho menos interesante, esto sí que funcionó, todo el mundo quería tocarlo, verlo, pasarlo a quien tenía al lado... Tenía que encontrar un punto medio entre contenido y medio.

Al llegar la asignatura de *Medios interactivos* tuve la primera oportunidad de hacer una instalación interactiva reactiva. Junto con mis compañeras Begoña Martínez, María Jesús Olmos hicimos *Macro-Micro*. Fue una buena conjunción, May venía de hacer Imagen y Sonido, Begoña de Bellas Artes y yo, como ya se sabe, de informática. Pusimos un poquito de todo para formar la instalación que en resumen no era más que la proyección de dos vídeos sobre viajes, pero de una forma muy cuidada. Queríamos que el espectador

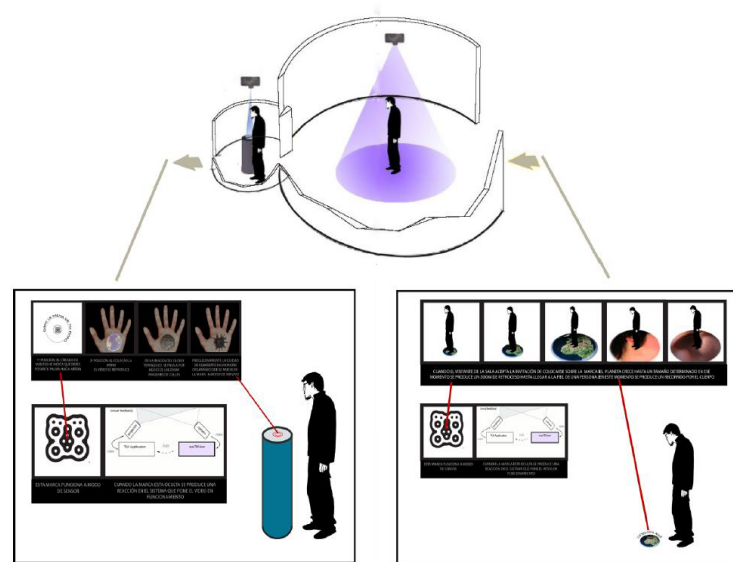


Capturas del corto  
*Casi 3 minutos*, 2006

Foto del Flipbook, 2006.



*MicroMacro*, 2006.  
Imágen del montaje del prototipo  
Diagrama de la instalación.



se encontrara con los videos, y de alguna manera sintiera que viajara con nosotras, influir en su percepción.

“Un lado te hará crecer, y el otro menguar dijo la Oruga: Bébeme”<sup>8</sup>

Es una de las invitaciones que Alicia encuentra al traspasar el umbral. Nuestra invitación ofrecía al usuario la oportunidad de adentrarse en el otro lado, allí donde las proporciones fluctúan de lo macro a lo micro, de lo natural a lo tecnológico: los mundos se abren bajo los pies o se encierran en la palma de la mano.

El espacio propuesto se dividía en dos salas contiguas separadas por una puerta de características manguantes. Por un lado se situaba la primera estancia, en el suelo una proyección incitaba al espectador a pisar la imagen de la tierra. Cuando un usuario se posicionaba encima, la imagen se agrandaba. Comenzaba entonces un viaje que iba desde el exterior del planeta tierra hasta llegar a un recorrido por la piel humana. En la segunda estancia, el usuario se encontraba con una peana en cuya superficie se leía una nota que orbitaba alrededor de una imagen del planeta tierra. El mensaje ordenaba “tiéndeme la palma de la mano”. Cuando éste tapaba la marca se activaba un video sobre la piel de su mano. La visión del pequeño planeta se aventuraba a través de un zoom cenital de una ciudad que paulatinamente se transformaba en un organismo microscópico en constante mutación. Este viaje tenía lugar dentro de los límites de la superficie de la mano.

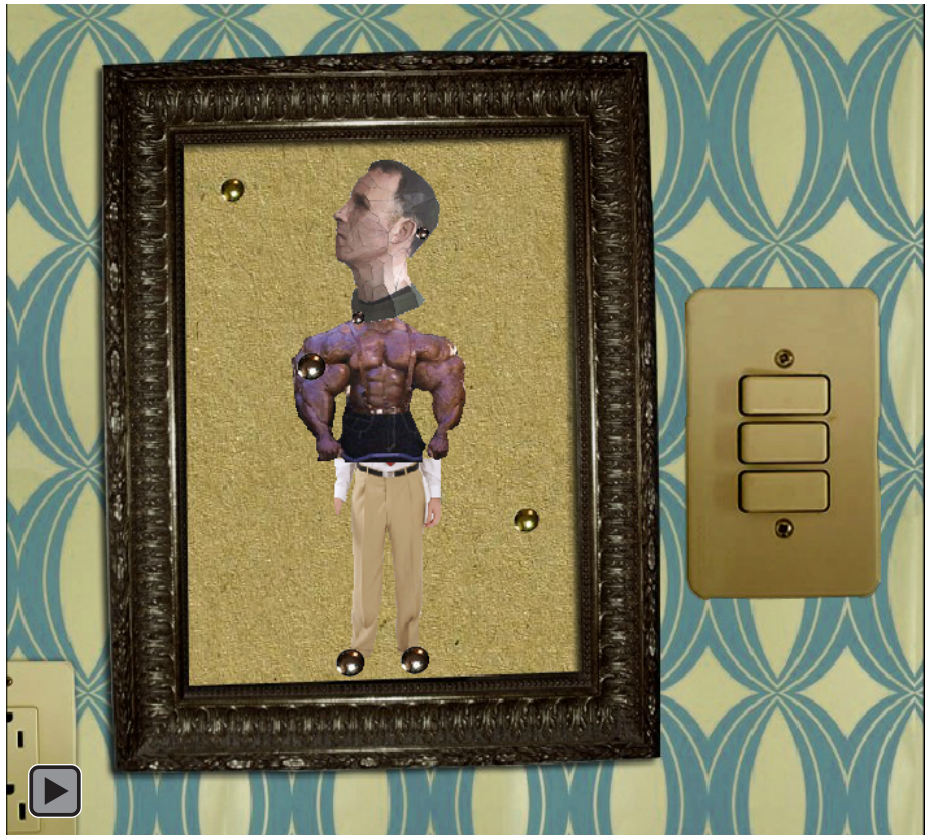
Esta instalación reunía todo lo que yo estaba buscando en aquel momento, habíamos conseguido crear una pieza en la que la curiosidad del usuario sea activada, querían saber qué sucedía cuando se colocaban aquí o ponían la mano allá, y percibían sin problema la poesía de la pieza. Contábamos algo que nos interesaba de una manera que resultaba entretenida.

Presentamos esta pieza a un par de concursos y obtuvimos buenos resultados, nos abrió las puertas para una beca residencia en el Medialab Prado para trabajar con el sistema Reactivision (A la que no pude asistir por el trabajo que tenía en aquel momento, pero que mis compañeras sí disfrutaron), y un vídeo sobre la pieza circuló por varias webs de instalaciones durante algunos meses.

A partir de este momento me centré en el tema de la interacción, y aunque los siguientes proyectos del máster fueron de menor envergadura seguí estudiando las posibilidades que me brindaba con cada uno de ellos. Hice junto con mis compañeras del máster Laura Orts, Esther Señor y Carmen Cifrián una pequeña pieza online que consistía en una suerte de cadáver exquisito. Un corcho en el que colgaban diferentes recortes de todo aquello que encontramos que nos evocaba una cabeza, un cuerpo o unas piernas, y mediante unos interruptores se podía transitar entre las diferentes piezas para formar el cuerpo que más apeteciera. Con esto introducía en mi trabajo la posibilidad espectador de formar parte del proceso creativo. Recurso que actualmente considero valiosísimo.

Captura de la pieza online *Cadaver Exquisito*, 2006.

NOTA.- Para hacerlo funcionar hay que apretar los interruptores Nótese que la imagen es la versión original en Flash, lista para jugar.



También en la asignatura de Video en tiempo real proyectamos una instalación interactiva, que lamentablemente nunca llegamos a montar. Junto con mis compañeros en la asignatura Laura Orts y Miguel Delicado preparé un programa en PD que pretendía crear imágenes en movimiento que recordaran a las de Eadweard Muybridge, pero en tiempo real.

La Instalación tenía dos sensores, uno al inicio y otro a la salida de la pieza. En uno de los laterales tenía una cámara junto a un proyector, la cámara fotografiaba a razón de tres veces por segundo lo que ocurría en la sala desde que se daba la señal de inicio hasta que se pisaba el sensor de la salida. El proyector por su parte se mantenía proyectando siempre la última imagen tomada, de manera que en cada imagen fotografiada por la cámara se registraba también la imagen proyectada, formándose así una estela con el transitar del público en la sala.

En uno de mis trabajos de arte y código de este mismo año, cuarto de grado, probé a hacer una aplicación que hacía algo parecido, superponiendo una imagen tomada a la que se había tomado anteriormente desde la web cam. Con las opciones que hoy en día nos ofrece Processing resultó increíblemente más sencillo y el resultado era más realista, pero perdía la poesía del proyector y la solemnidad del espacio reservado para la instalación.

Para el TFM me decidí por la realización de un libro de artista, por aquello de experimentar todo lo posible con lo que no estaba familiarizada y aprovechar al máximo el máster. En el libro titulado *Mentiras de Papel* procuré infiltrar el mayor número posible de juegos de todo tipo. Incluí despleables, flipbooks, troquelados, textos ocultos, transparencias... Hasta la explicación de cómo hacer una caja de papel origami. Todo ello combinado con un contenido muy íntimo y personal. Daba como resultado una pieza, que aunque pecaba un tanto de cursi, resultaba muy cercana.

Fotografías de algunas de las páginas del libro de artista *Mentiras de Papel*, 2007.







### 3.2. "JUGABLE" COMO PREMISA

En este apartado me centraré en aquellos proyectos realizados en los últimos cuatro años, durante el grado en *Bellas Artes*, que ya estaban orientados en la búsqueda de esta unión entre el juego y el arte. He hecho una selección de aquellos trabajos que mejor me han servido en este proceso.

Es importante señalar que cuando empecé el grado en Bellas Artes ya era un poco más consciente de lo mucho que me interesaba la búsqueda de lo lúdico, a cualquier trabajo que se me proponía añadía siempre una premisa más, que fuera jugable. Así fue como ya en primero, en cuanto se me ofreció la primera posibilidad de hacer un trabajo libre, en *Fundamentos del color y la pintura*, me dispuse a hacer un cuadro-juego.

En este caso el juego consistía en un laberinto, como aquellos que encontramos en las páginas de pasatiempos infantiles de las revistas. El cuadro representaba un laberinto cuadrulado, en una imagen que quería evocar al pixel sin serlo del todo. En el centro había pintado un jardín, recordando a los laberintos de los grandes jardines de Europa central. El soporte del cuadro estaba hecho de madera, y el laberinto estaba replicado al otro lado de la madera en cartón pluma.

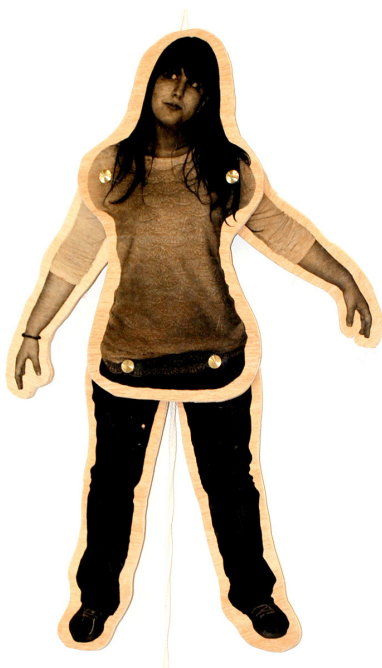
Para transitar por el laberinto coloqué un peón de ajedrez con un imán en la cara visible del cuadro, y otro imán en la parte posterior, de manera que si el peón chocaba con un muro pintado el imán de la cara posterior chocaría con el cartón pluma impidiendo su paso. El soporte tenía también una especie de repositorio inferior al que caería el imán en caso de separarse el peón, y otros dos en los laterales, uno para iniciar el juego, y otro donde caería el imán posterior en caso de ganar, para volver a colocarlo al inicio. El resultado en aquel momento del grado en el que aun no controlaba demasiado sobre técnicas pictóricas no fue del todo de mi gusto, pero la jugabilidad era indiscutible.

Hasta llegar a segundo no tuve muchas más opciones de hacer un proyecto enteramente personal hasta llegar a *Escultura II* con Sara Vilar, todavía recuerdo aquella asignatura como una de las que más despertaban la creatividad de los alumnos de todas las que he tenido durante el grado. En ella se nos ponía una premisa, normalmente de tipo conceptual, y a partir de ella podíamos trabajar a nuestro antojo. Empecé tímidamente haciendo un caleidoscopio con pajaritas de papel y una terrible rotonda giratoria que pretendía hacer las veces de traumatopo.

Logré mejores resultados con las propuestas sobre el cuerpo humano. Primero nos propuso crear una obra de carácter fotográfico utilizando nuestro propio cuerpo como soporte. En esta propuesta presenté una títere de madera sobre el que transferí fotos de mi propio cuerpo. Con la pieza hacía una reflexión sobre el propio cuerpo como objeto lúdico, en todos sus aspectos. Desde niños, haciendo volteretas, palmadas, corriendo, saltando, trepando a cualquier cosa, nuestro principal juguete es nuestro propio cuerpo. Cuan-



Imágenes del proceso de producción del cuadro laberinto, 2010-2011.  
Caleidoscopio de pajaritas de papel, 2011.



Tirere de madera, autorretrato, 2011.

Escenario y personajes del juego para dispositivo móvil *juEgos*, 2012.

do nos vamos haciendo mayores esta frase toma unas connotaciones más sexuales casi sin quererlo, pero no deja de ser cierta, digamos el número de juegos posibles aumenta. Para ello me mostraba a mí misma como un juguete colgado en la pared que cualquiera podría manipular, pero cuyas acciones estaban limitadas por la forma en la que yo misma lo había construido. Se trababa de una pieza alegre, fresquita y transparente, sin ningún significado oscuro oculto.

Los siguientes dos trabajos, *Perfiles* y *Hormigas* los incluyo enmarcados en el proyecto, así que estarán descritos con más detalle en apartados posteriores. Con estos dos trabajos había encontrado una forma de proceder, un esquema mental a seguir que me servía por lo menos para discernir lo que tenía que ver con mi propio trabajo y lo que no. Los dos eran juegos, de carácter serio, pero ni oscuro ni dramático, aunque el tema lo fuera un poco. Y estaban respaldados por trabajos plásticos muy diferentes.

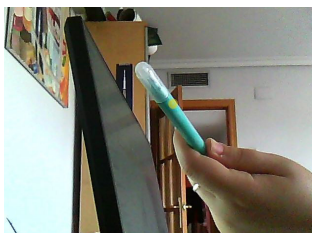
La mayor parte del tercer curso la dediqué a aprender técnicas nuevas, pero aun hubo cabida para los proyectos personales. Como proyecto final para la asignatura de *Animación bajo cámara y stop-motion*, junto con los compañeros Miguel Hernández, Yolanda Benalba y Nayara Artero, hicimos un juego para teléfono móvil, cuyo público objetivo eran los más pequeños de la casa. Hicimos un corto utilizando *stop-motion* en el que un niño jugaba con sus amigos imaginarios al escondite, y en determinado momento todos ellos desaparecían, el juego consistía en pasar el dedo por la pantalla, e ir encontrando a los distintos personajes. Entre nuestros parientes menudos triunfó.

Siguiendo en la línea en *Libro de artista*, con la compañeras Nayara Artero, Loredana Obreja y Ana Fdez, hicimos *Consejos para recién llegados*, en el que dábamos recomendaciones a los más pequeños que a nosotras nos habría gustado saber desde niñas. La editorial de la UPV tiene disponible un libro infantil, sólo uno, que es repuesto por otro diferente una vez se termina la tirada, este libro está a la venta y además es obsequiado a todos los miembros empleados de la UPV cuando son padres. Nuestro libro está diseñado para ser una opción cuando se dé el caso de que tengan que hacer tirada nueva.

Todas las ideas del libro, así como la paleta de colores, la tipografía a usar y algunos aspectos generales del tipo de gráfica, se decidieron entre todas, pero cada una escribió y diseñó en contenido de sus propias páginas. Así que volvía a tener libertad para hacer de ellas un juego. Mi parte consta de tres láminas, en una de ellas podemos ver las piernas de una niña en un baño, y el suelo que la rodea. Este suelo está basado en las baldosas del baño de la casa en la que crecí, y bajo ellos se encuentran los motivos que yo veía cuando me ponía a buscar dibujos en ellas.

En la segunda, haciendo un pequeño guiño al trabajo hecho en animación, dibujé a un grupo de niños celebrando un cumpleaños, todos disfrazados de monstruo, el juego aquí era el de descubrir cuál de ellos no era un niño, si no el amigo imaginario de uno de ellos disfrazado de humano. En la





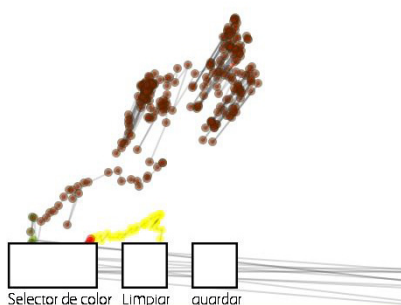
tercera dibuje un gran gusano, como si de un metro se tratara, y junto a este dispuse un desplegable en el que se puede ver a través de la luna delantera del metro a todos los que viajan en él.

Llegados a este curso estaba más centrada en la propuesta que había pensado para el TFG, y la elección de asignaturas estaba más orientada a ayudarme a llevarlo a cabo.

Aunque ninguno de los proyectos realizados en *Arte y código* están circunscritos como parte de este proyecto, sí me vi inspirada por todo lo que aquí hice para llevar a término la propuesta de *Quiz* que veremos más adelante. Además puse en marcha el *Paint Wireless*, una aplicación con la que podemos dibujar sin tocar el ordenador, y que este verano pondré en uso en alguno de los talleres de verano para discapacitados psíquicos que la empresa *IDEOCE*, seguramente como una de las actividades de la escoleta de verano que hacen en su propia sede, en Valencia..

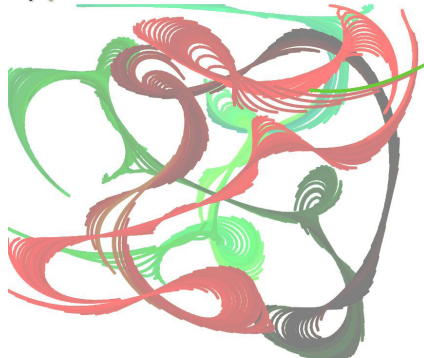
Describiré las últimas dos piezas realizadas durante el grado, también en el capítulo siguiente. Se trata de *Atravesar* hecha como proyecto en la asignatura de Instalaciones, y *Preparados, listos... Luz!*, proyecto de Medios interactivos y electrónicos del cual surgió también *Sambori*.

En la actualidad me encuentro desarrollando una serie de piezas online que no entrarán dentro de este Trabajo Final de Grado pero seguirán la misma línea, se trata de páginas en blanco en las que el espectador tendrá que intervenir. La idea es que al llegar a ella la página esta no es más que una página vacía, pero cada pequeña interacción del usuario se verá reflejada gráficamente llenando ese espacio. Se puede ver un ejemplo en el inicio de mi portfolio en <http://marinaportela.es>



### Marina Portela Ruiz - Portfolio

Pulsa en la muñeca para ver más cosas, mueve el ratón para distraerte un rato. Bienvenido.



Arriba captura de la aplicación *Paint Wireless*, 2013. Lo que captura de cámara y como se refleja en el ordenador.

Debajo captura de la página de inicio de mi portfolio.

### 3.3 PIEZAS ENMARCADAS EN EL PROYECTO

En este capítulo vamos a desarrollar el bloque central de mi investigación, todas las piezas que veremos a continuación están hechas durante el grado, y ya estaban centradas en la búsqueda del juego como herramienta común, en acercarse al espectador por encima de todo, sin perder en contenido ni en poética.

A continuación haré una descripción pormenorizada de cada una de las piezas, intentando explicar cómo, por qué y para qué fueron realizadas:

#### 3.3.1 *Perfiles, 2012*

La primera pieza que he seleccionado para formar parte del conjunto que forma este proyecto es *Perfiles*. Realizada como trabajo para la asignatura *Escultura II*. La propuesta proponía trabajar teniendo como tema central el cuerpo humano.

La obra trata sobre nuestro yo virtual, cómo lo relacionamos con nosotros mismos y lo alteramos para adecuarnos a nuestros ideales. Reflexiona sobre la forma en que afectan nuestros avatares en la imagen residual que dejamos al morir.

El formato de juego de mesa permite al espectador interactuar directamente con la obra decidiendo en cada momento la expresión del rostro, y tener la posibilidad de encontrarla y dejarla de una manera diferente. Un cambio que alude a nuestras propias transformaciones.

Durante el primer año de grado recibí una mala noticia, un amigo cercano decidió suicidarse. La familia nos pidió que nos encargásemos de hacer algo con su Facebook. Vivía fuera de su ciudad, y mucha gente que estaba acostumbrada a no verle en persona con frecuencia, empezaba a notar que tampoco se le podía localizar por internet, y preguntaban por él en su muro y colgaban algunas fotos... El caso es que querían parar aquello, pero tampoco querían borrar el recuerdo que suponía aquel espacio para ellos.

No hace falta investigar mucho para ver que Facebook ha contemplado esta opción. Tiene dos modos de cierre por defunción (O los tenía hace dos años) Uno es el cierre total, el usuario desaparece por completo del sistema como si nunca hubiera pasado por allí. Y el otro es dejar su muro en modo "homenaje". De esta manera se puede consultar pero ya no se puede escribir ni colgar nada, ni se le puede etiquetar en fotos nuevas. Y el resultado es curioso, porque lo que queda al final en su muro son todas aquellas cosas que escribieron aquellas personas que no sabían que había fallecido. Lo que queda más a la vista son los comentarios que otras personas le hicieron. Como si él hubiese puesto a nuestra disposición todo aquel material para que los demás lo manipulásemos.



Primer plano de algunas piezas de *Perfiles*, 2012.  
Foto de Noemi Novella

Esa idea permaneció en mi mente durante mucho tiempo, lo que somos y lo que queda de nosotros tras irnos, y como eso también está cambiando tanto con la era digital. Tengo que reconocer que me encantaría que todo esto hubiera existido desde hace mucho tiempo, y poder ahora consultar las viejas fotos de mis tatarabuelos (por ejemplo) y hacerme una idea de cómo eran, o ver que imágenes y que frases había colgado de ellos la gente que les conocía con la que yo pudiera formarme una idea más o menos acertada.

De todo este batiburrillo de reflexiones nació *Perfiles* Un autorretrato más o menos fiel que fuera manipulable, tergiversable y continuamente cambiante. Como el que yo vi de mi amigo durante aquellos días tan tristes que siguieron a su desaparición.

Scott Kim dice en la conferencia de la TED del 2008 *The art of puzzles*<sup>9</sup> que “Considero que los puzzles son una forma de arte. Son muy antiguos. Van tan atrás en el tiempo como la historia escrita. Es una forma de arte muy pequeña, como un chiste, un poema, un truco de magia o una canción, una forma de arte muy compacta. Lo malo es que son desechables, son para divertirse. Pero lo bueno es que pueden lograr algo más y crear una impresión memorable.” Vi esta conferencia tiempo después de crear esta pieza, pero me parece una justificación preciosa sobre porqué decidí hacer un puzzle en este caso.

La intención es que la gente, al situarse delante de las piezas tenga ganas de tocarlas y manipularlas. Y que cada persona que llegue modifique la cara que haya dejado la anterior, de manera que la pieza esté en constante cambio y dependa de la creatividad del espectador.

De alguna manera cuando veo el trabajo de perfiles pienso en aquellas piezas de ilustración de Isidro Ferrer, o las fotos de Chema Madoz, en las que mediando una dulce imagen cargada de imaginación se pueden criticar los hechos más duros. En esta ocasión en lugar de una ilustración se trata de un juego de mesa, como el que nos podía traer Zbigniew Libera en su *Campo de Concentración de Lego*, aunque recordando más al hundir la flota que Natalie Raum hace para conmemorar el hundimiento del Titanic.

Una vez expuesta decidí colocar un hashtag para que la gente pudiera hacer fotos a la pieza conforme la había dejado y colgarla en internet asociada a un nombre, de manera que se pudiera hacer una galería con ellas.

La mayoría de juegos de mesa interactivos de carácter artístico que conozco se basan en la tecnología de *Reactivisión*. Desde, claro, la *Reactable* de Sergi Jordà, Marcos Alonso, Günter Geiger y Martin Kaltenbrunner, o la mágica pieza de sombras aumentadas de Joon and Moon, pasando por un sin fin de piezas sobre visualización de datos. En este caso quería hacer un juego de mesa, lo más tradicional posible. Con su caja, sus fichas, y su movilidad. No quería depender de un gran equipo multimedia, ni de cámaras, ni proyectores... Buscaba una pieza muy personal que no resultara fría en absoluto.



*Ice watch*, Natalie Raum.

*Augmented Shadow*, Joon and Moon.

9 KIM, S. *The art of puzzles* [Multimedia] EG 2008 · 11:49 · Grabado Dec 2008 [consulta: 2014-07-04] <[https://www.ted.com/talks/scott\\_kim\\_takes\\_apart\\_the\\_art\\_of\\_puzzles](https://www.ted.com/talks/scott_kim_takes_apart_the_art_of_puzzles)>

Imágenes de algunas piezas antes y después de ser pintadas.



Para crear las piezas se han usado siempre como molde los mismos vasos desechables de papel. Se cortan la mitad de los vasos por la misma altura, se rellenan de alginato y se colocan rápidamente sobre la zona a registrar. En mi caso, ojos, nariz y boca. Previamente colocadas en la posición que queremos retratar, y manteniendo este gesto fijo durante unos segundos.

Seguidamente sacamos la pieza de alginato ya solidificado, y lo introducimos en uno de los vasos sin cortar. Después rellenaremos todos los vasos enteros con escayola. Cuando esta se seque retiramos los cartones y obtenemos las piezas en blanco.

Para el color usé acuarelas y acrílicos muy diluidos, en varias capas. Finalmente recorté trozos de fieltro del diámetro de las piezas y los pegué en la parte inferior de los mismos.

Para hacer la caja busqué ilustraciones de cajas de juegos antiguos hasta encontrar una de un parchís que me pareció adecuada y la retoqué para que en lugar del parchís estuvieran jugando con mis piezas, cambié el nombre e incluí un hashtag asociado al juego. La imprimí al tamaño de una caja de un juego viejo que tenía por casa en el que cabían todas las piezas menos cuatro.

Forré la caja con esta imagen, la rellené de esponja y cubrí esta con fieltro de un color parecido al de mi piel. Sobre este dispuse todas las piezas.

Para exponerlo coloqué la caja abierta con todas las piezas organizadas sobre ella. La tapa de pie a modo de cartel y delante de este las cuatro piezas que no caben formando un “perfil” de ejemplo.





Imágenes del proyecto en el  
"Hybrida, art festival 2013"  
Las Naves, Valencia.  
Foto tomada por el colectivo  
Señor Cifrián.

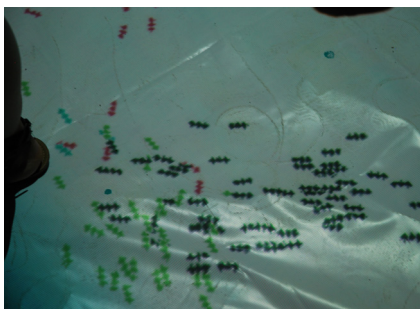
A la derecha la pieza en el momento de  
presentarla en clase.  
Foto de Noemi Novella



Ha estado expuesta ya en un par de ocasiones, y la parte interactiva funciona bastante bien, casi todo el mundo juega con la pieza y la modifica constantemente.

La parte de las fotos, por otro lado, no funciona tanto, se subieron algunas fotos, pero para ser honestos casi todos los que lo hicieron eran amigos cercanos. Creo que hacer una foto y colgarla aun supone demasiado esfuerzo participativo cuando la obra es de alguien anónimo, funciona mucho mejor si se coloca un sofá en el que se ha sentado Warhol<sup>10</sup>, por ejemplo. Lo más importante es que se trata de la pieza con la que definí mi modo de proceder ante un proyecto personal.

10 En referencia al éxito obtenido por la fundación bancaja en su propuesta de que la gente se fotografiara en el sofá de Warhol para conseguir una réplica exacta del mismo. Valencia, 2012-2013 [consulta: 2014-07-04] <<http://www.fundacionbancaja.es/sorteowarhol/>>



### 3.3.2 Hormigas, 2012

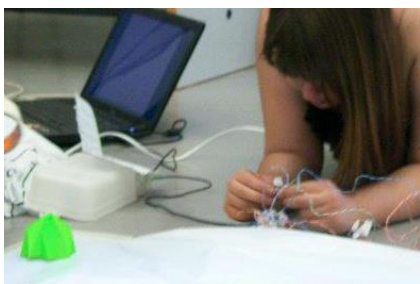
Poco después llegó la propuesta para el proyecto final de esta misma asignatura (*Escultura II*), en esta propuesta se nos invitaba a trabajar con el concepto de nido desde una perspectiva personal, que transmitiera nuestra posición frente a la escultura. En esta ocasión propuse *Hormigas*, en el que hablando sobre la ansiedad y el sentimiento de agobio coloqué una alfombra en blanco frente al espectador, que no tomará color hasta que este se coloque sobre ella. Momento en el cual una serie de hormigas de diferentes colores saldrán de sus nidos persiguiendo al espectador, de manera que este podrá hacer una especie de dibujo de hormigas a su paso, hasta ser completamente cubierto por ellas.



*Hormigas* es básicamente una alfombra interactiva. No tan vistosa ni impresionantes como las *Tapis Magiques* del artista francés Miguel Chevalier, ni tan divertidas como *ØNSKEBRØNN*, y *Words all over* de phase7, o *Photo Rain* de Refraction Labs, ni tan sugerente como el techo interactivo de *Cloud Pink* de Everyware. Pero a su manera entretenida. La pieza estaba pensada para un lugar oscuro, así que no podía depender de una cámara para obtener la posición de los participantes, aunque viendo el espectacular funcionamiento de *Cloud Pink*, bien podría hacerlo hecho con una kinet, como hacen ellos.



La alfombra, de hule blanco, está conectada mediante unos sensores a un teclado usb. Cada posición de la alfombra se traduce mediante el teclado por un carácter alfanumérico. El programa a su vez, identifica el carácter recibido con una posición en pantalla permitiéndome seguir de forma virtual al paseante.



A simple vista la pieza es una alfombra de plástico blanco, con cuatro semicírculos de origami de colores primarios, colocados uno en cada esquina, todo esto iluminado por una luz blanca. Sin interacción en la pieza no ocurre absolutamente nada más.

En el momento en el que el espectador se coloca sobre la alfombra, una serie de puntitos a modo de hormigas del mismo color que su respectivo "hormiguero" comienzan a viajar por la alfombra en busca del visitante. Cada vez más hormigas que persiguen y rodean al visitante hasta que salga de la alfombra. En el momento en el que este sale de la alfombra las hormigas volverán a sus pertinentes hogares. Si más de una persona entra a la alfombra a la vez (o si el que está separa mucho las piernas) las hormigas acudirán a la persona que tengan más cerca. Al estar colocados los hormigueros en las cuatro esquinas, es improbable que haya un visitante al que no persiga ninguna hormiga.

Exposición en clase de la pieza *Hormigas*, 2012.

Este movimiento de las hormigas sirve de alguna manera de reflejo, podemos ver nuestro propio movimiento dentro de la pieza, creando una relación entre el espacio y nuestro propio cuerpo. Olafur Eliasson, hablando sobre su instalación *The weather project* cuenta que se sorprendió de que en su intento por hacerla una pieza tangible, con los espejos y la bruma, la luz... y



*Hormigas*, 2012. montaje y puesta en marcha.



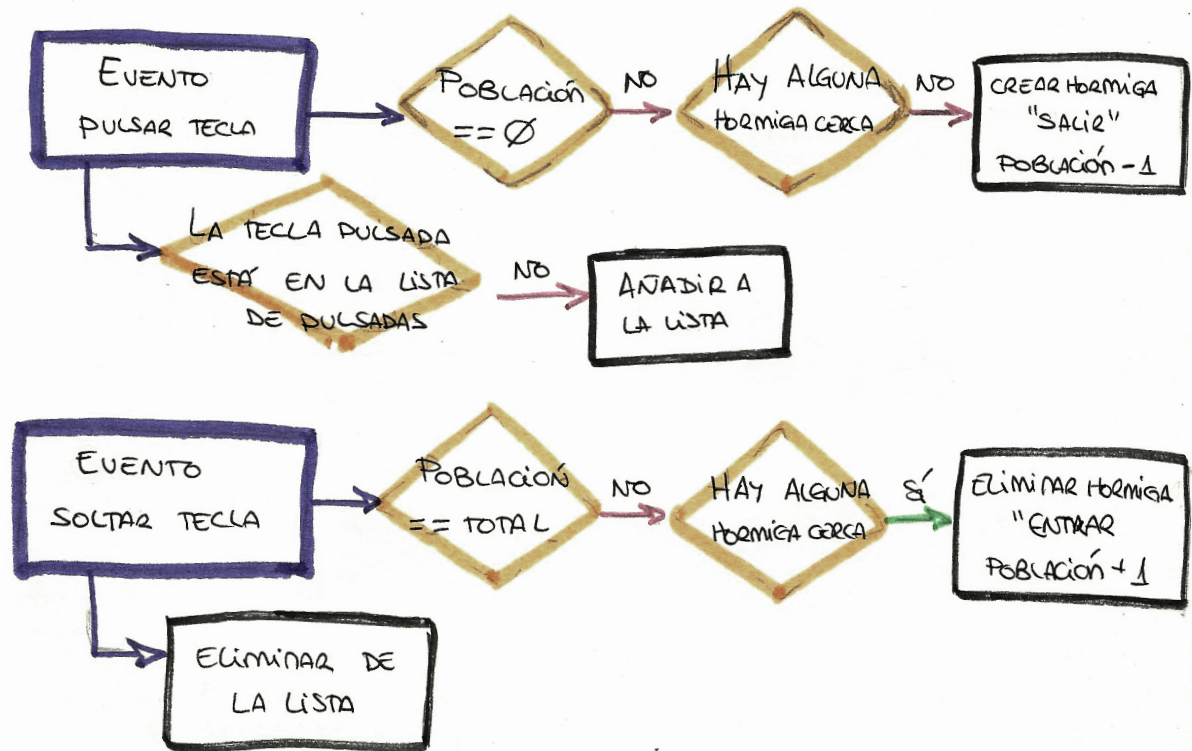
el público se tumbaba y se observaba dentro del contexto utilizando aquel espejo, sintiéndose atraído por esa posibilidad de situarse a si mismo dentro de la pieza. Ese reflejo como el que se obtiene en *Iris* de Hybe, o *The wooden mirror* de Daniel Rozin, atraen de alguna manera al espectador.

En este caso existían tres partes claramente diferenciadas de trabajo, la programación de la aplicación que gestionaría el movimiento de las hormigas, la creación del dispositivo que actuaría como sensor, y la construcción de la parte externa de la pieza.

Para convertir los sensores en las teclas del teclado debemos desmontar este último y conservar el circuito interno, soldamos el cableado de los sensores al circuito del teclado. Ahora solo tiene tantas teclas como sensores hayamos puesto. Es importante antes de soldar seguir el circuito de plástico que va en el interior del teclado para saber qué carácter irá asociado a nuestros sensores. Dentro de lo malo, siempre podemos probarlo una vez montado, pero es conveniente que no coincidan con ningún carácter especial para no tener problemas con la programación. A continuación se unen a los sensores. El teclado irá conectado por USB al ordenador en el que se ejecutará la aplicación de las Hormigas.

La manera de mover las hormigas estaba enteramente orientada a que realmente parecieran seres vivos. No puedo hablar de vida artificial en el arte sin nombrar a los espectaculares seres autómatas que crea Theo Jansen, o los simpáticos y tímidos *Pygmies* de Pors y Rao, Pero yo buscaba un efecto más parecido al generado por Cuppetelli y Mendoza en *Nervous Structure*, un movimiento que resultara natural, parecido en este caso al que hacen las propias hormigas, como el de los insectos que se mueven por los folios de la máquina de escribir de Christa Sommerer y Laurent Mignonneau en *Life Writer*.

Esta aplicación está programada en qt, el código integro de la pieza se encuentra en el apartado de Anexos. Cada hormiga es un objeto independiente. El código de la aplicación puede consultarse en los anexos, el sistema funciona como indica el diagrama siguiente:



HORMIGAS: CADA UNA LLEVA SU PROPIO TIEMPO DE ITERACION  $\pm$  RANDOM PARA NO GENERAR UN RITMO, EN CADA ITERACION:

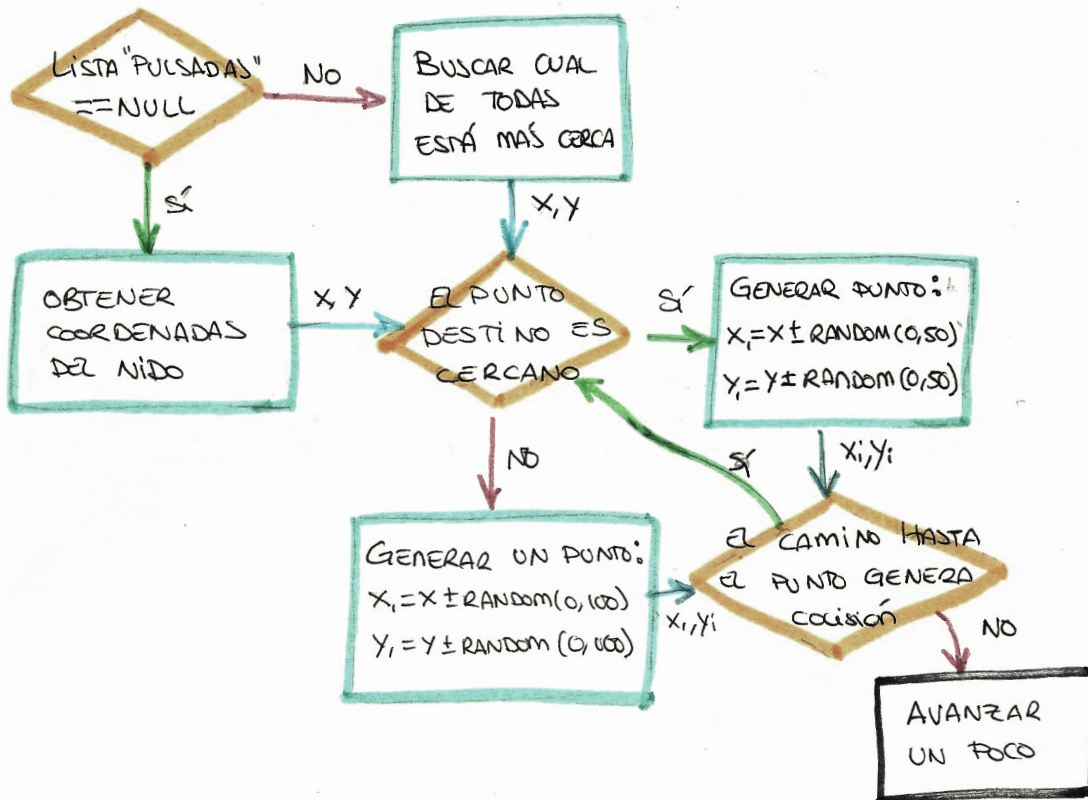


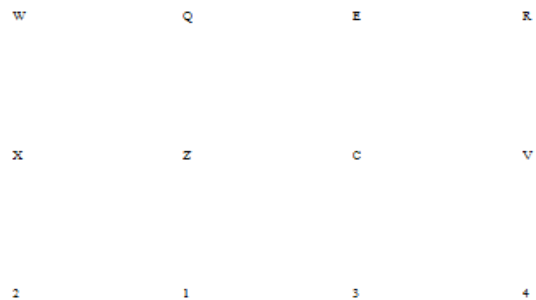
Diagrama de funcionamiento del programa que gestiona las hormigas

Los hormigueros están hechos con sencillos pliegues de origami formando una semicircunferencia de papel con un orificio en la parte superior.

Las dimensiones de la pieza serán variables, dependiendo del lugar donde se monte y de la capacidad del proyector. Se deberá hacer una prueba con el proyector ya colocado para tomar las medidas del área máxima que proyecta y saber dónde deberán colocarse los sensores. Para ello deberemos proyectar la siguiente. Una vez hecho esto ajustaremos las medidas de la alfombra a las obtenidas, y la colocaremos sobre los sensores, para finalizar colocamos los cuatro hormigueros cada uno en su esquina correspondiente.

Plantilla para colocar los sensores en el suelo.

NOTA.- En este caso sirve también como demo para ver el movimiento de las hormigas. Si se pone el ratón en la imagen y se pulsán las teclas que aparecen indicadas, las filas de hormigas deberían ir hacia ellas. Es una versión en Flash menos potente que la original. Pero puede ayudar para hacernos una idea.



En hormigas hago uso de los insectos para hablar de la ansiedad, de cómo continuar por el mismo camino y aguantar no soluciona el problema, sino que lo agrava. Todos asociamos ese paso de a poquito, lento y ligeramente aleatorio que mueve a las hormigas con cierto malestar dubitativo. Cuando estás sobre la alfombra el cuerpo se cubre literalmente de estos pequeños seres de colores siguiendo ese ritmo. Pero al estar formados por pequeñas manchas de luz geométricas y de colores planos no genera ese malestar. Con mucha imaginación puede llegar a dar un poco de grima, pero no creo que sea lo usual. Al entrar en la alfombra quieres que pase algo, y lo que quieres es que las hormigas te cubran y te persigan, y dibujar con ellas sobre la alfombra. Se consigue esa intención interactiva y creativa que tiene el trabajo.



*Video de la sesión de pruebas antes de la inauguración de la exposición. Atravesar, 2013.*



### 3.3.3 Atravesar, 2013

Este proyecto, elaborado junto con otras tres alumnas de la asignatura: Natali Brzoza Szwarzman, Eleni Liaskou, y Ana Martínez, forma parte de la exposición *Casos Favorables >>> Casos Posibles* en la que se nos daba la oportunidad de trabajar en cualquiera de las salas del Palacio Ducal de Gandía.



Existe en el palacio una sala que no es transitable, la sala de las águilas, que sólo puede verse desde sus dos extremos. Con este proyecto quisimos hacer que los visitantes pudieran de alguna manera atravesar la sala volando. Introdujimos en la sala un gran marco que no desentonara con el resto de la estancia, y ofrecimos al público visitante unos cuantos aviones de papel. Con estos dos formamos un juego en el que se incitaba a hacer diana en el marco para conseguir un efecto luminoso en la sala a modo de recompensa para quien lo consiguiera. De esta manera esta sala volvía a cobrar vida, y se inundaba de huellas de la gente, esta vez a modo de manto de aviones de papel.



Queríamos que el resultado final tuviera ese aspecto inocente y elegante que le dan los globos blancos a cualquier sala donde se instale *Scattered Crowd* de William Forsythe. De hecho inicialmente planteamos lanzar globos, pero era poco realista. También pensamos en pompas de jabón, como las burbujas digitales que se general al soplar en *Cale aux bulles* de Joelle Aeschlimann, Pauline Saglio y Mathieu Rivier.



Pusimos en uno de sus extremos un gran número de aviones de papel con los que poder atravesar la sala de forma simbólica, y dejar huella en ella. La idea era hacer una propuesta moderna y juguetona cuyo fisonomía conjuntara de forma amable con el entorno en el que se encontraba.

La pieza consta de un marco, como si del de un espejo de pie se tratara, adornado con material de desecho pintado de dorado, haciendo la ilusión de formar piezas de adorno barrocas. (Esta idea la tomamos de la pieza de nuestra compañera Zuzana Hegedusová, que decoró una de las puertas del palacio de este modo) el marco colocado en el centro de la sala reacciona cada vez que uno de los aviones de papel que es lanzado atraviesa su interior.

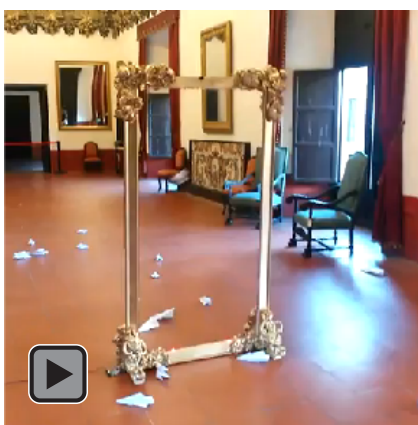


El marco, en su parte superior, lleva instalado un sensor de infrarrojos que detectaría la presencia de cualquier cosa que pase por su interior. Este sensor a su vez está conectado a una placa Arduino que recoge la señal y la envía a los diodos LEDs repartidos por todo el marco, y generando con ellos un pequeño juego de luces.

Los aviones se encuentran dispuestos en una mesa con las instrucciones del juego propuesto, dentro de una caja dorada, del mismo estilo que el marco.

Por una parte debíamos construir un marco hueco con patas que se mantuviera de pie sin problemas, se montara y se transportara sin mucha dificultad y fuera igual de vistoso por ambos lados. La arquitecta del equipo, Natalie, diseñó la siguiente estructura: Posteriormente hicimos los adornos con vasos de plástico, cajas de cereales, bolsas, tapones de botellas, etc. uniéndolo-

*Atravesar, 2013.* Día de la inauguración de la exposición *Casos Favorables >>> Casos Posibles*, en el Palacio Ducal de Gandía. Fotografías de Tomáš Hercog y Ana Fdez.



*Atravesar*, 2013.

Detalle del marco.

Foto realizada desde el marco durante la inauguración.

NOTA.- Vídeo de la pieza en el momento de las pruebas, antes de la inauguración.



los con silicona y deformándolos con pistolas de calor, después los pegamos sobre los listones del marco y los pintamos con espray dorado, quedando así: Mientras debíamos montar el circuito que repartía todos los diodos LED a lo largo del marco y los conectaba con el Arduino, y a este con el sensor. Estaban montados en tiras igual de largas que los listones, para poder colocarlos en medio en el momento del montaje.

También en paralelo debía de hacerse el trabajo de programación, la placa de Arduino debía de contener un código que estuviera alerta de los datos que recogiera el sensor y diera la señal a los LED cuando tuvieran que lucir, el esquema de funcionamiento del programa era el siguiente:

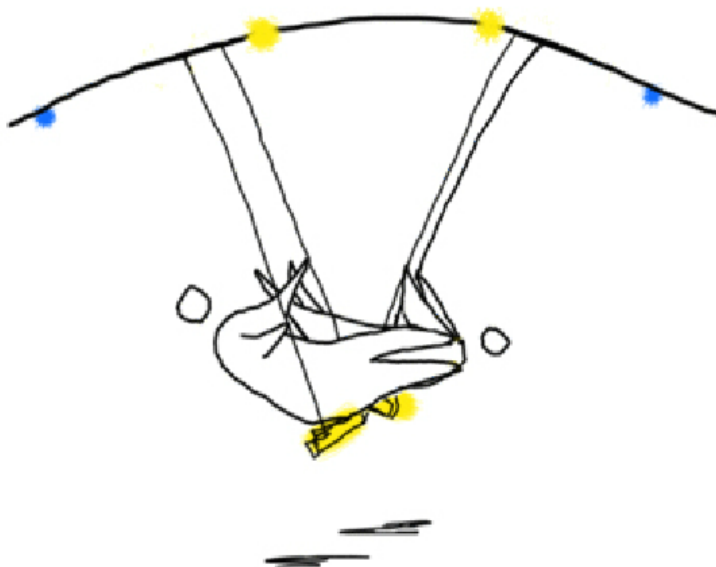
Existe una pieza de ART+COM, *Duality* en la que al pasar por un puente generas ondas en el agua que atraviesa, como si anduvieses por el agua, nosotras buscábamos ese mismo dejar huella por donde realmente no hemos estado. Y una interacción divertida con el espacio, como la que genera *Human Allergy* de Hugo Martínez Torno al estornudar cuando alguien pasa por delante. Con un objeto familiar, lo bueno del avión de papel es que raro es encontrar a alguien que nunca haya hecho alguno, y se dan muchos los casos de personas que tienen una forma peculiar de hacerlos, que identifican con sigo mismas.

Al principio dudamos si la gente que pasara por allí se pararía a leer las instrucciones de nuestro juego, o si se pararía a lanzar un avión. Pero durante el montaje no fueron pocos los visitantes que al ver el marco allí y el montón de papeles, ya intentaron acertar en la diana. Una vez montado lo probamos con algunos miembros del personal del museo, que se mostró muy participativo. Y los pocos aviones que quedaron como rastro de aquellas pruebas resultaron ser suficiente como instrucciones para la gente que llegó a la inauguración. Antes y mientras explicábamos la pieza, ya había gente lanzando aviones, y pude ir unas cuantas veces más y comprobar que cada vez el suelo estaba más blanco, incluso había gente que firmaba el avión antes de lanzarlo. Para nosotras eso era todo lo que buscábamos. Para nosotras funcionó.





Infografías diseñadas por Mar Albiol para la pieza *Preparados, listos...Luz!*.



### 3.3.4 Preparados, Listos... Luz!

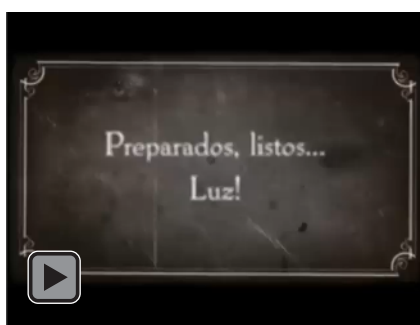
Preparados, listos... Luz! es una pieza interactiva diseñada en conjunto con Mar Albiol y que tenía como fin iluminar uno de los puntos oscuros de Valencia. Mar forma parte del colectivo *Carpe Via* que organiza todos los años el evento *Viajes al interior de la noche* para advertir sobre el peligro que pueden suponer de algunas zonas de Valencia durante la noche para las mujeres. Para ello hace un ejercicio de reactivación temporal de la zona haciendo alguna acción de carácter artístico y transforma el espacio en una zona de cena y fiesta. Ella me propuso hacer una pieza que pudiera funcionar en este contexto. Aquí teníamos una premisa más, querían que la pieza tuviera luz.

Era una ocasión interesante para probar una pieza con más público que no podía dejar escapar. Después de descartar muchas ideas se nos ocurrió que podíamos hacer un par de columpios interactivos que formarían un juego entre ellos. Estos columpios se podrían quedar en el "punto oscuro" hasta que alguien se los llevara o hasta que se terminaran las pilas, y serviría para alargar así el tiempo de vida de su propuesta. Por cuestiones técnicas la idea que planteamos no era la idónea para el viaje de este año. Pero les gustó y es la pieza que usarán en el viaje del 2015.

Existe un gran número de obras que usan elementos comunes de parques infantiles, como *Suspended* de Mona Hatoum lleno de columpios como el *Event of a thread* de Ann Hamilton. Los toboganes de Carsten Höller, o su tiiovivo. En *Neon Swing* de Su-Mei Tse encontramos un columpio luminoso, al que no podremos subir ya que está hecho de tubos fluorescentes. Y el columpio colgado de dos globos *Untitled (swing)* de Sorota Bucz, que ella misma usó para su performance *Balançoire* en la que lo deja volar. El uso de columpios nos devuelve piezas que nos retraen a la niñez y nos resultan acogedoras.



*Neon Swing*, Su-Mei Tse.



*Preparados, listos... Luz!*, 2013.  
Funcionando

NOTA.- Vídeo de la pieza.

La instalación está compuesta por dos columpios enfrentados, y consiste en un juego en el que los usuarios de los dos columpios deben chocar las manos entre sí mientras se columpian.

Aunque se preveía el uso de la pieza durante un evento concreto, la instalación puede funcionar de modo indefinido en el tiempo. El usuario encontraba las instrucciones de uso en un pequeño panel adherido al asiento del columpio.

El objetivo del juego es conseguir el mayor número de puntos antes de que venza el tiempo de partida. Para esto los jugadores situados en cada uno de los dos columpios deben coordinarse y trabajar en equipo del modo que a continuación se describe:

1.- El juego empieza cuando los jugadores juntan sus manos por primera vez, en este instante las tiras de LED se activan acompañando a los jugadores en su vaivén. Se encenderá la tira más cercana a la posición del columpio en cada caso. Cada uno de los jugadores tiene sus propias tiras de LED.

2.- Para seguir consiguiendo puntos los jugadores deberán volver a chocar sus manos. Por cada choque conseguirán un punto. Deben conseguir 4 puntos antes de que pasen 60 segundos. Cada vez que consigan un punto las luces de los dos se sincronizarán haciendo una ráfaga de luces. 3.- Si los jugadores consiguen superar el reto se hará una última fiesta de luces de mayor intensidad, tras ello las luces se apagarán para que un nuevo juego pueda comenzar. Si por el contrario no lo consiguieran, las luces se irían apagando poco a poco dando lugar, nuevamente, a recomenzar el juego.

4.- Como no deseamos que los columpios se iluminen si no se cumple la interacción, las luces permanecerán apagadas siempre y cuando no se realice un choque de manos.

Los columpios están formados por dos tablas de DM de 25x15x1 cm cada una puestas una encima de la otra, unidas por dos grilletes, uno a cada lado que las unían a su vez con las cadenas de 3 metros cada una y 7 mm de diámetro. Para que los agujeros en la madera que permitían esta unión quedaran más aseados colocamos arandelas en la parte de fuera, un total de 8 por columpio.

Para sujetar el columpio en el soporte ideamos dos opciones con la intención de que se pudiera adaptar al mayor número de lugares posibles. La primera es mediante dos grilletes más por columpio, utilizando la propia cadena como sujeción y estos como cierre. En la segunda opción incluimos hilo de acero trenzado, por si el diámetro de la estructura a la que los sujetamos es muy grande, de forma que el hilo engancha la cadena y este es el que soporta la pieza. Para cerrar el hilo tenemos dos sujetacables por columpio.

Trenzada a una de las cadenas de cada columpio irá un cable que terminará en el guante que, en el estado de prototipo, permitirá que los dos jugadores cierren el circuito.

En cada uno de los asientos, hay además cuatro piezas de vinilo adhesivo con las instrucciones de uso.

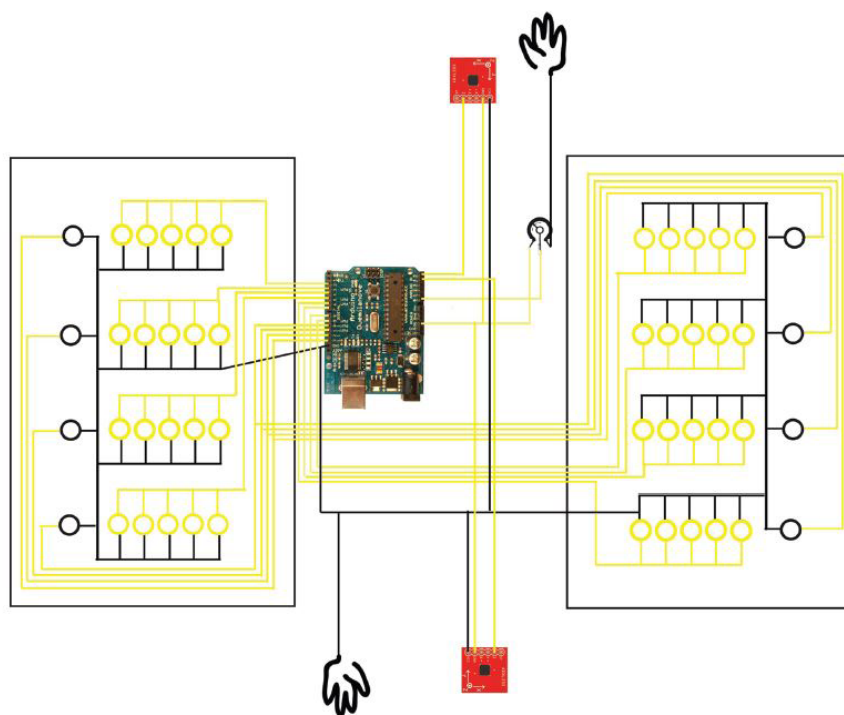
La instalación se apoya en el uso de la placa Arduino, ella nos permite controlar el valor de los sensores en cada momento así como el de los dispositivos de salida. Este Arduino irá conectado con una pila de 9V, para no depender en ningún caso de enchufes. Si dejásemos la pieza en la calle un tiempo indeterminado duraría lo que durara la pila. A no ser que pudiésemos ir cambiándola.

Usamos un potenciómetro a modo de interruptor que nos permite tanto apagar y encender la pieza como controlar el hecho de si los guantes están o no en contacto.

Para saber que inclinación lleva el columpio en cada momento hemos instalado un acelerómetro al principio de una de las cadenas de cada columpio. Las luces que se activan por su movimiento están tejidas a una tela de fieltro



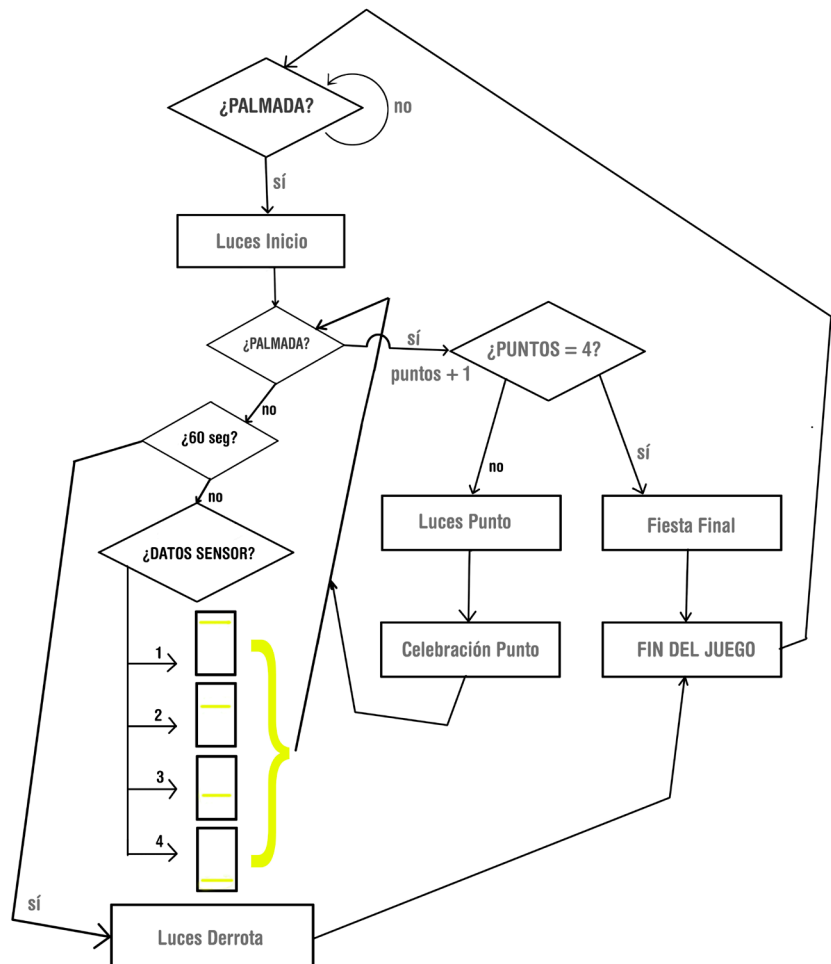
*Preparados, listos... Luz!*, 2013  
Fotos de las pruebas y el montaje  
Esquema de circuitos.



de 2x0,5 m cada una. En cuatro tiras de ocho LEDs, seis blancos y uno verde. Los verdes son los que llevarán la cuenta de la puntuación.

Creemos que podríamos mejorar la instalación si en lugar de dos alfombras para los columpios hiciéramos una con los dos. La idea de hacer dos era para ser más flexibles en cuanto a la colocación de los mismos, pero al fin y al cabo es necesario que los participantes puedan tocarse las manos, así que esa distancia realmente no es tan flexible, y se podría llevar mucha faena adelantada al montaje final. En el modelo del prototipo la instalación in situ duraba unas dos horas que debemos reducir.

Preparados, listos... Luz!, 2013  
Diagrama de funcionamiento.

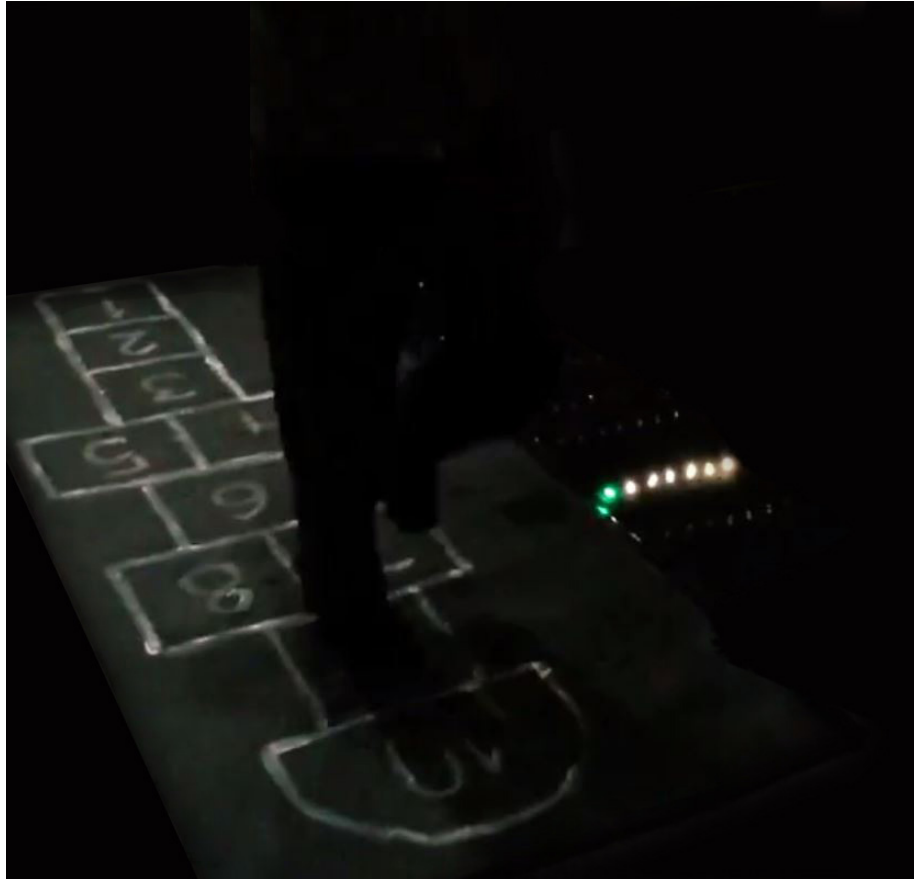


A pesar de los problemas que hemos podido tener pienso que la pieza funciona correctamente, y que cumple el objetivo lúdico, festivo e iluminador que nos marcamos en un comienzo.

Obviamente necesita algunas mejoras en las que ya estamos trabajando. Pensamos que la pieza debería incluir su propio soporte para los columpios, para poder colocarlo en cualquier parte de forma independiente. Y creemos que con el uso de unos transistores podemos potenciar la señal de manera que dejemos de necesitar el uso de los guantes para la señal de contacto, y sea el propio cuerpo humano el que se encargue de esto. Por lo demás estamos deseando que llegue la fecha de colocar la instalación y ver qué ocurre.



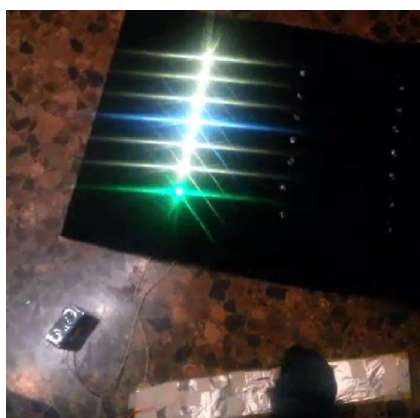
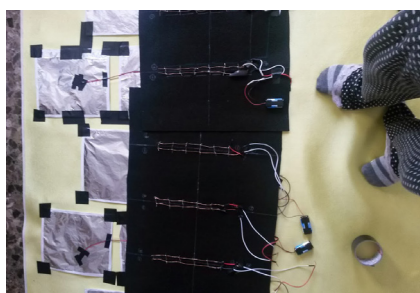
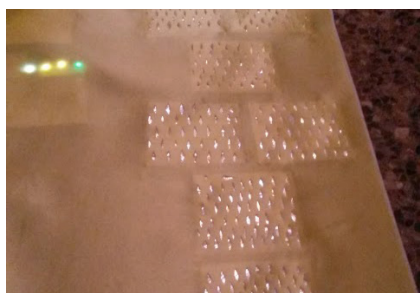
*Sambori, 2014* Fotos de la intervención en el tunel del Viveros con motivo del evento *Viaje al interior de la noche de Carpe Via*



### **3.3.5 Sambori, 2014**

Al parecer Carpe Vía quedó muy satisfecho con el trabajo que habíamos hecho, y me pidió que preparara una segunda propuesta, esta vez sí para el evento del 2014. Propuesta que debía adaptarse a variar cosas: primero, tenía un presupuesto muy limitado (aunque podía usar lo que quisiera del material que compré con Mar para Preparados, listos... Luz!, pero nada de lo prestado por la UPV, adiós al Arduino y a los sensores) segundo, tiempo de desarrollo limitado. Me propusieron participar un viernes cuando el evento tenía lugar el martes siguiente. Y tercero, la pieza debía montarse y desmontarse en cuestión de minutos, porque el tiempo del que disponían para el evento era muy limitado, y del que disponían para desmontar casi nulo. Aun así me parecía interesante participar, acepté.

La filosofía de la misma debía de ser similar a la anterior. Tenemos un lugar oscuro que queremos que se ilumine de forma participativa, Carpe tenía la intención de invitar a un grupo de bailarines, y se les ocurrió que quizás podía hacer una alfombra que se encendiera cuando bailaran sobre ella. La idea de utilizar sensores de presión, similares a los de *Hormigas* me pareció inteligente dados el presupuesto y el tiempo del que disponía para el montaje. Así que tomé esto como punto de partida.



*Sambori*, 2014 Fotos del montaje y prototipos.

La luz ofrece un maravilloso mundo de posibilidades para la interacción, vemos obras increíbles como *River-is* de Art+COM, que utiliza los reflejos para hacer una serie de figuras en el techo de la sala. *Water Light Graffiti* de Antonio Fournau presenta un soporte luminoso en el que dibujar con agua. Olafur Eliasson tiene multitud de piezas maravillosas basadas en el uso de la luz, desde la citada *The weather project*, pasando por *It's a real shadow* que me interesa especialmente por su simplicidad, lo único que hace es poner varias luces filtradas transformando nuestra sombra, o sus *Little sun* que usa en los grandes museos para crear una visita diferente al espacio, en el que cada uno se encarga de iluminar lo que quiere ver, y a su vez ha ideado un dispositivo lumínico que carga con energía solar y usa para ayudar a los países en vía de desarrollo. También están las geniales pantallas de bombillas que Jim Campbell usa para reproducir vídeos en una peculiar forma tridimensional en sus tres *Exploded Views*. O la mágica forma en la que *Candas Sisman* hace viajar la luz a través de un pasillo en *IPO-cle*, sin olvidarnos de *Fireflies on the water* de Yayoi Kusama y muchas otras... En este caso lo que yo proponía era muy limitado, pero servía para su propósito.

Finalmente la pieza sí era una alfombra con sensores, pero estos estaban dispuestos en forma de sambori (rayuela) y estaban conectados a una segunda pieza con diodos LED, que se iluminaban dependiendo de la casilla sobre la que estuvieras (en el caso de las dobles haría falta pisar las dos a la vez).

Carpe planeaba decorar el túnel con una gran alfombra de fieltro para cambiar su imagen durante el evento. Así que yo preparé la mía en un material parecido para que se fusionara con la general, de manera que los transeúntes no tomaran conciencia real sobre la pieza a no ser que se cruzara en su camino. En ese instante alguna de las luces se encenderían, si continua caminando las luces seguirían a su paso, y en algún momento el transeúnte notaría que algo extraño estaba pasando. En este momento la parte principal, de iluminar el túnel ya se estaría llevando a cabo, y es el momento en el que el uso que se le dé la pieza está en manos del espectador, que puede elegir entre jugar o pasar de largo.

En esta ocasión los sensores no iban conectados a ningún tipo de placa que los gestionara, en esta pieza no hay nada de programación. Los sensores se usan directamente para permitir o evitar que el circuito se cierre y la energía de la pila llegue a los diodos LED, así que la tarea principal era diseñar un buen circuito que lo hiciera posible. No es un circuito complicado, sería como sigue: Tampoco tenía sentido, en este caso, que los sensores estuvieran separados, preparé la pieza de foam para que separara las piezas de papel de plata de todos los sensores. Para no tener problemas de contacto entre sensores contorneé todas las piezas de papel de plata con cinta aislante, y las pegué en su correspondiente posición. A continuación sobre todo esto monté ambas caras del circuito, las conecté con las luces, puse todas las pilas y probé que todo funcionara correctamente.



*Sambori, 2014. Fotos de la intervención en el túnel del Viveros con motivo del evento Viaje al interior de la noche de Carpe Via. Se adivinan las luces en el fondo pero la gente cubre la pieza.*

NOTA.- Vídeo de la pieza.

Tras esto me dispuse a tapar todos los circuitos cubriendo el sistema, por la parte inferior con un material plástico que me sirviera también de aislante por si había charcos o cualquier otra cosa en el suelo. Y por la superior de la tela gris, sobre la que dibujé el sambori con mucho cuidado de que coincidiera exactamente con la posición de los sensores.

De esta manera todo el conjunto estaba formado por una sola pieza, y si montaje es tan sencillo como extender la alfombra en el lugar deseado. Una vez en casa, y para evitar posibles problemas, sólo habría que retirar las pilas y se podría guardar.

Llegados al evento la pieza se montaba tan rápido que fue lo primero en estar listo, esto en realidad fue un problema ya que el resto de cosas (la banda que tocaba, la mesa con la comida, etc.) se dispusieron a su alrededor y nadie pudo encontrarse la pieza por sorpresa. Además el evento fue muy concurrido y había tanta gente en el túnel que la mayoría del tiempo estaban todas las luces encendida, porque todas sus casillas tenían a alguien encima. No creo que la pieza en si fuera un fracaso, sencillamente aquella noche no funcionó. Considero que no tenía cabida, aunque sí cumplió su función de iluminar de forma más o menos interactiva, y pude ver a alguno que otro recorrerla a saltos. Quizás sea una pieza que funcione mejor sin evento que la adorne, simplemente abandonada en un túnel oscuro.





Quiz, 2014. Vista previa de la pieza desde la web de desarrollo [consulta: 2014-04-07] <<http://programalaplaza.medialab-prado.es/sketch/view/d74f74ea659d8be040c9a-65f69dd77a4#>> Tengase en cuenta al probarla que se maneja con el movimiento que capture la cámara de vídeo. En este caso el vídeo de muestra que ofrece la página.

### 3.3.6 Quiz, 2014

El Medialab del Prado lanzó la propuesta de poner a disposición de todo el que quisiera un apantalla gigante que vestiría uno de los muros de su fachada. En ella se podrías proyectar aplicaciones programadas en Processing desde un IDE online que también ponían a disposición pública. Al entrar a ver el IDE comprobé que además contaban con un par de inputs. Una cámara de vídeo al que ya le habían asociado una librería que gestionaba el tracking de manera que podías seguir sin ninguna dificultad el paso de la gente que se encuentra frente al edificio. Y un estímulo desde el móvil. Una especie de botón al que se puede acceder desde tu propio teléfono móvil entrando a una página web cuando estás allí. (Reconozco que esta segunda opción no me interesó tanto, y no hice uso de ella) La oferta me pareció muy oportuna, así que decidí hacer algo para participar.

Barajé unas cuantas ideas, paseando por la galería se podían ver muchas propuestas que ya habían sido colgadas, casi todas potenciando el aspecto visual de la herramienta. Pensé que era una buena ocasión para hacer otro juego. Sopesé seriamente la idea de hacer algo más visual, un montón de bolas de colores a las que el paseante pudiera golpear virtualmente, una fiesta de disfraces en la que cada persona que pasara fuera tapada por un personaje distinto, una rayuela virtual similar a la hecha para el viaje en el interior de la noche... Me tentó especialmente hacer un juego del tipo “escapa de la habitación” transformando la fachada en un cuarto con pistas del que escapar, no descarto esta idea para el futuro. Pero en esta propuesta no nos decían nada sobre el tiempo que estarían pasando cada aplicación, así que debía de pensar en algo que pudiera funcionar bien tanto un periodo corto de tiempo como uno largo. Y un puzzle largo no era una buena solución.

Se me ocurrió partiendo de la idea del puzzle y la premisa del tiempo que podría hacer un juego de preguntas, y me daría lo mismo si sólo pudieran contestar a una, o a muchas. Preparé un lote de 48 preguntas que de alguna manera estuvieran relacionadas con las artes visuales, me decanté por preguntas de verdadero/falso para tener un buen reparto del espacio para la interfaz. Un contador fijo indicaría cuántos aciertos y cuántos fallos estaba teniendo la plaza aquella noche.

Para responder, los espectadores deberán colocar su reflejo en la pantalla sobre la respuesta deseada. Se trata de una pequeña esfera en un color aleatorio que seguirá a cualquiera que esté al alcance de la cámara. Este punto se genera cada vez que alguien entra, y se elimina cuando sale. Si la misma persona sale y vuelve a entrar en esta área tendrá, seguramente distinto color. Si varias personas se encuentran contestando a la vez, y no están de acuerdo, saldrá un aviso en la pantalla informando de que sólo puede contestar uno, y permanecerá ahí hasta que alguno de los huecos de respuesta se encuentre vacío.





Medialab Prado, Pantalla gigante para la que está diseñado el proyecto Quiz.

Atari light, Pierre Huyghe.

Los únicos materiales que necesité para este proyecto fueron mi ordenador e internet, lo genial de la propuesta es que ellos ponían todo lo demás. Desde el IDE en el que programarlo, en este caso basado en Processing, pasando por la pantalla gigante, la web-cam, y hasta el público.

El esquema de funcionamiento del código era el siguiente:

En este caso la pieza es un juego, lo único que yo he hecho es el código, no he creado ninguna interfaz, ni he diseñado el modo en el que se va a ver, nada. Esto me suscitó algunas dudas, porque desde mi punto de vista se trata al 100% de una pieza de carácter artístico, pero no estaba muy segura de saber cómo justificarlo.

Me ayudó toparme en otra conferencia TED con estas palabras de Paola Antonelli en las que explica porque compró el juego Pac-Mac para el MoMA<sup>11</sup>:

“Luego apareció otro artículo muy pretencioso que salió en The New Republic, tan pretencioso, de Liel Leibovitz que decía: “El MoMA ha confundido videojuegos con arte”. Otra vez. “El Museo pone el Pac-Man junto a Picasso”. Otra vez. “No entienden la idea”. Discúlpame. Tú no entiendes la idea. Y aquí, miren, la pregunta anterior se hace sin rodeos: “¿Los video juego son arte? No. Los video juegos no son arte porque son más bien otra cosa: código”. Así que Picasso no es arte porque es pintura de aceite. ¿Verdad?” [sic]

En este caso en particular, me sirvió de gran ayuda hacer una búsqueda de referentes. Por ejemplo, trabajos de un carácter similar como *We feel fine* de Jonathan Harris nos muestra, mediante una bonita y diseñada visualización de datos, todos los textos que encuentra que se han escrito durante el día que comiencen por la frase “Yo siento” o “Yo estoy sintiendo”. En este caso el se encarga del aspecto visual, pero el contenido depende por completo de lo que la gente haya escrito durante ese día.

*33 Preguntas por minuto* de Rafael Lozano Hemmer consta de veintidós displays electrónicos mostrando preguntas a una velocidad que no deja margen de tiempo al espectador para reflexionar sobre su contenido. Aunque en este caso las preguntas son generadas de forma aleatoria el trasfondo es similar, es el de generar dudas y reflexiones. Por último encontré otro mástil de apoyo en *Atari light* que forma parte de *Le chateau de Turing* de Pierre Huyghe, toda la pieza es de mucho interés, pero en este caso rescato *Atari Light* por el parecido con mi propio trabajo, transformando un juego tradicional (creo que ya podemos considerar al *Pong* como un juego tradicional) en una pieza interactiva a gran escala.

11 ANTONELLI, P, Why I brought Pac-Man to MoMA [Multimedia] TEDSalon NY2013 • Grabado May 2013 [consulta: 2014-07-04] < [https://www.ted.com/talks/paola\\_antonelli\\_why\\_i\\_brought\\_pacman\\_to\\_moma](https://www.ted.com/talks/paola_antonelli_why_i_brought_pacman_to_moma)>

*Quiz* por mi parte, se ofrece en forma de juego colaborativo. Retando a un conjunto de gente, que de alguna manera tendrá que decidir cómo gestionar las respuestas y qué contestar. De alguna manera se presenta como una chistosa evaluación de conocimientos, aunque en realidad quiere ser un reto sobre participación colectiva.

## 4. CONCLUSIONES

Finalmente puedo decir que he encontrado un camino por el que me interesa seguir trabajando. Las obras que he conseguido idear alcanzan aquella intención inicial de acercarse al espectador. Por su mismo carácter lúdico amplían necesariamente el tiempo que el espectador emplea en su visualización, posibilitan que se haga una mayor reflexión sobre ellas y a la vez son entretenidas y divierten.

Nadie piensa al verlas que soy una talentosa ingeniera, o que tenga grandes destrezas plásticas, pero es algo que nunca he buscado. Son piezas sencillas, asequibles tanto en técnica como en presupuesto, que transmiten esa cualidad del “hágaselo usted mismo” que motiva al espectador a sentirse un potencial artista y recordar que es un ser creativo. Sin perder lo que sí considero importante, el contenido poético y la interactividad.

Esta interactividad que en mis piezas siempre busca una asociación con la de aquellos juegos que tenemos entre las memorias de la infancia, reactivando haciendo que sean más usables y sencillas, que la reacción del espectador ante ellas sea más espontánea y cercana.

Son piezas además que encajan en el espacio de forma suave, despacio, para que sea el espectador quien las descubra por sí mismo. Intentando ampliar esa implicación entre ambos.

## 5. BIBLIOGRAFÍA

AARTSETH,E. *Investigación sobre juegos: aproximaciones metodológicas al análisis de juegos*. En: Artnodes: Revista de arte, ciencia y tecnología, UOC, 2007, num. 7, ISSN: 1695-5951. [Consulta: 2014-07-04]. Disponible en: <<http://www.uoc.edu/artnodes/7/dt/esp/aarseth.pdf>>

ALSINA, P. *Nodo << Jugabilidad: arte, videojuegos y cultura>>* En: Artnodes: Revista de arte, ciencia y tecnología, UOC, 2007, num. 7, ISSN: 1695-5951. [Consulta: 2014-07-04]. Disponible en: <<http://www.uoc.edu/artnodes/7/dt/esp/alsina.pdf>>

ARS GAMES. [consulta: 2014-07-04]. Disponible en: <<http://arsgames.net/blog/>>

ÁVILA VALDES, N. *Interactividad y arte interactivo. La realidad Virtual Inmersiva*. En: *Arte, individuo y sociedad*, Madrid: Facultad de Bellas Artes (UCM), 2013, , num. 7, ISSN: 1131-5598 2003. . [Consulta: 2014-07-04]. Disponible en: <<http://revistas.ucm.es/index.php/ARIS/article/viewFile/ARIS0303110163A/5839>>

BANZI, M. *Getting started with Arduino*. O'Reilly, 2011 [Libro online]. ISBN: 978-1-449-309879 [Consulta: 2014-07-04]. Disponible en: <<http://it-ebooks.info/read/1338/>>

BOOOOOOOM. [consulta: 2014-07-04]. Disponible en: <<http://www.booooooom.com/>>

CUARTO DERECHA. [consulta: 2014-07-04]. Disponible en: <<http://cuartoderecha.com/>>

DESIGN-MILK. [consulta: 2014-07-04]. Disponible en: <<http://design-milk.com/>>

ERIC ZIMMERMAN, *Essays and Articles* [consulta: 2014-07-04]. Disponible en: <<http://ericzimmerman.com/portfolio/essays-and-articles/>>

ERROR500. [consulta: 2014-07-04]. Disponible en: <<http://www.error500.net/>>

FUBIZ [consulta: 2014-07-04]. Disponible en: <<http://www.fubiz.net/category/technology/>>

GADAMER, H-G. *La actualidad de lo bello*: Paidós Ibérica, 1991

GONZÁLEZ TARDÓN,C; CAÑAS,D. *¿Puede un computador escribir un poema de amor? Tecnorromanticismo y poesía electrónica*. Madrid: Devenir Ensayo, 2010

GUTIÉRREZ PARRAGA M.T. *La significación del juego en el arte moderno y sus implicaciones en la educación artística* [tesis doctoral]. Madrid: Universidad Complutense de Madrid, 2004.

LABORAL. CENTRO DE ARTE Y CREACIÓN INDUSTRIAL, *Gameworld: Videojuegos en la frontera entre arte, tecnología y cultura* [catálogo], Gijón: Laboral y Puerto de Gijón, 2007. <<http://www.laboralcentrodearte.org/es/files/2007/exposiciones/gameworld-doc/catalogo-gameworld>>

LARRAÑAGA, J. *Instalaciones*. Donostia - San Sebastián: EFCA S.A., 2001

MARTÍNEZ, B. *La postproducción cultural. Una estética de relaciones* [tesis de fin de máster]. Valencia: Universitat Politècnica de València, 2007.

MEIAC, *El discreto encanto de la tecnología* [catálogo], Badajoz, Ministerio de Cultura, Secretaría General Técnica, Subdirección General de Publicaciones, Información y Documentación; Junta de Extremadura, Consejería de Cultura y Turismo; MEIAC, Museo Extremeño e Iberoamericano de Arte Contemporáneo, 2008.

METROPOLIS. *Nerd Art* [documental]. España: RTVE La 2, 2009

MUSEO NACIONAL CENTRO DE ARTE REINA SOFÍA, *Playgrounds: reinventar la plaza* [catálogo] Madrid: Museo Nacional Centro de Arte Reina Sofía y Siruela, 2014.

PAISEA 002 *EL JUEGO - PLAYSCAPES*. Valencia: Paisea Revista S. L., 2012. Num 22, ISSN: 1887-2557

SHELL, J. *The art of game design, a book of lenses*. Burlington: Morgan Kaufmann Publishers, 2009.

SWEET-STATION. [consulta: 2014-07-04]. Disponible en: <<http://sweet-station.com/blog/>>

SWISS MISS. [consulta: 2014-07-04]. Disponible en: <<http://www.swiss-miss.com/>>

TED, Ideas worth spreading\*. [consulta: 2014-07-04]. Disponible en: <<http://www.ted.com/>>

THE FOX IS BLACK. [consulta: 2014-07-04]. Disponible en: <<http://www.thefoxisblack.com/>>

## 6. ANEXOS

### 6.1 CÓDIGO DE *HORMIGAS*

#### *Main.cpp*

```
#include <QtGui/QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    //w.show();
    w.showFullScreen();
    return a.exec();
}
```

#### *CollisionEngine.h*

```
#ifndef COLLISIONENGINE_H
#define COLLISIONENGINE_H

////
// Funciones y metodos de colisión de objetos.
//
// Búsqueda del vector de proyección:
// http://www.metanetsoftware.com/technique/tutorialA.html
// http://chortle.ccsu.edu/VectorLessons/vectorIndex.html
//
// SweepTest:
// http://www.gamasutra.com/view/feature/3383/simple\_intersection\_
tests\_for\_games.php
////

#include <QPair>
#include <QPointF>
#include <QGraphicsPixmapItem>
#include <QRectF>

class CCoin;
class CFloor;
class CFloorConcav;
```

```

class CPolTri;
class CPolConcav;
class CCollisionEngine
{
public:
    CCollisionEngine();
    ~CCollisionEngine();

    bool    calculateProjectionVector( QGraphicsPixmapItem* pObject,
    QGraphicsPixmapItem* pObjectCollided,
        QPair<qreal, qreal>& pairProyVector, qreal& fAngle);

    bool    checkCollTwoRect(QRectF rec1, QRectF rec2);
    bool    ptIsInsideRect(QPointF pt, QRectF rect);

private:
    bool    calcProjVCoinWithFloor(CCoin* pCoin, CFloor* pFloor,
    QPair<qreal, qreal>& pairProyVector, qreal& fAngle);

    bool    calcProjVCircleWithRect(QRectF* pRecCircle, QRectF* pRect,
    QPair<qreal, qreal>& pairProyVector, qreal& fAngle);
    bool    calcProjVCircleWithTri(QRectF* pRecCircle, CPolTri* pTri,
    QPair<qreal, qreal>& pairProyVector, qreal& fAngle);
    bool    calcProjVCircleWithConcav(QRectF* pRecCircle, CPolConcav* pConcav,
    QPair<qreal, qreal>& pairProyVector, qreal& fAngle);

    qreal    calcProjVecVertAxis(qreal fPlayerTop, qreal fPlayerBottom,
    qreal fFloorTop, qreal fFloorBottom);
    qreal    calcProjVecHorAxis(qreal fPlayerRight, qreal fPlayerLeft,
    qreal fFloorRight, qreal fFloorLeft);
    qreal    calcProjVecHypoAxisTRI(QRectF* pRecCircle, CPolTri* pTri);
    //qreal    calcProjVecVoronAxisRec(QRectF* pRecCircle, QRectF*
    pRect, QPointF ptCloserVertex, QPointF ptFurtherVertex);
    //qreal    calcProjVecVoronAxisTri(QRectF* pRecCircle, CPolTri* pTri,
    QPointF ptCloserVertex, QPointF ptOtherVertex);

    //Voronoi region
    bool    ptIsInsideVoronoiRegion(QRectF* rect, QPointF* ptAux,
    QPointF& ptClosestVertex, QPointF& ptOppositeVertex);
    bool    ptIsInsideVoronoiRegion(CPolTri* tri, QPointF* ptAux,
    QPointF& ptClosestVertex, QPointF& ptOtherVertex);
    bool    ptIsInsideVoronoiRegion(CPolConcav* pConcav, QPointF*
    ptAux,
        QPointF& ptClosestVertex, qreal& fDistance);

```

```

        //Concavo
        bool   recIsInQuadrant(QRectF* pRecCircle, CPolConcav* pPolConcav, QPointF& ptClosest);

        QPointFproject(QPointF vectToProj, QPointF vectUnitAxis);
        qreal   calcLenghtProjection(QPointF vectToProj1, QPointF vectToProj2, QPointF vectUnitAxis);
        qreal   calcLenghtProjection(QPointF vectToProj1, QPointF vectToProj2, QPointF vectUnitAxis,
                                     QPointF& vecProjected1, QPointF& vecProjected2);
    };

#endif

```

### ***CollisionEngine.cpp***

```

#include "CollisionEngine.h"
#include "../Characters/Coin.h"
#include "../Floor/FloorRec.h"
#include "../Floor/FloorTri.h"
#include "../Floor/FloorConcav.h"
#include "../Floor/FloorMagnet.h"
#include "math.h"
#include <QVector>
#include <QDebug>

CCollisionEngine::CCollisionEngine()
{}

CCollisionEngine::~CCollisionEngine()
{}

bool CCollisionEngine::checkCollTwoRect(QRectF rec1, QRectF rec2)
{
    if(((rec1.top() > rec2.top() && rec1.top() < rec2.bottom()) ||
        (rec1.bottom() > rec2.top() && rec1.bottom() < rec2.bottom()))&&
        ((rec1.right() > rec2.left() && rec1.right() < rec2.right()) ||
         (rec1.left() > rec2.left() && rec1.left() < rec2.right()))))
        return true;
    else
        return false;
}

```



```

    }

    bool CCollisionEngine::ptIsInsideRect(QPointF pt, QRectF rect)
    {
        if(pt.x() > rect.left() && pt.x() < rect.right() &&
            pt.y() < rect.bottom() && pt.y() > rect.top())
            return true;
        else
            return false;
    }

    bool CCollisionEngine::calculateProjectionVector( QGraphicsPixmapItem*
    pObject,
    QGraphicsPixmapItem* pObjectCollided, QPair<qreal, qreal>& pairPro-
    yVector, qreal& fAngle)
    {
        if(static_cast<CCoin*>(pObject) != 0)
        {
            if(static_cast<CFloor*>(pObjectCollided) != 0)
                if (calcProjVCoinWithFloor(static_
cast<CCoin*>(pObject),
                static_cast<CFloor*>(pObjectCollided), pair-
ProyVector, fAngle))

                    return true;

        }

        return false;
    }

    QPointF CCollisionEngine::project( QPointF vectToProj, QPointF vectUni-
    tAxis)
    {
        qreal fpt1X = ((vectToProj.x() * vectUnitAxis.x() + vectToProj.y() *
vectUnitAxis.y()) * vectUnitAxis.x());
        qreal fpt1Y = ((vectToProj.x() * vectUnitAxis.x() + vectToProj.y() *
vectUnitAxis.y()) * vectUnitAxis.y());
        return QPointF(fpt1X, fpt1Y);
    }

    qreal CCollisionEngine::calcLenghtProjection( QPointF vectToProj1,
    QPointF vectToProj2, QPointF vectUnitAxis)
    {

```

```

        QPointF vecProjected1 = project(vectToProj1, vectUnitAxis);
        QPointF vecProjected2 = project(vectToProj2, vectUnitAxis);
        return sqrt((vecProjected1.x() - vecProjected2.x()) * (vecProjected1.x()
- vecProjected2.x())
                + (vecProjected1.y() - vecProjected2.y()) * (vecProjected1.y()
- vecProjected2.y()));
    }

```

```

qreal CCollisionEngine::calcLenghtProjection( QPointF vectToProj1,
QPointF vectToProj2, QPointF vectUnitAxis,

```

```

        QPointF& vecProjected1, QPointF& vecProjected2)
    {
        vecProjected1 = project(vectToProj1, vectUnitAxis);
        vecProjected2 = project(vectToProj2, vectUnitAxis);
        return sqrt((vecProjected1.x() - vecProjected2.x()) * (vecProjected1.x()
- vecProjected2.x())
                + (vecProjected1.y() - vecProjected2.y()) * (vecProjected1.y()
- vecProjected2.y()));
    }

```

```

bool CCollisionEngine::calcProjVCoinWithFloor(CCoin* pCoin, CFloor*
pFloor, QPair<qreal, qreal>& pairProyVector, qreal& fAngle)
    { //Choque de coin con suelo. Coin es redondo.

```

```

        if(pFloor->getType() == CFloor::FLOOR_RECTANGLE
            || pFloor->getType() == CFloor::FLOOR_MOV_RECT
            || pFloor->getType() == CFloor::FLOOR_MAGNETIC)
        {
            CFloorRec* pFloorRecAux = static_cast<CFloorRec*>(pFloor);
            QRectF recFloorMapped;
            if(pFloor->getType() == CFloor::FLOOR_MAGNETIC)
            { //Caso especial. El BoundingRect indica la sección a pintar.
                //Hay un boundingRect especial para colisiones
                CFloorMagnet* pFloorAuxMagnet = static_
cast<CFloorMagnet*>(pFloor);
                recFloorMapped.setTopLeft(pFloorAuxMagnet->mapToItem(pCoin,
pFloorAuxMagnet->getBoundingRectForCollisions().topLeft());
                recFloorMapped.setWidth(pFloorAuxMagnet-
>getBoundingRectForCollisions().width());
                recFloorMapped.setHeight(pFloorAuxMagnet-
>getBoundingRectForCollisions().height());
                pFloorAuxMagnet = 0;
            }
        }
    }

```

```

        else
        {
            recFloorMapped.setTopLeft(pFloorRecAux->mapToItem(pCoin,
pFloorRecAux->getCollisionBoundingRect().topLeft()));
            recFloorMapped.setWidth(pFloorRecAux-
>getCollisionBoundingRect().width());
            recFloorMapped.setHeight(pFloorRecAux-
>getCollisionBoundingRect().height());
        }

        pFloorRecAux = 0;
        QRectF recCoin = pCoin->boundingRect();

        if(calcProjVCircleWithRect(&recCoin, &recFloorMapped,
pairProyVector, fAngle))
            return true;
        else
            return false;
    }
    else if(pFloor->getType() == CFloor::FLOOR_TRIANGLE)
    {
        CFloorTri* pFloorTriAux = static_cast<CFloorTri*>(pFloor);

        QPointF pt1 = pFloorTriAux->mapToItem(pCoin, pFloor-
TriAux->getPolTri()->at(0));
        QPointF pt2 = pFloorTriAux->mapToItem(pCoin, pFloorTriAux-
>getPolTri()->at(1));
        QPointF pt3 = pFloorTriAux->mapToItem(pCoin, pFloorTriAux-
>getPolTri()->at(2));

        pFloorTriAux = 0;
        CPolTri polTriMapped(pt1, pt2, pt3);
        QRectF recCoin = pCoin->boundingRect();
        if(calcProjVCircleWithTri(&recCoin, &polTriMapped, pairPro-
yVector, fAngle))
            return true;
        else
            return false;
    }
    else if(pFloor->getType() == CFloor::FLOOR_CONCAV)
    {
        CFloorConcav* pFloorConcavAux = static_
cast<CFloorConcav*>(pFloor);
        QPointF pt1 = pFloorConcavAux->mapToItem(pCoin, pFloor-

```

```

ConcavAux->getPolConcav()->getPt1());
    QPointF pt2 = pFloorConcavAux->mapToItem(pCoin, pFloor-
ConcavAux->getPolConcav()->getPt2());
    QPointF pt3 = pFloorConcavAux->mapToItem(pCoin, pFloor-
ConcavAux->getPolConcav()->getPt3());
    pFloorConcavAux = 0;
    CPolConcav concavAux(pt1, pt2, pt3);
    QRectF recCoin = pCoin->boundingRect();
    if(calcProjVCircleWithConcav(&recCoin, &concavAux, pair-
ProyVector, fAngle))
        return true;
    else
        return false;
}

return false;
}

```

```

bool CCollisionEngine::calcProjVCircleWithRect(QRectF* pRecCircle,
QRectF* pRect, QPair<qreal, qreal>& pairProyVector, qreal& fAngle)
    //El rectángulo SIEMPRE estará mapeado en el eje de coordenadas de la
moneda.

```

```

    //Comprobamos los dos ejes (horizontal, vertical). Si uno falla no hay
colisión

```

```

    qreal fVertAxis = calcProjVecVertAxis(pRecCircle->top(), pRecCircle-
>bottom(),
        pRect->top(), pRect->bottom());

```

```

    if(fVertAxis <= 0) //No hay colisión
        return false;

```

```

    qreal fHorAxis = calcProjVecHorAxis(pRecCircle->right(), pRecCircle-
>left(),
        pRect->right(), pRect->left());

```

```

    if(fHorAxis <= 0) //No hay colisión
        return false;

```

```

    QPointF ptCloserVertex;
    QPointF ptFurtherVertex;
    qreal fVertexAxis = 9999;
    if(ptsInsideVoronoiRegion(pRect, &(pRecCircle->center()), ptCloser-
Vertex, ptFurtherVertex))
    {

```

```

        //Caso directo: está a más distancia que el radio del círculo,

```

no hay colisión

```

        QLineF lnAux(pRecCircle->center(), ptCloserVertex);
        if(lnAux.length() > pRecCircle->width()/2)
            return false;
        else
            fVertexAxis = pRecCircle->width()/2 - lnAux.length();

        /*fVertexAxis = calcProjVecVoronAxisRec(pRecCircle, pRect,
        ptCloserVertex, ptFurtherVertex);
        if(fVertexAxis == 666)//Está en región voronoi y no superpone
            return false;*/
    }

    //Se proyecta el menor de todos.
    if((fVertAxis < fHorAxis) && (fVertAxis < fVertexAxis))
    { //Proyección vertical
        if(pRecCircle->bottom() > pRect->bottom())
            pairProyVector.second = fVertAxis;
        else
            pairProyVector.second = -fVertAxis;

        fAngle = 0;
    }
    else if((fHorAxis < fVertAxis) && (fHorAxis < fVertexAxis))
    { //Proyección horizontal
        if(pRecCircle->right() > pRect->right())
            pairProyVector.first = fHorAxis;
        else
            pairProyVector.first = -fHorAxis;
    }
    else //proyección Voronoi
    {
        QLineF lineAux(ptCloserVertex, pRecCircle->center());
        qreal fAngleAux = lineAux.angle();
        lineAux.setP1(QPointF(0, 0));
        lineAux.setAngle(fAngleAux);
        lineAux.setLength(fVertexAxis);
        pairProyVector.first = lineAux.p2().x();
        pairProyVector.second = lineAux.p2().y();
    }

    return true;

```

```

}

bool CCollisionEngine::calcProjVCircleWithTri( QRectF* pRecCircle, CPol-
Tri* pTri, QPair<qreal, qreal>& pairProyVector, qreal& fAngle)
{//El triángulo SIEMPRE estará mapeado en el eje de coordenadas de la
moneda.
    //Comprobamos los dos ejes (horizontal, vertical). Si uno falla no hay
colisión
    qreal fVertAxis = calcProjVecVertAxis(pRecCircle->top(), pRecCircle-
>bottom(),
        pTri->getPTTop().y(), pTri->getPTBottom().y());
    if(fVertAxis <= 0) //No hay colisión
        return false;
    qreal fHorAxis = calcProjVecHorAxis(pRecCircle->right(), pRecCircle-
>left(),
        pTri->getPTRight().x(), pTri->getPTLeft().x());
    if(fHorAxis <= 0) //No hay colisión
        return false;
    qreal fDiagAxis = calcProjVecHypoAxisTRI(pRecCircle, pTri);
    if(fDiagAxis <= 0) //No hay colisión
        return false;
    QPointF ptClosestVertex;
    QPointF ptOtherVertex;
    qreal fVoronAxis = 9999;

    if(ptsInsideVoronoiRegion(pTri, &pRecCircle->center(), ptCloses-
tVertex, ptOtherVertex))
    {
        //Caso directo: está a más distancia que el radio del círculo,
no hay colisión
        QLineF lnAux(pRecCircle->center(), ptClosestVertex);
        if(lnAux.length() > pRecCircle->width()/2)
            return false;
        else
            fVertAxis = pRecCircle->width()/2 - lnAux.length();
    }
    //Se proyecta el menor de todos.
    if((fDiagAxis < fVertAxis) && (fDiagAxis < fHorAxis) && (fDiagAxis <
fVoronAxis))
        {//Proyección perpendicular a la hipotenusa
            QLineF line(0, 0, 1, 1);
            line.setAngle(pTri->getPerpHypoNormalAngle());
            line.setLength(fDiagAxis);

```

```

        pairProyVector.first += line.p2().x();
        pairProyVector.second += line.p2().y();
        fAngle = pTri->getPerpHypoNormalAngle() + 90;
    }
    else if((fVoronAxis < fVertAxis) && (fVoronAxis < fHorAxis) && (fVoronAxis < fDiagAxis))
    { //Proyección en área de Voronoi
        QLineF lineAux(ptClosestVertex, pRecCircle->center());
        qreal fAuxAngle = lineAux.angle();
        lineAux.setP1(QPointF(0, 0));
        lineAux.setAngle(fAuxAngle);
        lineAux.setLength(fVertAxis);
        pairProyVector.first = lineAux.p2().x();
        pairProyVector.second = lineAux.p2().y();
    }
    else if((fVertAxis < fHorAxis) && (fVertAxis < fDiagAxis) && (fVertAxis < fVoronAxis))
    { //Proyección vertical
        if(pRecCircle->bottom() > pTri->getPTBottom().y())
            pairProyVector.second = fVertAxis;
        else
            pairProyVector.second = -fVertAxis;
        fAngle = 0;
    }
    else
    { //Proyección horizontal
        if(pRecCircle->right() > pTri->getPTRight().x())
            pairProyVector.first = fHorAxis;
        else
            pairProyVector.first = -fHorAxis;
    }

    return true;
}

```

```

bool CCollisionEngine::calcProjVCircleWithConcav( QRectF* pRecCircle,
CPolConcav* pConcav, QPair<qreal, qreal>& pairProyVector, qreal& fAngle)
{ //El polígono SIEMPRE estará mapeado en el eje de coordenadas de la moneda.

```

```

    //Primero miramos la región voronoi, si no la cóncava, y luego la horizontal y la vertical a la vez.

```

```

    QPointF ptClosestVertex;

```

```

//Voronoi
qreal fDistance;
if(ptIsInsideVoronoiRegion(pConcav, &pRecCircle->center(), ptClosestVertex, fDistance))
{
    if(fDistance > pRecCircle->width() / 2)
        return false;
    else
    {
        qreal fLength = pRecCircle->width() / 2 - fDistance;
        QLineF lnAux(ptClosestVertex, pRecCircle->center());
        qreal fAngle = lnAux.angle();
        lnAux.setP1(QPointF(0,0));
        lnAux.setAngle(fAngle);
        lnAux.setLength(fLength);
        pairProyVector.first = lnAux.p2().x();
        pairProyVector.second = lnAux.p2().y();
        return true;
    }
}

//Cóncavo
if(reclsInQuadrant(pRecCircle, pConcav, ptClosestVertex))
{
    //Si la distancia centro círculo de figura cóncava - vértice más
    cercano
    //es mayor que el radio, hay colisión.
    QLineF lnAux(ptClosestVertex, pConcav->getPtCenter());
    qreal fDistance = lnAux.length();
    if(fDistance < pConcav->getRadius())
        return false;
    else
    {
        //Hay colisión, calcular el vector proyección.
        qreal fProjLength = fDistance - pConcav->getRadius();
        QLineF ln(ptClosestVertex, pConcav->getPtCenter());
        QLineF lnAux(QPointF(0,0), QPointF(1,1)); //por
    ejemplo
        lnAux.setAngle(ln.angle());
        lnAux.setLength(fProjLength);
        pairProyVector.first = lnAux.p2().x();
        pairProyVector.second = lnAux.p2().y();
        return true;
    }
}

```



```

    }
    //Horizontal y vertical
    else
    {
        qreal fVertAxis = calcProjVecVertAxis(pRecCircle->top(), pRecCircle->bottom(),
                                                pConcav->getPTTop().y(),          pConcav->getPTBottom().y());
        if(fVertAxis <= 0) //No hay colisión
            return false;
        qreal fHorAxis = calcProjVecHorAxis(pRecCircle->right(), pRecCircle->left(),
                                                pConcav->getPTRight().x(),          pConcav->getPTLeft().x());
        if(fHorAxis <= 0) //No hay colisión
            return false;
        if(fVertAxis < fHorAxis)
        { //Proyección vertical
            if(pRecCircle->bottom() > pConcav->getPTBottom().y())
                pairProyVector.second = fVertAxis;
            else
                pairProyVector.second = -fVertAxis;

            fAngle = 0;
        }
        else
        { //Proyección horizontal
            if(pRecCircle->right() > pConcav->getPTRight().x())
                pairProyVector.first = fHorAxis;
            else
                pairProyVector.first = -fHorAxis;
        }

        return true;
    }

    return false;
}

qreal CCollisionEngine::calcProjVecVertAxis( qreal fPlayerTop, qreal fPlayerBottom, qreal fFloorTop, qreal fFloorBottom )
{
    if((fPlayerBottom <= fFloorTop) || (fPlayerTop >= fFloorBottom))

```

```

        return -999;//No hay colisión

        //Calcular el movimiento necesario hacia arriba y abajo, nos quedamos con el menor.
        qreal fMovUp = fFloorTop - fPlayerBottom;
        qreal fMovDown = fFloorBottom - fPlayerTop;

        if(fMovUp < 0)
            fMovUp = -fMovUp;
        if(fMovDown < 0)
            fMovDown = -fMovDown;

        if(fMovUp < fMovDown)
            return fMovUp;
        else
            return fMovDown;
    }

    qreal CCollisionEngine::calcProjVecHorAxis( qreal fPlayerRight, qreal fPlayerLeft, qreal fFloorRight, qreal fFloorLeft )
    {
        if((fPlayerLeft >= fFloorRight) || (fPlayerRight <= fFloorLeft))
            return -999;//No hay colisión

        //Calcular el movimiento necesario hacia la izq. y der., nos quedamos con el menor.
        qreal fMovRight = fFloorRight - fPlayerLeft;
        qreal fMovLeft = fFloorLeft - fPlayerRight;

        if(fMovRight < 0)
            fMovRight = -fMovRight;
        if(fMovLeft < 0)
            fMovLeft = -fMovLeft;

        if(fMovRight < fMovLeft)
            return fMovRight;
        else
            return fMovLeft;
    }

    qreal CCollisionEngine::calcProjVecHypoAxisTRI( QRectF* pRectCircle, CPolTri* pTri )

```

```

{
    //El eje de proyección es el vector unitario normal a la hipotenusa.
    QLineF line(QPointF(0,0), QPointF(1,1)); //por ejemplo
    line.setAngle(pTri->getPerpHypoNormalAngle());
    QPointF pt2UnitVec = (line.unitVector()).p2();

    //Proyectar el triángulo en el eje.
    //La hipotenusa siempre proyecta cero, mapeamos cualquiera de las
    otras dos.
    QPointF ptProjTriangle1 = pTri->at(0);
    QPointF ptProjTriangle2;
    if(pTri->getLine1() != pTri->getHypotenuse())
        ptProjTriangle2 = pTri->at(1);
    else
        ptProjTriangle2 = pTri->at(2);

    QPointF ptProjMappedTriangle1;
    QPointF ptProjMappedTriangle2;
    calcLenghtProjection(ptProjTriangle1, ptProjTriangle2, pt2UnitVec,
    ptProjMappedTriangle1, ptProjMappedTriangle2);

    //Proyectar cuadrado en el eje.
    //Proyectar los vértices opuestos dos a dos y nos quedamos con la
    proyección mayor.
    QPointF ptProjMappedRec1;
    QPointF ptProjMappedRec2;
    QPointF ptProjMappedRecAUX11;
    QPointF ptProjMappedRecAUX21;
    QPointF ptProjMappedRecAUX12;
    QPointF ptProjMappedRecAUX22;
    qreal proYRec13 = calcLenghtProjection(pRectCircle->topLeft(), pRect-
    Circle->bottomRight(), pt2UnitVec,
        ptProjMappedRecAUX11, ptProjMappedRecAUX21);
    qreal proYRec24 = calcLenghtProjection(pRectCircle->topRight(),
    pRectCircle->bottomLeft(), pt2UnitVec,
        ptProjMappedRecAUX12, ptProjMappedRecAUX22);
    if(proYRec13 > proYRec24)
    {
        ptProjMappedRec1 = ptProjMappedRecAUX11;
        ptProjMappedRec2 = ptProjMappedRecAUX21;
    }
    else
    {
        ptProjMappedRec1 = ptProjMappedRecAUX12;
    }
}

```

```

        ptProjMappedRec2 = ptProjMappedRecAUX22;
    }

    //Calcular la proyección de ambos en el eje. Hacemos un rectángulo
    //con los puntos que cortan de cada uno y su diagonal será la longi-
    tud a mover.
    //Problema: si el cuadrado está completamente dentro del triángulo
    dará el máximo, que
    //es la diagonal del cuadrado en sí. Probar.
    QRectF recProjREC(ptProjMappedRec1, ptProjMappedRec2);
    QRectF recProjTRI(ptProjMappedTriangle1, ptProjMappedTrian-
    gle2);
    QRectF recIntersec = recProjREC.intersected(recProjTRI);
    if(recIntersec.isNull())
        return -999; //no hay colisión

    QLineF lnFinal(recIntersec.topLeft(), recIntersec.bottomRight());
    return lnFinal.length();
}

bool CCollisionEngine::ptIsInsideVoronoiRegion( QRectF* rect, QPointF*
ptAux,

    QPointF& ptClosestVertex, QPointF& ptOppositeVertex)
{
    if(ptAux->x() > rect->right() && ptAux->y() < rect->top())
    {
        ptClosestVertex = rect->topRight();
        ptOppositeVertex = rect->bottomLeft();
        return true;
    }
    else if(ptAux->x() > rect->right() && ptAux->y() > rect->bottom())
    {
        ptClosestVertex = rect->bottomRight();
        ptOppositeVertex = rect->topLeft();
        return true;
    }
    else if(ptAux->x() < rect->left() && ptAux->y() > rect->bottom())
    {
        ptClosestVertex = rect->bottomLeft();
        ptOppositeVertex = rect->topRight();
        return true;
    }
}

```

```

else if(ptAux->x() < rect->left() && ptAux->y() < rect->top())
{
    ptClosestVertex = rect->topLeft();
    ptOppositeVertex = rect->bottomRight();
    return true;
}
return false;
}

bool CCollisionEngine::ptIsInsideVoronoiRegion( CPolTri* tri, QPointF*
ptAux, QPointF& ptClosestVertex, QPointF& ptOtherVertex)
{
    //Solo nos interesa saber si está dentro de una reg. voronoi del vértice
más cercano.
    QLineF lnAux1(*ptAux, tri->at(0));
    QLineF lnAux2(*ptAux, tri->at(1));
    QLineF lnAux3(*ptAux, tri->at(2));
    qreal fLength1 = lnAux1.length();
    qreal fLength2 = lnAux2.length();
    qreal fLength3 = lnAux3.length();
    //Crear un polígono con puntos de las normales que delimitan las
regiones
    //de voronoi y vemos si contiene el punto que buscamos.
    if((fLength1 < fLength2) && (fLength1 < fLength3))
    {
        //Lo más seguro es que no esté, si está lejos del punto.
        //Caso directo:
        ptClosestVertex = tri->at(0);
        ptOtherVertex = tri->at(1); //Por ejemplo
        QLineF lnVoroPrev = tri->getLine3();
        QLineF lnVoroNext = tri->getLine1();
        lnVoroNext.setAngle(lnVoroNext.angle() + 90);
        lnVoroPrev.setLength(lnVoroPrev.length() * 2);
        lnVoroPrev.setP1(ptClosestVertex);
        lnVoroPrev.setAngle(lnVoroPrev.angle() + 90);
        QVector<QPointF> vPoints;
        vPoints << ptClosestVertex << lnVoroNext.p2() << lnVoro-
Prev.p2() << ptClosestVertex;
        QPolygonF polVoron(vPoints);

        if(polVoron.containsPoint(*ptAux, Qt::OddEvenFill))
            return true;
        else
            return false;
    }
}

```

```

}
else if((fLength2 < fLength1) && (fLength2 < fLength3))
{
    ptClosestVertex = tri->at(1);
    ptOtherVertex = tri->at(2); //Por ejemplo
    QLineF InVoroPrev = tri->getLine1();
    QLineF InVoroNext = tri->getLine2();
    InVoroNext.setAngle(InVoroNext.angle() + 90);
    InVoroPrev.setLength(InVoroPrev.length() * 2);
    InVoroPrev.setP1(ptClosestVertex);
    InVoroPrev.setAngle(InVoroPrev.angle() + 90);
    QVector<QPointF> vPoints;
    vPoints << ptClosestVertex << InVoroNext.p2() << InVoro-
Prev.p2() << ptClosestVertex;
    QPolygonF polVoron(vPoints);

    if(polVoron.containsPoint(*ptAux, Qt::OddEvenFill))
        return true;
    else
        return false;
}
else
{
    ptClosestVertex = tri->at(2);
    ptOtherVertex = tri->at(0); //Por ejemplo
    QLineF InVoroPrev = tri->getLine2();
    QLineF InVoroNext = tri->getLine3();
    InVoroNext.setAngle(InVoroNext.angle() + 90);
    InVoroPrev.setLength(InVoroPrev.length() * 2);
    InVoroPrev.setP1(ptClosestVertex);
    InVoroPrev.setAngle(InVoroPrev.angle() + 90);
    QVector<QPointF> vPoints;
    vPoints << ptClosestVertex << InVoroNext.p2() << InVoro-
Prev.p2() << ptClosestVertex;
    QPolygonF polVoron(vPoints);

    if(polVoron.containsPoint(*ptAux, Qt::OddEvenFill))
        return true;
    else
        return false;
}

return false;

```

```

}

bool CCollisionEngine::ptsInsideVoronoiRegion( CPolConcav* pConcav,
QPointF* ptAux, QPointF& ptClosestVertex, qreal& fDistance)
{
    if(pConcav->getQuadrant() == 1)
    {
        if((ptAux->x() < pConcav->getPt1().x()) || (ptAux->y() > pCon-
cav->getPt3().y()))
        {
            QLineF ln1(*ptAux, pConcav->getPt1());
            QLineF ln2(*ptAux, pConcav->getPt3());
            if(ln1.length() > ln2.length())
            {
                ptClosestVertex = pConcav->getPt3();
                fDistance = ln2.length();
            }
            else
            {
                ptClosestVertex = pConcav->getPt1();
                fDistance = ln1.length();
            }
        }
        else
            return false;
    }
    else if(pConcav->getQuadrant() == 2)
    {
        if((ptAux->x() > pConcav->getPt2().x()) || (ptAux->y() > pCon-
cav->getPt3().y()))
        {
            QLineF ln1(*ptAux, pConcav->getPt2());
            QLineF ln2(*ptAux, pConcav->getPt3());
            if(ln1.length() > ln2.length())
            {
                ptClosestVertex = pConcav->getPt3();
                fDistance = ln2.length();
            }
            else
            {
                ptClosestVertex = pConcav->getPt2();
                fDistance = ln1.length();
            }
        }
    }
}

```

```

        else
            return false;
    }
    else if(pConcav->getQuadrant() == 3)
    {
        if((ptAux->x() > pConcav->getPt2().x()) || (ptAux->y() < pCon-
cav->getPt1().y()))
        {
            QLineF ln1(*ptAux, pConcav->getPt1());
            QLineF ln2(*ptAux, pConcav->getPt2());
            if(ln1.length() > ln2.length())
            {
                ptClosestVertex = pConcav->getPt2();
                fDistance = ln2.length();
            }
            else
            {
                ptClosestVertex = pConcav->getPt1();
                fDistance = ln1.length();
            }
        }
        else
            return false;
    }
    else if(pConcav->getQuadrant() == 4)
    {
        if((ptAux->x() < pConcav->getPt3().x()) || (ptAux->y() < pCon-
cav->getPt1().y()))
        {
            QLineF ln1(*ptAux, pConcav->getPt1());
            QLineF ln2(*ptAux, pConcav->getPt3());
            if(ln1.length() > ln2.length())
            {
                ptClosestVertex = pConcav->getPt3();
                fDistance = ln2.length();
            }
            else
            {
                ptClosestVertex = pConcav->getPt1();
                fDistance = ln1.length();
            }
        }
        else
            return false;
    }

```



```

    }

    return true;
}

bool CCollisionEngine::recIsInQuadrant( QRectF* pRecCircle, CPolConcav* pPolConcav, QPointF& ptClosest)
{
    //Vértice más cercano al vértice del ángulo recto.
    QLineF ln1(pRecCircle->topLeft(), pPolConcav->getPtRectAngle());
    QLineF ln2(pRecCircle->topRight(), pPolConcav->getPtRectAngle());
    QLineF ln3(pRecCircle->bottomRight(), pPolConcav->getPtRectAngle());
    QLineF ln4(pRecCircle->bottomLeft(), pPolConcav->getPtRectAngle());
    qreal fMinLength = ln1.length();
    ptClosest = pRecCircle->topLeft();
    if(ln2.length() < fMinLength)
    {
        fMinLength = ln2.length();
        ptClosest = pRecCircle->topRight();
    }
    if(ln3.length() < fMinLength)
    {
        fMinLength = ln3.length();
        ptClosest = pRecCircle->bottomRight();
    }
    if( ln4.length() < fMinLength)
    {
        fMinLength = ln4.length();
        ptClosest = pRecCircle->bottomLeft();
    }

    //Comprobar el cuadrante.
    QPointF ptRecAngle = pPolConcav->getPtRectAngle();
    QPointF ptCenter = pPolConcav->getPtCenter();
    QPointF ptCenterCircle = pRecCircle->center();
    if(pPolConcav->getQuadrant() == 1)
    {
        if((ptCenterCircle.x() < ptRecAngle.x()) && (ptCenterCircle.y() > ptRecAngle.y()))
            return true;
        else
            return false;
    }
}

```

```

    }
    else if(pPolConcav->getQuadrant() == 2)
    {
        if((ptCenterCircle.x() > ptRecAngle.x()) && (ptCenterCircle.y()
> ptRecAngle.y()))
            return true;
        else
            return false;
    }
    else if(pPolConcav->getQuadrant() == 3)
    {
        if((ptCenterCircle.x() > ptRecAngle.x()) && (ptCenterCircle.y()
< ptRecAngle.y()))
            return true;
        else
            return false;
    }
    else
    {
        if((ptCenterCircle.x() < ptRecAngle.x()) && (ptCenterCircle.y()
< ptRecAngle.y()))
            return true;
        else
            return false;
    }
}

```

### **Ant.h**

```

#ifndef ANT_H
#define ANT_H

#include <QGraphicsPixmapItem>
#include <QPixmap>
#include <QList>
#include <QPointF>
#include <QTimer>
#include <QGraphicsScene>

class CAnt : public QObject, public QGraphicsPixmapItem
{

```

```
Q_OBJECT

public:
    enum EDirection{GOING_NEST, GOING_KEY};

    explicit CAnt(QPixmap px, int nest, QGraphicsScene* scene, QGraphicsItem * parent = 0 );
    void render();
    void paint ( QPainter * painter, const QStyleOptionGraphicsItem * option, QWidget * widget = 0);
    QList<QString>* pLKeysPressed;
    int pOrigNest; //Hormiguero. De 1 a DATA.numNests

    QPointF pCenter;
    QPointF pOrigPos;
    qreal pRadius;
    EDirection pDir;
    QPointF pNestPos;
    QPointF pScenePosCenter;

private:
    qreal pAngle;
    QPixmap pOrigPixmap;
    QPointF pDestPos;
    qreal pDist;
    QTimer pTimer;
    QGraphicsScene* pScene;

    qreal calcDistance(QPointF ptDest);
    qreal sqr(float number);
    QPointF gimmeNestPos();
    QPointF gimmeKeyPos(QString s);
    void setRandToRoute();

signals:

public slots:
    void changeRoute();
    void goNest();

};

#endif // ANT_H
```

**Ant.cpp**

```

#include "ant.h"
#include "DATA.h"
#include <QList>
#include <qmath.h>
#include <QLineF>
#include <QPainter>
#include <QDebug>
#include <QTime>

CAnt::CAnt(QPixmap px, int nest, QGraphicsScene* scene, QGraphicsItem
* parent) :
    QGraphicsPixmapItem(px, parent)
{
    pAngle = 0;
    pCenter.setX(px.width()/2);
    pCenter.setY(px.height() / 2);
    pScene = scene;
    this->pOrigNest = nest;
    this->pNestPos = this->gimmeNestPos();
    pOrigPos = pNestPos;
    this->setPos(pNestPos.x() - pCenter.x(), pNestPos.y() - pCenter.y());
    pOrigPixmap = px;
    pAngle = 0;
    pRadius = sqr(qPow(pCenter.x(), 2) + qPow(pCenter.y(), 2));
    this->pDestPos = pOrigPos;
    connect(&pTimer, SIGNAL(timeout()), this, SLOT(changeRoute()));
    pTimer.start(timeRoute);
    pDir = CAnt::GOING_KEY;
}

void CAnt::render()
{
    this->pScenePosCenter = this->pos();
    pOrigPos = pScenePosCenter;
    pScenePosCenter.setX(pScenePosCenter.x() + pCenter.x());
    pScenePosCenter.setY(pScenePosCenter.y() + pCenter.y());
    QPointF ptNextPos;
    qreal dist = calcDistance(this->pDestPos);
    qreal numSpeed = ((qreal)speed) / 10;
    if(dist < numSpeed)

```

```

    //Ya ha llegado a su destino.
    ptNextPos = pDestPos;
}
else
    //Avanza hacia el destino.
    QLineF ln(pScenePosCenter, pDestPos);
    ln.setLength(numSpeed);
    ptNextPos = ln.p2();
}
ptNextPos.setX(ptNextPos.x() - pCenter.x());
ptNextPos.setY(ptNextPos.y() - pCenter.y());
this->setPos(ptNextPos);
}

void CAnt::changeRoute()
{
    pDist = 666666666;
    for(int i = 0; i < pLKeysPressed->size(); i++)
    {
        QPointF pos = gimmeKeyPos(pLKeysPressed->at(i));
        qreal aux = calcDistance(pos);
        if(aux < pDist)
        {
            pDestPos.setX(pos.x());
            pDestPos.setY(pos.y());
            pDist = aux;
        }
        pDir = CAnt::GOING_KEY;
    }
    if(pLKeysPressed->isEmpty())
    //No hay teclas activadas, vuelve al hormiguero.
    int rand = qrand() % ((2500 + 1) - 0) + 0;
    QTimer::singleShot(rand, this, SLOT(goNest()));
    return;
}

setRandToRoute();
}

void CAnt::goNest()
{
    pDestPos.setX(pNestPos.x());
    pDestPos.setY(pNestPos.y());
    pDist = calcDistance(pDestPos);
}

```

```

        pDir = CAnt::GOING_NEST;
        setRandToRoute();
    }

void CAnt::setRandToRoute()
{
    int spRand = spaceRand;
    // if(dist <= 200)
        //spRand = spaceRand / 10;
    if(pDist <= 200)
        spRand = spaceRand / 2;
    int randX = qrand() % ((spRand + 1) - -spRand) + -spRand;
    int randY = qrand() % ((spRand + 1) - -spRand) + -spRand;
    int randTime = qrand() % ((timeRand + 1) - -timeRand) + -timeRand;
    pDestPos.setX(pDestPos.x() + randX);
    pDestPos.setY(pDestPos.y() + randY);
    QLineF ln(pScenePosCenter, pDestPos);
    pAngle = ln.angle();
    pTimer.start(timeRoute+randTime);
}

void CAnt::paint(QPainter * painter, const QStyleOptionGraphicsItem *
option, QWidget * widget)
{
    painter->translate(this->pCenter.x(), this->pCenter.y());
    painter->rotate(-this->pAngle);
    painter->translate(-this->pCenter.x(), -this->pCenter.y());
    QGraphicsPixmapItem::paint(painter, option, widget);
}

qreal CAnt::calcDistance(QPointF ptDest)
{
    return sqrt(qPow(ptDest.x() - pScenePosCenter.x(), 2) + qPow(ptDest.y()
- pScenePosCenter.y(), 2));
}

QPointF CAnt::gimmeKeyPos(QString s)
{
    //LETRAS:
    //W-Q-E-R
    //X-Z-C-V
    //2-1-3-4
    int w = this->pScene->width() / 5;
    int h = this->pScene->height() / 4;
}

```

```
QPointF pt;
if(s == "1")
{
    pt.setX((qreal)_1x + 2*w);
    pt.setY((qreal)_1y + h*3);
}
else if(s == "2")
{
    pt.setX((qreal)_2x + w);
    pt.setY((qreal)_2y + h*3);
}
else if(s == "3")
{
    pt.setX((qreal)_3x + 3*w);
    pt.setY((qreal)_3y + 3*h);
}
else if(s == "4")
{
    pt.setX((qreal)_4x + 4*w);
    pt.setY((qreal)_4y + 3*h);
}
else if(s == "5")
{
    pt.setX((qreal)_5x);
    pt.setY((qreal)_5y);
}
else if(s == "6")
{
    pt.setX((qreal)_6x);
    pt.setY((qreal)_6y);
}
else if(s == "7")
{
    pt.setX((qreal)_7x);
    pt.setY((qreal)_7y);
}
else if(s == "8")
{
    pt.setX((qreal)_8x);
    pt.setY((qreal)_8y);
}
else if(s == "A")
{
```

```
        pt.setX((qreal) Ax);
        pt.setY((qreal) Ay);
    }
    else if(s == "B")
    {
        pt.setX((qreal) Bx);
        pt.setY((qreal) By);
    }
    else if(s == "C")
    {
        pt.setX((qreal) Cx + 3*w);
        pt.setY((qreal) Cy + h*2);
    }
    else if(s == "D")
    {
        pt.setX((qreal) Dx);
        pt.setY((qreal) Dy);
    }
    else if(s == "E")
    {
        pt.setX((qreal) Ex + 3*w);
        pt.setY((qreal) Ey + h);
    }
    else if(s == "F")
    {
        pt.setX((qreal) Fx);
        pt.setY((qreal) Fy);
    }
    else if(s == "G")
    {
        pt.setX((qreal) Gx);
        pt.setY((qreal) Gy);
    }
    else if(s == "H")
    {
        pt.setX((qreal) Hx);
        pt.setY((qreal) Hy);
    }
    else if(s == "I")
    {
        pt.setX((qreal) Ix);
        pt.setY((qreal) Iy);
    }
    else if(s == "J")
```



```
{
    pt.setX((qreal) Jx);
    pt.setY((qreal) Jy);
}
else if(s == "K")
{
    pt.setX((qreal) Kx);
    pt.setY((qreal) Ky);
}
else if(s == "L")
{
    pt.setX((qreal) Lx);
    pt.setY((qreal) Ly);
}
else if(s == "M")
{
    pt.setX((qreal) Mx);
    pt.setY((qreal) My);
}
else if(s == "N")
{
    pt.setX((qreal) Nx);
    pt.setY((qreal) Ny);
}
else if(s == "O")
{
    pt.setX((qreal) Ox);
    pt.setY((qreal) Oy);
}
else if(s == "P")
{
    pt.setX((qreal) Px);
    pt.setY((qreal) Py);
}
else if(s == "Q")
{
    pt.setX((qreal) Qx + 2*w);
    pt.setY((qreal) Qy + h);
}
else if(s == "R")
{
    pt.setX((qreal) Rx + 4*w);
    pt.setY((qreal) Ry + h);
}
```

```
else if(s == "S")
{
    pt.setX((qreal) Sx);
    pt.setY((qreal) Sy);
}
else if(s == "T")
{
    pt.setX((qreal) Tx);
    pt.setY((qreal) Ty);
}
else if(s == "U")
{
    pt.setX((qreal) Ux);
    pt.setY((qreal) Uy);
}
else if(s == "V")
{
    pt.setX((qreal) Vx + 4*w);
    pt.setY((qreal) Vy + 2*h);
}
else if(s == "W")
{
    pt.setX((qreal) Wx + w);
    pt.setY((qreal) Wy + h);
}
else if(s == "X")
{
    pt.setX((qreal) Xx + w);
    pt.setY((qreal) Xy + h*2);
}
else if(s == "Y")
{
    pt.setX((qreal) Yx);
    pt.setY((qreal) Yy);
}
else if(s == "Z")
{
    pt.setX((qreal) Zx + 2*w);
    pt.setY((qreal) Zy + h*2);
}
return pt;
}
```

QPointF CAnt::gimmeNestPos()

```
{
    QPointF pt;
    if(pOrigNest == 1)
    {
        pt.setX(0 + (qreal) nest1X);
        pt.setY(0 + (qreal) nest1Y);
    }
    else if(pOrigNest == 2)
    {
        pt.setX(pScene->width() - (qreal) nest2X);
        pt.setY(0 + (qreal) nest2Y);
    }
    else if(pOrigNest == 3)
    {
        pt.setX(pScene->width() - (qreal) nest3X);
        pt.setY(pScene->height() - (qreal) nest3Y);
    }
    else if(pOrigNest == 4)
    {
        pt.setX(0 + (qreal) nest4X);
        pt.setY(pScene->height() - (qreal) nest4Y);
    }
    return pt;
}
```

```
qreal CAnt::sqr(float number)
{
    long i;
    float x, y;
    const float f = 1.5F;

    x = number * 0.5F;
    y = number;
    i = * ( long * ) &y;
    i = 0x5f3759df - ( i >> 1 );
    y = * ( float * ) &i;
    y = y * ( f - ( x * y * y ) );
    y = y * ( f - ( x * y * y ) );
    return number * y;
}
```

**MainWindow.h**

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QList>
#include <QTime>
#include "QGraphicsTextItem"

class CAnt;
namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

    void keyPressEvent (QKeyEvent *);
    void keyReleaseEvent(QKeyEvent *);

private:
    Ui::MainWindow* ui;
    QGraphicsScene* pScenePREV;
    QGraphicsScene* pScene;
    QList<CAnt*> pLAnts;
    QList<QString> pLKeysPressed; //Teclas presionadas a cada momen-
to.
    QList<int> pLNestPopulation; //Población de cada uno de los hormi-
gueros.
    QTime pTimer;
    int pFramesCount;
    QGraphicsTextItem* pSpaceBarText;
    int pState; //0 titulo, 1 hormigazz
    bool pLettersVisible;
    QList<QGraphicsTextItem*> pListItems;
    QList<QGraphicsRectItem*> pListItemsRect;

```

```

void setupUI();
CAnt* createAnt(int nest);
void checkCollision(CAnt* ant);
qreal sqr(float number);
qreal calcDistance(QPointF orig, QPointF dest);
QPointF gimmeNestPos(int nest);
QPointF gimmeKeyPos(QString s);
void hideLetters();
void showLetters();

private slots:
void renderTimer();
};

#endif // MAINWINDOW_H

```

### ***MainWindow.cpp***

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "QKeyEvent"
#include "ant.h"
#include "DATA.h"
#include "qmath.h"
#include <QTimer>
#include "QDebug"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    pFramesCount = 0;
    this->setFocusPolicy(Qt::StrongFocus);
    //setupUI();
    QTime time = QTime::currentTime();
    qsrand((uint)time.msec());
    pState = 0;
    pLettersVisible = false;

    ui->view->setInteractive(false);

```

```

        ui->view->setViewportUpdateMode(QGraphicsView::FullViewportUpdate);
    }
    pScenePREV = new QGraphicsScene(ui->view);
    ui->view->setScene(this->pScenePREV);
    this->pSpaceBarText = pScenePREV->addText("Pulsa barra espaciadora
para comenzar");
    pSpaceBarText->setPos(pScenePREV->width() / 2 - 80, pScenePREV->
height() / 2);
}

MainWindow::~MainWindow()
{
    delete ui;
    pScene = 0;
    pScenePREV = 0;
    pSpaceBarText = 0;
}

void MainWindow::renderTimer()
{
    pTimer.start();
    pFramesCount++;
    if(this->pFramesCount == 25)
        pFramesCount = 0;

    //Crear homigas si hay botón pulsado, quedan hormigas por salir
    //y no hay ninguna a una distancia especifica.
    if(pLKeysPressed.size() > 0)
    {
        for(int i = 0; i < this->pLNestPopulation.size(); i++)
        {
            if(pScene->itemAt(gimmeNestPos(i+1)) == NULL)
                this->createAnt(i+1);
        }
    }

    //Render
    QList<CAnt*> lAntsToRemove;
    for(int i = 0; i < pLAnts.size(); i++)
    {
        pLAnts.at(i)->render();
        //Eliminar si entra en el hormiguero
        if(calcDistance(pLAnts.at(i)->pNestPos, pLAnts.at(i)->pScenePosCenter)

```

```

<= 50)
{
    if(pLAnts.at(i)->pDir == CAnt::GOING_NEST)
    {
        lAntsToRemove.append(pLAnts.at(i));
    }
}
//checkCollision(pLAnts.at(i));
}
for(int i = 0; i < lAntsToRemove.size(); i++)
{
    this->pScene->removeItem(lAntsToRemove.at(i));
    pLAnts.removeOne(lAntsToRemove.at(i));
    int nest = lAntsToRemove.at(i)->pOrigNest;
    this->pLNestPopulation[nest-1] += 1;
}

int iElapsed = 40-pTimer.elapsed(); //20 == 50 fps
if(iElapsed > 0)
{
    QTimer::singleShot(iElapsed, this, SLOT(renderTimer()));
}
else
{//No nos quedamos fuera de tiempo pase lo que pase.
    qDebug() << "¡¡¡MAAAAL!!! Ha tardado " << iElapsed << " ms!!!!!!!!";
    QTimer::singleShot(15, this, SLOT(renderTimer()));
}
}

void MainWindow::setupUI()
{
    //Scene
    pScene = new QGraphicsScene(ui->view);
    pScene->setSceneRect(0, 0, ui->view->width(), ui->view->height());
    //View
    ui->view->setScene(pScene);

    pScenePREV->removeItem(pSpaceBarText);
    QGraphicsTextItem* txt = 0;
    QPointF pos;

    txt = pScene->addText("W");
    pListItems.append(txt);
    txt->setScale(2);

```

```
pos = this->gimmeKeyPos("W");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("Q");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("Q");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("E");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("E");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("R");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("R");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("X");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("X");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("Z");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("Z");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```



```
txt = pScene->addText("C");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("C");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("V");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("V");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("2");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("2");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("1");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("1");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("3");
pListItems.append(txt);
txt->setScale(2);
pos = this->gimmeKeyPos("3");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);
```

```
txt = pScene->addText("4");
pListItems.append(txt);
txt->setScale(2);
```

```

pos = this->gimmeKeyPos("4");
txt->setPos(pos);
pListItemsRect.append(pScene->addRect(pos.x(), pos.y(), 30, 30));
txt->setVisible(false);

this->hideLetters();

for(int i = 0; i < numNests; i++)
    pLNestPopulation.append((qreal)antsPerNest);}

void MainWindow::checkCollision(CAnt* ant)
{
    QPointF ptPos = ant->pos();
    ptPos.setX(ptPos.x() + ant->pCenter.x());
    ptPos.setY(ptPos.y() + ant->pCenter.y());
    for(int i = 0; i < this->pLAnts.size(); i++)
    {
        CAnt* antAux = this->pLAnts.at(i);
        if(antAux == ant)
            continue;
        QPointF ptOther = antAux->pos();
        ptOther.setX(ptOther.x() + antAux->pCenter.x());
        ptOther.setY(ptOther.y() + antAux->pCenter.y());
        qreal dist = this->calcDistance(ptPos, ptOther);
        if(dist < (antAux->pRadius + ant->pRadius))
           {//chocan
                ant->setPos(ant->pOrigPos);
                break;
            }
    }
}

void MainWindow::keyPressEvent(QKeyEvent* ev)
{
    if(!ev->isAutoRepeat())
    {
        bool setToAnts = false;
        if(ev->key() == Qt::Key_Space)
        {
            if(pState == 0)
            {
                pState = 1;
                setupUI();
                renderTimer();
            }
        }
    }
}

```

```
    }  
  }  
  if(ev->key() == Qt::Key_Escape)  
  {  
    this->close();  
  }  
  else if(ev->key() == Qt::Key_W)  
  {  
    if(!pLKeysPressed.contains("W"))  
    {  
      pLKeysPressed.append("W");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_Q)  
  {  
    if(!pLKeysPressed.contains("Q"))  
    {  
      pLKeysPressed.append("Q");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_E)  
  {  
    if(!pLKeysPressed.contains("E"))  
    {  
      pLKeysPressed.append("E");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_R)  
  {  
    if(!pLKeysPressed.contains("R"))  
    {  
      pLKeysPressed.append("R");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_X)  
  {  
    if(!pLKeysPressed.contains("X"))  
    {  
      pLKeysPressed.append("X");  
      setToAnts = true;  
    }  
  }  
}
```

```
    }  
  }  
  else if(ev->key() == Qt::Key_Z)  
  {  
    if(!pLKeysPressed.contains("Z"))  
    {  
      pLKeysPressed.append("Z");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_C)  
  {  
    if(!pLKeysPressed.contains("C"))  
    {  
      pLKeysPressed.append("C");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_V)  
  {  
    if(!pLKeysPressed.contains("V"))  
    {  
      pLKeysPressed.append("V");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_2)  
  {  
    if(!pLKeysPressed.contains("2"))  
    {  
      pLKeysPressed.append("2");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_1)  
  {  
    if(!pLKeysPressed.contains("1"))  
    {  
      pLKeysPressed.append("1");  
      setToAnts = true;  
    }  
  }  
  else if(ev->key() == Qt::Key_3)  
  {
```

```
        if(!pLKeysPressed.contains("3"))
        {
            pLKeysPressed.append("3");
            setToAnts = true;
        }
    }
    else if(ev->key() == Qt::Key_4)
    {
        if(!pLKeysPressed.contains("4"))
        {
            pLKeysPressed.append("4");
            setToAnts = true;
        }
    }
    else if(ev->key() == Qt::Key_P)
    {
        if(pLettersVisible)
            hideLetters();
        else
            showLetters();
    }
    if(setToAnts)
    {
        for(int i = 0; i < this->pLAnts.size(); i++)
        {
            pLAnts.at(i)->pLKeysPressed = &pLKeysPressed;
        }
    }
    QMainWindow::keyPressEvent(ev);
    this->setFocus();
}
}

void MainWindow::keyReleaseEvent(QKeyEvent* ev)
{
    if(!ev->isAutoRepeat())
    {
        if(ev->key() == Qt::Key_W)
        {
            pLKeysPressed.removeOne("W");
        }
        else if(ev->key() == Qt::Key_Q)
        {
            pLKeysPressed.removeOne("Q");
        }
    }
}
```

```
}
else if(ev->key() == Qt::Key_E)
{
    pLKeysPressed.removeOne("E");
}
else if(ev->key() == Qt::Key_R)
{
    pLKeysPressed.removeOne("R");
}
else if(ev->key() == Qt::Key_X)
{
    pLKeysPressed.removeOne("X");
}
else if(ev->key() == Qt::Key_Z)
{
    pLKeysPressed.removeOne("Z");
}
else if(ev->key() == Qt::Key_C)
{
    pLKeysPressed.removeOne("C");
}
else if(ev->key() == Qt::Key_V)
{
    pLKeysPressed.removeOne("V");
}
else if(ev->key() == Qt::Key_2)
{
    pLKeysPressed.removeOne("2");
}
else if(ev->key() == Qt::Key_1)
{
    pLKeysPressed.removeOne("1");
}
else if(ev->key() == Qt::Key_3)
{
    pLKeysPressed.removeOne("3");
}
else if(ev->key() == Qt::Key_4)
{
    pLKeysPressed.removeOne("4");
}
for(int i = 0; i < this->pLAnts.size(); i++)
{
    pLAnts.at(i)->pLKeysPressed = &pLKeysPressed;
```

```
    }
    QMainWindow::keyPressEvent(ev);
    this->setFocus();
}

void MainWindow::hideLetters()
{
    this->pLettersVisible = false;
    for(int i = 0; i < pListItems.size(); i++)
    {
        pListItems.at(i)->setVisible(false);
        pListItemsRect.at(i)->setVisible(false);
    }
}

void MainWindow::showLetters()
{
    this->pLettersVisible = true;
    for(int i = 0; i < pListItems.size(); i++)
    {
        pListItems.at(i)->setVisible(true);
        pListItemsRect.at(i)->setVisible(true);
    }
}

CAnt* MainWindow::createAnt(int nest)
{
    if(pLNestPopulation.at(nest-1) == 0)
        return NULL;
    CAnt* ant;
    if(nest == 1)
        ant = new CAnt(QPixmap(":/img/ant_img_1"), nest, this->pScene);
    else if(nest == 2)
        ant = new CAnt(QPixmap(":/img/ant_img_2"), nest, this->pScene);
    else if(nest == 3)
        ant = new CAnt(QPixmap(":/img/ant_img_3"), nest, this->pScene);
    else if(nest == 4)
        ant = new CAnt(QPixmap(":/img/ant_img_4"), nest, this->pScene);

    this->pScene->addItem(ant);
    this->pLAnts.append(ant);
    this->pLNestPopulation[nest-1] -= 1;
```

```
    ant->pLKeysPressed = &this->pLKeysPressed;
    return ant;
}
```

```
QPointF MainWindow::gimmeNestPos(int nest)
{
    QPointF pt;
    if(nest == 1)
    {
        pt.setX(0 + (qreal) nest1X);
        pt.setY(0 + (qreal) nest1Y);
    }
    else if(nest == 2)
    {
        pt.setX(pScene->width() - (qreal) nest2X);
        pt.setY(0 + (qreal) nest2Y);
    }
    else if(nest == 3)
    {
        pt.setX(pScene->width() - (qreal) nest3X);
        pt.setY(pScene->height() - (qreal) nest3Y);
    }
    else if(nest == 4)
    {
        pt.setX(0 + (qreal) nest4X);
        pt.setY(pScene->height() - (qreal) nest4Y);
    }
    /*else if(nest == 5)
    {
        if(nest5X == 9999)
            pt.setX(pScene->width());
        else if(nest5X == 5555)
            pt.setX(pScene->width()/2);
        else
            pt.setX(nest5X);
        if(nest5Y == 9999)
            pt.setY(pScene->height());
        else if(nest5Y == 5555)
            pt.setY(pScene->height()/2);
        else
            pt.setY(nest5Y);
    }
    else if(nest == 6)
    {
```



```

        if(nest6X == 9999)
            pt.setX(pScene->width());
        else if(nest6X == 5555)
            pt.setX(pScene->width()/2);
        else
            pt.setX(nest6X);
        if(nest6Y == 9999)
            pt.setY(pScene->height());
        else if(nest6Y == 5555)
            pt.setY(pScene->height()/2);
        else
            pt.setY(nest6Y);
    }*/
    return pt;
}

```

```

QPointF MainWindow::gimmeKeyPos(QString s)
{
    //LETRAS:
    //W-Q-E-R
    //X-Z-C-V
    //2-1-3-4
    int w = this->pScene->width() / 5;
    int h = this->pScene->height() / 4;

    QPointF pt;
    if(s == "1")
    {
        pt.setX((qreal)_1x + 2*w);
        pt.setY((qreal)_1y + h*3);
    }
    else if(s == "2")
    {
        pt.setX((qreal)_2x + w);
        pt.setY((qreal)_2y + h*3);
    }
    else if(s == "3")
    {
        pt.setX((qreal)_3x + 3*w);
        pt.setY((qreal)_3y + 3*h);
    }
    else if(s == "4")
    {
        pt.setX((qreal)_4x + 4*w);
    }
}

```

```
    pt.setY((qreal)_4y + 3*h);
}
else if(s == "5")
{
    pt.setX((qreal)_5x);
    pt.setY((qreal)_5y);
}
else if(s == "6")
{
    pt.setX((qreal)_6x);
    pt.setY((qreal)_6y);
}
else if(s == "7")
{
    pt.setX((qreal)_7x);
    pt.setY((qreal)_7y);
}
else if(s == "8")
{
    pt.setX((qreal)_8x);
    pt.setY((qreal)_8y);
}
else if(s == "A")
{
    pt.setX((qreal) Ax);
    pt.setY((qreal) Ay);
}
else if(s == "B")
{
    pt.setX((qreal) Bx);
    pt.setY((qreal) By);
}
else if(s == "C")
{
    pt.setX((qreal) Cx + 3*w);
    pt.setY((qreal) Cy + h*2);
}
else if(s == "D")
{
    pt.setX((qreal) Dx);
    pt.setY((qreal) Dy);
}
else if(s == "E")
{
```

```
    pt.setX((qreal) Ex + 3*w);
    pt.setY((qreal) Ey + h);
}
else if(s == "F")
{
    pt.setX((qreal) Fx);
    pt.setY((qreal) Fy);
}
else if(s == "G")
{
    pt.setX((qreal) Gx);
    pt.setY((qreal) Gy);
}
else if(s == "H")
{
    pt.setX((qreal) Hx);
    pt.setY((qreal) Hy);
}
else if(s == "I")
{
    pt.setX((qreal) Ix);
    pt.setY((qreal) Iy);
}
else if(s == "J")
{
    pt.setX((qreal) Jx);
    pt.setY((qreal) Jy);
}
else if(s == "K")
{
    pt.setX((qreal) Kx);
    pt.setY((qreal) Ky);
}
else if(s == "L")
{
    pt.setX((qreal) Lx);
    pt.setY((qreal) Ly);
}
else if(s == "M")
{
    pt.setX((qreal) Mx);
    pt.setY((qreal) My);
}
else if(s == "N")
```

```
{
    pt.setX((qreal) Nx);
    pt.setY((qreal) Ny);
}
else if(s == "O")
{
    pt.setX((qreal) Ox);
    pt.setY((qreal) Oy);
}
else if(s == "P")
{
    pt.setX((qreal) Px);
    pt.setY((qreal) Py);
}
else if(s == "Q")
{
    pt.setX((qreal) Qx + 2*w);
    pt.setY((qreal) Qy + h);
}
else if(s == "R")
{
    pt.setX((qreal) Rx + 4*w);
    pt.setY((qreal) Ry + h);
}
else if(s == "S")
{
    pt.setX((qreal) Sx);
    pt.setY((qreal) Sy);
}
else if(s == "T")
{
    pt.setX((qreal) Tx);
    pt.setY((qreal) Ty);
}
else if(s == "U")
{
    pt.setX((qreal) Ux);
    pt.setY((qreal) Uy);
}
else if(s == "V")
{
    pt.setX((qreal) Vx + 4*w);
    pt.setY((qreal) Vy + 2*h);
}
```

```

else if(s == "W")
{
    pt.setX((qreal) Wx + w);
    pt.setY((qreal) Wy + h);
}
else if(s == "X")
{
    pt.setX((qreal) Xx + w);
    pt.setY((qreal) Xy + h*2);
}
else if(s == "Y")
{
    pt.setX((qreal) Yx);
    pt.setY((qreal) Yy);
}
else if(s == "Z")
{
    pt.setX((qreal) Zx + 2*w);
    pt.setY((qreal) Zy + h*2);
}
return pt;
}

qreal MainWindow::calcDistance(QPointF orig, QPointF dest)
{
    return sqrt(qPow(dest.x() - orig.x(), 2) + qPow(dest.y() - orig.y(), 2));
}

qreal MainWindow::sqr(float number)
{
    long i;
    float x, y;
    const float f = 1.5F;

    x = number * 0.5F;
    y = number;
    i = * ( long * ) &y;
    i = 0x5f3759df - ( i >> 1 );
    y = * ( float * ) &i;
    y = y * ( f - ( x * y * y ) );
    y = y * ( f - ( x * y * y ) );
    return number * y;
}

```

**DATA.h**

```
#ifndef DATA_H
#define DATA_H

#define speed 30 // /10
#define timeRoute 600
#define timeRand 200
#define spaceRand 100
#define antsPerNest 200

//Las letras en pantalla se ponen formando una malla.
//Estos números ajustan su posición. Si x es 20 quiere decir que estará
movido 20
//píxels a la derecha de donde le toca.
//Recordar que Y creciente baja, Y decreciente sube. El punto 0,0 es la
esquina
//superior izquierda.
//HORMIGUEROS:
// 1: arr-izq, 2: arr-der, 3: abajo-der, 4: abajo-izq
#define numNests 4
#define nest1X 50
#define nest1Y 50
#define nest2X 50
#define nest2Y 50
#define nest3X 50
#define nest3Y 50
#define nest4X 50
#define nest4Y 50
#define nest5X -666
#define nest5Y -666
#define nest6X -666
#define nest6Y -666

//Las letras en pantalla se ponen formando una malla.
//Estos números ajustan su posición. Si x es 20 quiere decir que estará
movido 20
//píxels a la derecha de donde le toca.
//Recordar que Y creciente baja, Y decreciente sube. El punto 0,0 es la
esquina
//superior izquierda.
//LETRAS:
```

```
//W-Q-E-R
//X-Z-C-V
//2-1-3-4
#define _1x 0
#define _1y 0
#define _2x 0
#define _2y 0
#define _3x 0
#define _3y 0
#define _4x 0
#define _4y 0
#define _5x 0
#define _5y 0
#define _6x 0
#define _6y 0
#define _7x 0
#define _7y 0
#define _8x 0
#define _8y 0
#define Ax 0
#define Ay 0
#define Bx 0
#define By 0
#define Cx 0
#define Cy 0
#define Dx 300
#define Dy 200
#define Ex 0
#define Ey 0
#define Fx 0
#define Fy 0
#define Gx 100
#define Gy 100
#define Hx 0
#define Hy 0
#define Ix 0
#define Iy 0
#define Jx 0
#define Jy 0
#define Kx 0
#define Ky 0
#define Lx 0
#define Ly 0
#define Mx 0
```

```
#define My 0
#define Nx 0
#define Ny 0
#define Ox 0
#define Oy 0
#define Px 0
#define Py 0
#define Qx 0
#define Qy 0
#define Rx 0
#define Ry 0
#define Sx 0
#define Sy 0
#define Tx 0
#define Ty 0
#define Ux 0
#define Uy 0
#define Vx 0
#define Vy 0
#define Wx 0
#define Wy 0
#define Xx 0
#define Xy 0
#define Yx 0
#define Yy 0
#define Zx 0
#define Zy 0
#define BlocMayx 0
#define BlocMayy 0
#define Tabx 0
#define Taby 0
#define F3x 0
#define F3y 0

#endif // DATA_H
```



## 6.2 CÓDIGO DE *PREPARADOS, LISTOS... LUZ!*

```
int sensor1 = 0;
int sensor2 = 0;
int angulo1 = 400;//Probar los valores allí, a lo mejor es mejor un acelerómetro
int angulo2 = 450;
int angulo3 = 500;
int angulo4 = 550;
String estado = "seltas";
int boton = 1025; //por resolver
int tiempoPartida = 3000;
boolean jugando = false;
int puntos = 0;
int tiempoInicial = 0;
int tiempoActual = 0;

void setup(){

  Serial.begin(9600);

  //alfombra1
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);

  //alfombra2
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);

  //puntos
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);

  //boton = 1;//para las pruebas

}
```

```

void loop(){

    boton = analogRead(0);//lo hacemos analógico
    //Serial.println("Boton = "+boton);
    if (boton < 1020 && jugando == false){ //Cambiar botón al valor que tenga lo que sea que conectemos.... ufffff estoy preocupadaa!!
        estado="palmada";
        EscaleraLuces();
        EscaleraLuces();
        EscaleraLuces();
        EscaleraLuces();
        //boton= 2000;//Para las pruebas
        Serial.println("boton");
        Serial.println(boton);
        jugando = true;

        int puntos = 0;
        tiempoInicial = int(millis()/10);
    }
    //Funcionamiento normal, no estará a false hasta que pase x tiempo o consigan los 5 puntos.
    if(jugando == true){
        Serial.println("Jugando");
        sensor1 = analogRead(5);
        Serial.println("SENSOR");
        Serial.println(sensor1);
        //delay(250);
        sensor2 = analogRead(4);
        RecorridoLuces(sensor1,sensor2);
        if(boton <1020 && estado.equals("sueltas")){
            estado = "palmada";
            puntos++;
            LucesPuntos();//se puede repetir
            if(puntos == 4){
                jugando = false;
                LucesPuntos();
                FiestaFinal();//Efecto luces, deben de terminar apagadas
                delay(200);
                puntos=0;
                LucesPuntos();
            }
        }
        else{
            CelebracionPunto();}
    }
}

```

```

    }
    tiempoActual=int(millis()/10);
    Serial.println("tiempoActual");
    Serial.println(tiempoActual - tiempoInicial);
    if(tiempoActual - tiempoInicial >= tiempoPartida){
        puntos=0;
        LucesPuntos();
        LucesDerrota();

        jugando = false;}
    }
    if(boton > 1020){estado = "seltas";}
}

```

```

void LucesPuntos(){
    Serial.println("LucesPuntos");
    switch(puntos){
        case 0:{
            digitalWrite(10,LOW);
            digitalWrite(11,LOW);
            digitalWrite(12,LOW);
            digitalWrite(13,LOW);
            break;}
        case 1:{
            digitalWrite(10,HIGH);
            digitalWrite(11,LOW);
            digitalWrite(12,LOW);
            digitalWrite(13,LOW);
            break;}
        case 2:{
            digitalWrite(10,HIGH);
            digitalWrite(11,HIGH);
            digitalWrite(12,LOW);
            digitalWrite(13,LOW);
            break;}
        case 3:{
            digitalWrite(10,HIGH);
            digitalWrite(11,HIGH);
            digitalWrite(12,HIGH);
            digitalWrite(13,LOW);
            break;}
        case 4:{

```

```
        digitalWrite(10,HIGH);
        digitalWrite(11,HIGH);
        digitalWrite(12,HIGH);
        digitalWrite(13,HIGH);
        break;}}
}

void RecorridoLuces(int sensor1, int sensor2){
  Serial.println("RecorridoLuces");
  if(sensor1<=angulo1){
    EncenderUnoA1(2);}

  if((sensor1>angulo1)&&(sensor1<angulo2)){
    EncenderUnoA1(3);}

  if(sensor1>angulo2&&sensor1<angulo3){
    EncenderUnoA1(4);}

  if(sensor1>angulo3){
    EncenderUnoA1(5);}

  //Alfombra2
  if(sensor2<angulo1){
    EncenderUnoA2(6);}

  if(sensor2>angulo1&&sensor2<angulo2){
    EncenderUnoA2(7);}

  if(sensor2>angulo2&&sensor2<angulo3){
    EncenderUnoA2(8);}

  if(sensor2>=angulo3){
    EncenderUnoA2(9);}
}

void EscaleraLuces(){
  Serial.println("EscaleraLuces");
  EncenderUnoA1(2);
  EncenderUnoA2(9);

  delay(100);
```

```
    EncenderUnoA1(3);
    EncenderUnoA2(8);

    delay(100);

    EncenderUnoA1(4);
    EncenderUnoA2(7);

    delay(100);

    EncenderUnoA1(5);
    EncenderUnoA2(6);

    delay(100);

    EncenderUnoA1(4);
    EncenderUnoA2(7);

    delay(100);

    EncenderUnoA1(3);
    EncenderUnoA2(8);

    delay(100);

    EncenderUnoA1(2);
    EncenderUnoA2(9);

    delay(100);

    EncenderUnoA1(99);
    EncenderUnoA2(99);

    delay(100);}

void CelebracionPunto(){

    EncenderUnoA1(32);
    EncenderUnoA2(32);

    delay(100);

    EncenderUnoA1(33);
    EncenderUnoA2(33);
```

```
delay(100);

  EncenderUnoA1(32);
  EncenderUnoA2(32);

delay(100);

  EncenderUnoA1(33);
  EncenderUnoA2(33);

delay(100);

  EncenderUnoA1(32);
  EncenderUnoA2(32);

delay(100);

  EncenderUnoA1(33);
  EncenderUnoA2(33);

delay(100);

  EncenderUnoA1(99);
  EncenderUnoA2(99);

Serial.println("CelebracionPunto");
}

void FiestaFinal(){
  Serial.println("FiestaFinal");
  EncenderUnoA1(0);
  EncenderUnoA2(0);

  delay(80);

  EncenderUnoA1(33);
  EncenderUnoA2(33);

  delay(80);
  EncenderUnoA1(32);
  EncenderUnoA2(32);
```

```
delay(80);  
EncenderUnoA1(0);  
EncenderUnoA2(0);  
  
delay(80);  
  
EncenderUnoA1(33);  
EncenderUnoA2(33);  
  
delay(80);  
EncenderUnoA1(32);  
EncenderUnoA2(32);  
  
delay(80);  
EncenderUnoA1(0);  
EncenderUnoA2(0);  
  
delay(80);  
  
EncenderUnoA1(33);  
EncenderUnoA2(33);  
  
delay(80);  
EncenderUnoA1(32);  
EncenderUnoA2(32);  
  
delay(80);  
EncenderUnoA1(0);  
EncenderUnoA2(0);  
  
delay(80);  
  
EncenderUnoA1(33);  
EncenderUnoA2(33);  
  
delay(80);  
EncenderUnoA1(32);  
EncenderUnoA2(32);  
  
delay(80);  
EncenderUnoA1(0);  
EncenderUnoA2(0);  
  
delay(80);
```

```
    EncenderUnoA1(33);
    EncenderUnoA2(33);

    delay(80);
    EncenderUnoA1(32);
    EncenderUnoA2(32);

    delay(80);
    EncenderUnoA1(0);
    EncenderUnoA2(0);

    delay(80);

    EncenderUnoA1(33);
    EncenderUnoA2(33);

    delay(80);
    EncenderUnoA1(32);
    EncenderUnoA2(32);

    delay(80);

    EncenderUnoA1(99);
    EncenderUnoA2(99);

}

void LucesDerrota(){
    Serial.println("LucesDerrota");
    EncenderUnoA1(0);
    EncenderUnoA2(0);

    delay(300);

    EncenderUnoA1(99);
    EncenderUnoA2(99);

    delay(100);

    EncenderUnoA1(0);
    EncenderUnoA2(0);

    delay(200);
```



```
    EncenderUnoA1(99);
    EncenderUnoA2(99);

    delay(200);

    EncenderUnoA1(0);
    EncenderUnoA2(0);

    delay(50);

    EncenderUnoA1(99);
    EncenderUnoA2(99);

    //delay(50);
}

void EncenderUnoA1(int pin){
    //return;
    switch (pin) {
    case 2:
        digitalWrite(2,HIGH);
        digitalWrite(3,LOW);
        digitalWrite(4,LOW);
        digitalWrite(5,LOW);
        break;
    case 3:
        digitalWrite(2,LOW);
        digitalWrite(3,HIGH);
        digitalWrite(4,LOW);
        digitalWrite(5,LOW);
        break;
    case 4:
        digitalWrite(2,LOW);
        digitalWrite(3,LOW);
        digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
        break;
    case 5:
        digitalWrite(2,LOW);
        digitalWrite(3,LOW);
        digitalWrite(4,LOW);
        digitalWrite(5,HIGH);
        break;
    }
```

```
case 99://Este los apaga todos
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  break;
case 0://Este los enciende todos
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  break;
case 32://Pares
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  break;
case 33://Impares
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  break;
}}

void EncenderUnoA2(int pin){
  //return;
  switch (pin) {
  case 6:
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
    break;
  case 7:
    digitalWrite(6,LOW);
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
    break;
  case 8:
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
```

```
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
break;
case 9:
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
digitalWrite(9,HIGH);
break;
case 99://Este los apaga todos
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
digitalWrite(9,LOW);
break;
case 0://Este los enciende todos
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
digitalWrite(9,HIGH);
break;
case 33://Impares
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
digitalWrite(9,HIGH);
break;
case 32://Impares
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
break;
}}
```

### 6.3 CÓDIGO DE QUIZ

```
final int POS_FUERA = 0;
final int POS_VERDADERO = 2;
final int POS_FALSO = 1;
int DOS_SOLUCIONES = 0;
```

String[] preg={"Los colores primarios son aquellos que pueden producir la mayor gama de colores, al mezclados entre sí.","La composición fue considerada principio fundamental en el arte europeo hasta principios del siglo XVII","La Proporción Áurea se puede ver en las nervaduras de las hojas, en el grosor de las ramas y en el cuerpo humano, entre otros.","La tangente a una curva en uno de sus puntos, es una recta que toca a la curva en el punto dado","Los tintes sirven para colorear superficies, como puede ser un cuadro, mientras que los pigmentos sirven para cubrir materiales, como los tejidos.","El ojo humano sólo percibe las longitudes de onda cuando la iluminación es abundante. Con poca luz se ve en blanco y negro.","Un pigmento o un tinte es un material que cambia el color de la luz que refleja","Todo lo que no es color aditivo es color sustractivo","Los colores complementarios producen un efecto de calma, provocado por la máxima analogía al utilizarlos juntos.";

"El (255,255,255) es el negro en RGB","El concepto de armonía del color se relaciona con la oposición de las tonalidades perceptuales del color","Función poética o estética: Consiste en aquella imagen que busca la belleza con sentido artístico, estético. Son imágenes que valen en sí mismas.","En el Renacimiento se estudia el método de reflejar la realidad con la mayor fidelidad posible, con arreglo a normas matemáticas y geométricas impecables","Los símbolos son dibujos que son reconocidos dentro de una cultura.","El apunte es el dibujo rápido que se usa para captar y recordar las características de lo que se va a dibujar después.","La definición de arte es abierta, subjetiva y discutible","El término arte procede del latín ars, y es el equivalente al término griego (téchne, de donde proviene 'técnica')","En la mitología griega, Jasón venció a un cíclope, resistió a las sirenas y consiguió volver junto a su esposa.","la invención y difusión de la escritura se dio a partir del fin de la vida nómada","La ley de la proporción Aurea es la denominación del modo peculiar empleado por los antiguos egipcios en sus representaciones escultóricas de la figura humana.","La música programática es la que tiene por objetivo evocar ideas o imágenes extra-musicales en la mente del oyente.","Kefrón, Kefrén y Kefrá son los nombres de las pirámides de Giza","El monumento a Colón de España está situado en Madrid","La pintura de 'El beso' se encuentra actualmente en Francia.","La fachada de la catedral de Santiago de Compostela es Gótica","Existen tres tipos de bóvedas: Cañón, media y alta.","¿En qué se diferencia un bodegón de la naturaleza muerta? En en color.","El arte y las guerras eran el tema principal en las artes plásticas egipcias","La alegoría es la representación de algo abstracto a través de

una imagen de algo concreto.”En los años 60, en el ámbito publicitario, se generaliza la noción de Dirección de Arte”Benjamín Franklin es el primero en incluir la publicidad en las portadas de los periódicos y en las primeras revistas americanas.”Schwitters es un artista alemán dadaísta propietario de una agencia de publicidad. Su actividad artística se funda sobre el collage y la escultura de deshecho.”A partir de los años 60, con la revolución creativa operada en la publicidad norteamericana, el arte predomina sobre la redacción”El plano, en geometría no tiene límites. En plástica tiene posición y dirección, pero no grosor. Puede definir los límites de un volumen.”La línea, en geometría no tiene límites. En plástica tiene posición y dirección, pero no grosor. Puede definir los límites de un volumen.”La ley de proximidad o lejanía dice que los elementos (puntos y líneas) se agrupan perceptivamente de modo que sigan el recorrido más suave.”La composición es la interacción que se produce entre los elementos situados sobre el capo visual en función del su tamaño, forma y posición relativa, tono y color.”La velocidad de obturación configura la sensibilidad de los sensores de la cámara.”En un plano picado la cámara se sitúa a la altura del objeto retratado.”Se establece como figuras simples las formas asimétricas, con giros, mayor simetría y ángulos menores.”La forma se basa en la captación de las relaciones que existen entre sus partes (Katz).”Las culturas tienen preferencias visuales arraigadas pero existe un sistema visual perceptivo básico que todos compartimos.”El rombo es una forma que proviene de la intersección de dos círculos, pero es una estructura independiente, con sus lados y diagonales. Su estructura modular es un enrejado de triángulos.”En la composición intervienen aspectos únicamente formales.”El arte rupestre era un arte realista y conceptual.”

“El humanismo es la corriente de pensamiento que articula el Renacimiento.”El daltonismo es un defecto genético que ocasiona dificultad para distinguir los colores amarillo y verde.”Sólo existe un tipo de daltonismo”

};

```
int[] resp= {1,0,1,1,0,1,1,1,0,0,0,1,1,1,1,1,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,
1,0,1,1,0,0,0,1,1,1,0,1,1,0,0};
```

```
int actual = 0;
```

```
int aciertos = 0;
```

```
int fallos = 0;
```

```
int verdadero = 0;
```

```
int falso = 0;
```

```
var x = new Array();
```

```
var y = new Array();
```

```
int sol = 0;
```

```
int bien, mal ,diametro = 0;
```

```
boolean dentro = false;
```

```
void setup() {
```

```
size(192,168);
stroke(255);
actual = int(random(preg.length));
PFont font;
font = loadFont("DINPro-Medium-8.vlw");
textFont(font);
textSize(9);
smooth();
tracking.connect();
}

void draw() {

// pos = posicion();
background(000);

Trakear();
ComprobarSolucion();

if(mal == 1){
  noStroke();
  if(sol != POS_FUERA){
    PintarRojo();
  }
  else{
    mal =0;
    diametro=0;
  }
}
else{
  if(bien == 1){
    noStroke();
    if(sol != POS_FUERA){
      PintarVerde();
    }
  }
  else{
    bien = 0;
    diametro=0;
  }
}

}

else {
```

```
if(DOS_SOLUCIONES == 1){
  fill(0,122,122);
  rect(15,15,177,153);
  fill(255);
  textAlign(CENTER);
  text("SÓLO UNA SOLUCIÓN", 90,80);
  text("POR FAVOR", 90,100);
  textAlign(LEFT);
  cont--;
  DOS_SOLUCIONES = 0;
}
```

Enunciado());

```
if (sol == POS_VERDADERO) {
//verdadero
if(resp[actual]==1){
  aciertos ++;
  bien=1;
  actual = int(random(preg.length));}
  else{
  fallos ++;
  mal =1;
  actual = int(random(preg.length));
  }
} else {
  if (sol == POS_FALSO) {
  //falso
  if(resp[actual]==0){
  aciertos ++;
  bien=1;
  actual = int(random(preg.length));
  }
  else{
  fallos ++;
  mal=1;
  actual = int(random(preg.length));
  }
  }
}
```

```
    }}}

    void Enunciado(){

        stroke(255);
        fill(22,258,22);
        text("Aciertos: "+aciertos, 45,30);
        fill(255,22,22);
        textAlign(RIGHT);
        text("Fallos: "+fallos, 150,30);
        textAlign(LEFT);
        fill(255);

        textAlign(CENTER);
        text(preg[actual], 15,50,162,95);
        textAlign(LEFT);
        noFill();
        ellipse(49, 132, 45, 45);
        text("V", 46,135);
        noFill();
        ellipse(140, 132, 45, 45);
        text("F", 137,135);
        noFill();
    }

    void ComprobarSolucion(){
        if(verdadero > 0 && falso > 0)
            { //Estan en los dos recuadros, se toma como fuera
              //Text("No pueden ser las dos cosas", 20,100)
              DOS_SOLUCIONES = 1;
              sol=POS_FUERA;}
            else{
                if(verdadero > 0)
                    {sol = POS_VERDADERO;}
                else{
                    if(falso > 0)
                        {sol = POS_FALSO;}
                    else{
                        sol=POS_FUERA;
                    }
                }
            }
        verdadero = 0; falso = 0;
    }
}
```



```
}

void PintarVerde(){
  fill(22,108,22);
  ellipse(95,85,diametro,diametro);
  fill(255);
  textAlign(CENTER);
  text("CORRECTO", 90,80);
  text("Sal del cuadrado, por favor", 90,100);
  textAlign(LEFT);
  if(diametro<400){
    diametro+=5;}
}

void PintarRojo(){
  fill(165,22,22);
  ellipse(95,85,diametro,diametro);
  fill(255);
  textAlign(CENTER);
  text("INCORRECTA", 90,80);
  text("Sal del cuadrado, por favor", 90,100);
  textAlign(LEFT);
  if(diametro<400){
    diametro+=5;}
}

void Trakear(){
  //Control de posiciones
  tracking.getBlobs(function(blobs) {
    // console.log(blobs);

    for(int i = 0; i < blobs.length; i++){
      x[i] = blobs[i].x;
      y[i] = blobs[i].y;

      fill(0,100,100);
      ellipse(x[i], y[i], 5, 5);

      if (( x[i]> 120) && ( x[i] < 160) && ( y[i] > 110) && ( y[i] < 160))
      {falso++; //Está en falso
      }
      else if ((x[i] > 30) && (x[i] < 70) && (y[i] > 110) && (y[i] < 150))
      {verdadero++;//Está en Verdadero
      }
    }
  }
}
```

```
    }  
  }); //end tracking  
}
```



