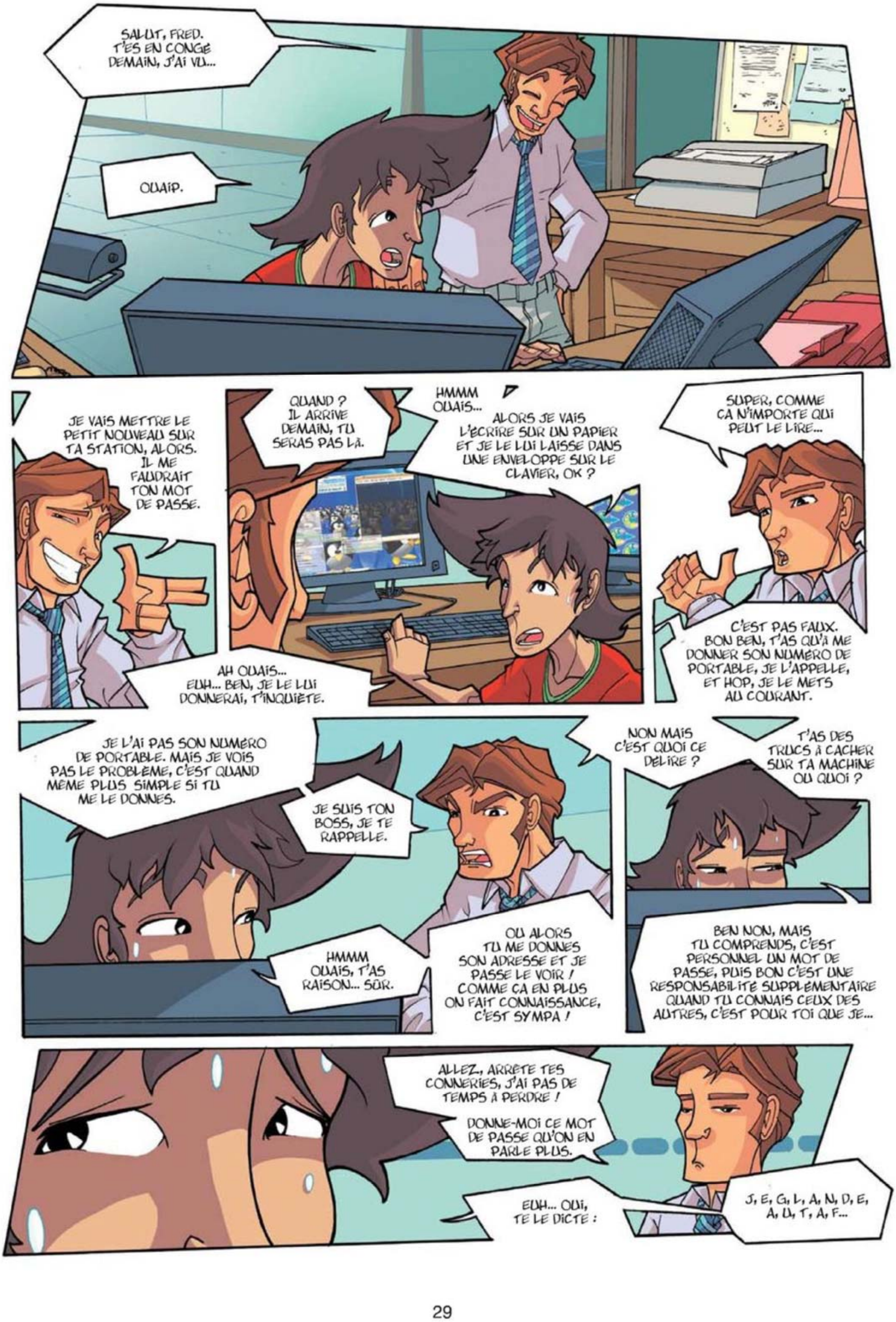
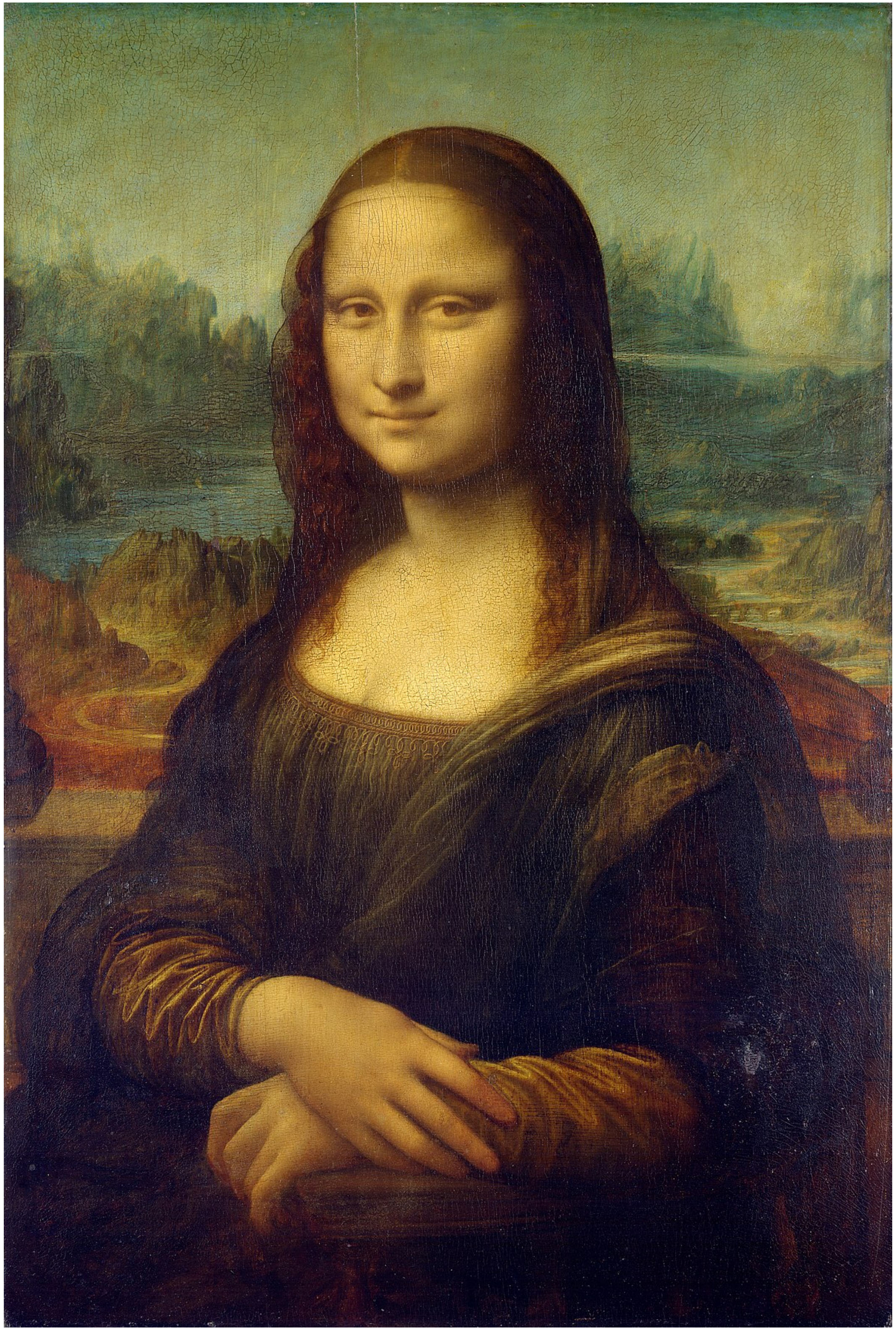

SALUT, FRED. T'ES EN CONGÉ DEMAIN, J'AI VU...
OUAIP.
JE VAIS METTRE LE PETIT NOUVEAU SUR TA STATION, ALORS. IL ME FAUDRAIT TON MOT DE PASSE.
AH OUAIS... EUH... BEN, JE LE LUI DONNERAI, T'INQUIÈTE.
QUAND ? IL ARRIVE DEMAIN, TU SERAS PAS LÀ.
HMMM OUAIS...
ALORS JE VAIS L'ÉCRIRE SUR UN PAPIER ET JE LE LUI LAISSE DANS UNE ENVELOPPE SUR LE CLAVIER, OK ?
SUPER, COMME ÇA N'IMPORTE QUI PEUT LE LIRE...
C'EST PAS FAUX. BON BEN, T'AS QU'À ME DONNER SON NUMÉRO DE PORTABLE, JE L'APPELLE, ET HOP, JE LE METS AU COURANT.
JE L'AI PAS SON NUMÉRO DE PORTABLE. MAIS JE VOIS PAS LE PROBLÈME, C'EST QUAND MÊME PLUS SIMPLE SI TU ME LE DONNES.
JE SUIS TON BOSS, JE TE RAPPELLE.
HMMM OUAIS, T'AS RAISON... SÛR.
OU ALORS TU ME DONNES SON ADRESSE ET JE PASSE LE VOIR ! COMME ÇA EN PLUS ON FAIT CONNAISSANCE, C'EST SYMPA !
NON MAIS C'EST QUOI CE DÉLIRE ?
T'AS DES TRUCS À CACHER SUR TA MACHINE OU QUOI ?
BEN NON, MAIS TU COMPRENDS, C'EST PERSONNEL UN MOT DE PASSE, PUIS BON C'EST UNE RESPONSABILITÉ SUPPLÉMENTAIRE QUAND TU CONNAIS CEUX DES AUTRES, C'EST POUR TOI QUE JE...
ALLEZ, ARRÊTE TES CONNERIES, J'AI PAS DE TEMPS À PERDRE !
DONNE-MOI CE MOT DE PASSE QU'ON EN PARLE PLUS.
EUH... OUI, TE LE DICTE :
J, E, G, L, A, N, D, E, A, U, T, A, F...


ROBERT
DOWNEY JR.
TERRENCE
HOWARD
JEFF
BRIDGES
AND GWYNETH
PALTROW
IRON MAN
MARVEL
5.2.08
IronManMovie.com




CLAC CLAC
DEUX CRÂNES SUR LE CÔTÉ... J'AI ENCORE GAGNÉ.
HÉ HÉ! HÉ!
PFF...
CINQ JOURS SONT PASSÉS DEPUIS...
BON, MOI, JE VAIS Y ALLER.
ZAT!
HEIN ? OÙ ÇA, RYÛK ?
PARCE QU'ICI, DANS LE MONDE DES MORTS, IL N'Y A PAS GRAND-CHOSE À VOIR. HÉ ! HÉ !
J'AI PERDU MON DEATH NOTE.
QUOI ? HA ! HA ! HA ! TOI, VRAIMENT, TU N'EN RATES PAS UNE !
MAIS AU FAIT, TU N'AVAIS PAS MENTI À NOTRE ROI POUR QU'IL T'EN DONNE UN DEUXIÈME ? TU N'AS QUAND MÊME PAS PERDU LES DEUX ?

collections of self-propelled physical objects (e.g., mini robots) that can move collectively and react to user input. Swarm user interfaces can be seen as a coarse-grained version of Sutherland's and Ishii's futuristic visions of user interfaces based on programmable matter.

Due to zooids' ability to freely and quickly reconfigure themselves spatially, a collection of zooids can act as a display and can provide meaningful user output. Due to their ability to sense user actions, zooids can also support rich input. For example, users can either move zooids one by one, or manipulate many zooids at once using "sweeping" gestures [35]. Sophisticated interactive behaviors can be implemented on the application side, e.g., zooids can act as controls or as handles for manipulating others zooids; they can even move other light objects. At the same time, since all input *and* output can be mediated through the same physical elements, the system is able to achieve a complete fusion between input and output and provide a full experience of physical manipulation. Finally, the system is relatively lightweight and only requires the use of a compact DLP projector ($122\,mm \times 115\,mm \times 48\,mm$) for optical tracking. Zooids can operate on any horizontal surface (e.g., a sheet of paper, a messy office desk, a dining table, or a game board), making it possible to blend swarm user interfaces with everyday physical environments. To stimulate future research on swarm user interfaces, we distribute our Zooids tabletop swarm user interface platform in open-source and open-hardware.

In summary, our contributions are:

- A working definition for swarm user interfaces with several implemented examples,
- The first open-source hardware/software platform for experimenting with tabletop swarm user interfaces,
- A set of scenarios to illustrate the unprecedented possibilities offered by our system and by tabletop swarm user interfaces in general,
- A discussion of some general design principles and design challenges for swarm user interfaces.

Furthermore, as benefits, Zooids:

- can coexist in large numbers, in comparison to previous actuated tangible user interfaces,
- can act as individual objects, while being small enough to also act as *"pixels"* of a physical display,
- can be manipulated either individually or collectively, including with physical tools such as rulers,
- are lightweight, can operate on any horizontal surface, and relatively cost-effective: about 50 USD each now, down to $20 if mass manufactured.

BACKGROUND

Our work is related to several research areas, namely: tabletop tangible user interfaces, shape displays, swarm robotics and data physicalization.

Tabletop Tangible User Interfaces

Although tangible user interfaces (TUIs) come in many different forms (including modular assemblies of sensors and actuators [19, 40]), *tabletop TUIs* are particularly common. Tabletop TUIs allow users to interact with digital information by moving physical objects (tangibles) on flat table surfaces [72, 73]. These systems have been used for a range of applications such as systems engineering control [51], musical composition [52], urban planning [74], and education [23].

One limitation with classical tabletop TUIs is the one-way mapping between digital and physical objects — if the former change, the latter can become inconsistent [26]. A number of technologies have been proposed to actuate tangibles, including 2D gantries [6, 38], arrays of electromagnets [48, 50, 78, 76], arrays of ultrasonic transducers [42], electro-static attraction [80, 4], vibration [57, 81] and mobile robots [60, 30, 58, 47, 43, 53, 49]. These systems also support position tracking through a variety of means such as optical tracking with cameras of LEDs or markers (including those using an optical multitouch table), or projector-based tracking. The tangibles range in size from coin-sized [48] to 10 cm [53].

A variety of interaction techniques have been explored on actuated tabletop TUIs, primarily based on the direct manipulation of a single tangible per hand [48], or of small groups of tangibles through multitouch input [53]. Patten [50] explored the use of passive tools in conjunction with actuated tangibles for specifying computational constraints. Other researchers have added dimensions such as vertical displacement to actuated tangibles [43]. These active tangibles can provide haptic feedback while interacting, by applying force directly to the user's hand as they translate along the surface [48, 41]. Actuated TUIs also provide opportunities for remote collaboration, as remote objects can be kept in sync [6, 58].

The design space of tabletop TUIs is vast, and a lot has been explored. However, previous systems have not considered interaction with many (e.g., 10, 30 or more) small actuated tangibles, which we show opens up possibilities for novel interactions and applications. Also, in many previous systems [48, 50, 60, 58, 47, 43, 53, 49, 81], tangibles are used in conjunction with a graphical display, so the tangibles primarily act as handles for digital information. We are interested in user interfaces where the tangibles are used not only as controllers, but also as representations of digital content.

Shape Displays and Programmable Matter

Shape displays are user interfaces involving physical surfaces or volumes that can sense user input and whose geometry can be computer-controlled [56, 61]. Many such systems support discretized shape control of 2.5D surfaces using arrays of motorized bars [54, 39, 15], while other systems support continuous shape control using pneumatic or hydraulic actuation [14, 82] or shape-memory alloys [61]. Currently, many of these systems require a heavy equipment and only allow limited control over physical geometry and topology. In particular, none of the previous systems can emulate separate, detached physical objects.

These research efforts are partly inspired by visions such as Sutherland's and Ishii's discussed before, where computers would be able to reconfigure physical matter to recreate any physical shape. Other fields such as robotics and material science have been interested in realizing this dream of "pro-

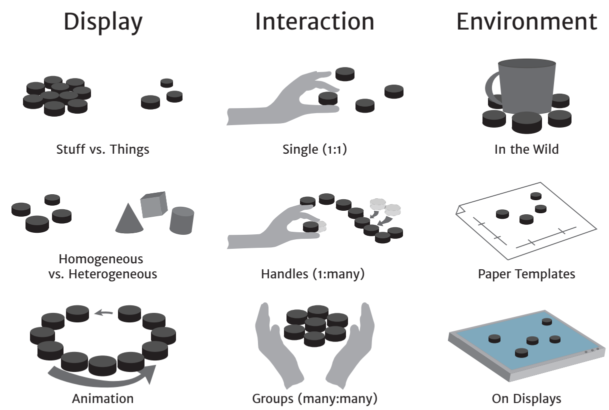


Figure 8. Design Space explored with Zooids.

SWARM UIS: DESIGN PRINCIPLES AND CHALLENGES

Swarm UIs radically change the way we think of user interfaces, not only from an end user's perspective but also from an application designer's perspective. We discuss new concepts, and highlight the major differences here.

Figure 8 gives an overview of the design space of Swarm Interfaces. They can be organized into an *interaction* aspect (interacting with one zooid, controlling many with one zooid, or with groups), a *display* aspect, and an *environment* aspect (operating in a neutral area, in a crowded desk populated with external objects, over a static background layer, or over a dynamic display). We expand on some of these aspects below.

Display: Things vs. Stuff

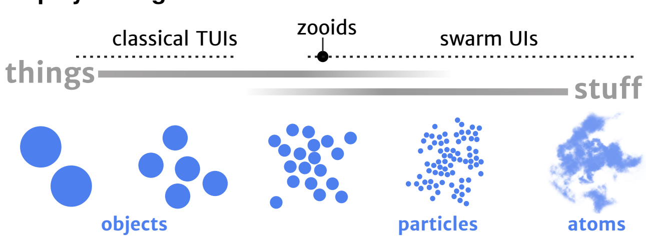


Figure 9. The continuum between "things" and "stuff".

Designing swarm UIs requires thinking both in terms of "things" and of "stuff". In our previous examples, a zooid can stand for an individual object (e.g., a widget) or be part of a larger collection of objects (e.g., a circle). Figure 9 illustrates the continuum between these two paradigms: *things* are physical entities experienced as individual, solid objects; *Stuff* consist in physical entities experienced as shapes and material that can be reshaped, divided, merged, or temporarily solidified to emulate things. The elements making up stuff can be large enough to be visible (particles) or too small to be visible (atoms). Typical TUIs are located to the left of the continuum — they are made of things. In contrast, Swarm UIs occupy the right half of the continuum. As a low-resolution swarm UI implementation, zooids stand in the gray area of the continuum and have both the affordance of things and stuff.

The thing-stuff continuum also applies to traditional graphical displays. Many computer displays from the 80's were very low

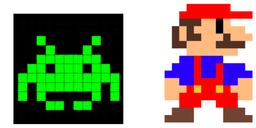


Figure 10. Alien from the game Space Invaders from Taito (1978) and main character from the game Mario Bros by Nintendo (1983).

resolution (semi-graphics from the Sinclair ZX-81 and Tandy TRS-80 were 64×48 pixels), thus pixels were discernible particles much like the zooids in our previous examples (see Figure 10). Now with ultra-high resolution displays pixels became practically invisible, i.e., they became atoms. There are however major conceptual differences between pixel-based displays and swarm UIs, which we discuss next.

Display: Fixed vs. Movable Elements

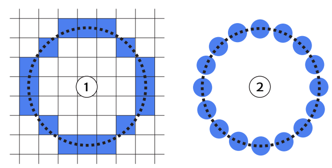


Figure 11. A circle obtained by assembling 16 elements using (1) Bresenham's algorithm and (2) free object positioning.

We are used to program graphics on computer displays where the elements (pixels) are arranged on a regular grid, and only their color is controlled. Although elements of swarm UIs can also have different colors (in our system, each zooid embeds a color LED), a major difference is that they can be positioned freely. Even at equal resolution between the two systems, the way elements can be combined into shapes is very different (see Figure 11). In general, free positioning allows finer shape control than simply turning pixels on and off. At the same time, this extra flexibility comes at the cost of slower response time and higher engineering complexity, with algorithmic problems such as collision avoidance and optimal element-target assignment. In addition, with systems with few elements such as zooids, designers need to think carefully about how to use every zooid optimally, the same way designers from the 80's had to think carefully about how to best use every pixel. It will become less of a concern as the resolution of swarm UIs increases, but on the other hand, engineering and algorithmic challenges will likely become harder. In addition, as shown in Figure 8, the display elements may be homogeneous, as with zooids, or heterogeneous.

Display: Fixed vs. Variable Numbers of Elements

On regular graphical displays the total number of pixels is generally fixed, and the illusion of having more or less content on the screen is achieved by simply manipulating pixel color (e.g., having more or less dark pixels on a white background). In contrast, many swarm applications (e.g, our drawing application) require the number of elements to actually change over time. Zooids cannot be created or destroyed, but as we saw, unassigned zooids can be placed in a dedicated region and moved to the working area whenever they are needed. This type of object persistence contributes to realism and can help