

Technologie de reconnaissance d'objet sur écran capacitif.

volumique



2016

---

## introduction

---

*Volumique* a développé une technologie de reconnaissance d'objet posé sur écran capacitif.

Ce document résume d'une manière didactique les différents aspects de cette technologie et montre aussi de nombreux usages possibles à développer par la suite.

Ce manuel s'adresse à toutes les personnes (développeurs, designers, Game Designers, Makers...) s'intéressant au développement de projets dans le domaine des objets connectés avec un minimum de connaissance technique. Nous allons détailler la manière de développer rapidement un projet utilisant cette technologie de reconnaissance d'objet. Nous étudierons les aspects techniques de fabrications, informatiques et ergonomiques, illustrés par de nombreux exemples.

Ce document est le résultat de plus de quatre ans de recherche et de travail dans ce domaine.

---

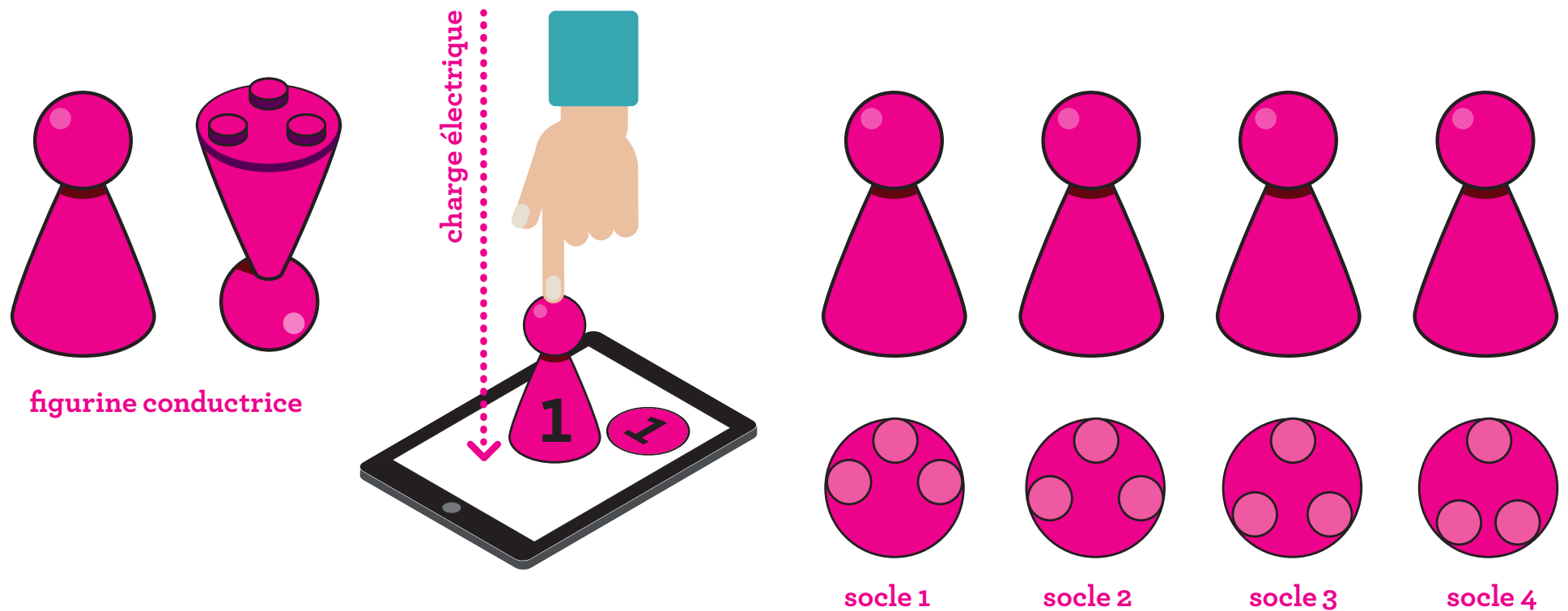
# introduction

---

---

 la technologie volumique
 

---



Cette technologie permet de reconnaître simplement un objet posé sur un écran de type capacitif. Les objets (cartes, figurines, socles, impressions 3D...) reconnus n'ont aucune électronique ou batterie embarquées. Sont reconnus en temps réel : l'objet, sa position ainsi que sa rotation sur l'écran.

La détection se fait par contact de la main avec

l'objet posé sur l'écran. La charge électrique de la main passe à travers l'objet et va donc « faire croire » que des doigts touchent l'écran.

Les objets possèdent sous leurs bases des points reconnus comme des « touches » par l'écran.

Le système reconnaît donc une configuration de « touches » et va en déduire une reconnaissance de l'objet ainsi que sa position et sa rota-

tion sur l'écran. Chaque objet ayant une configuration au préalable définie (tout comme un code-barres). Les objets reconnus utilisent une encre ou un matériau conducteur permettant de transmettre la charge électrique de la main vers l'écran.

## La technologie RFID versus Volumique

technologie	RFID/NFC	Volumique
nombre d'objet détecté	presque infini	dépendant de la surface de l'objet
batterie	non	non
necessite un lecteur externe	oui	non
necessite de l'électronique dans l'objet	oui	non
détection de la position	non	oui
détection de la rotation	non	oui

Ces jeux hybrides utilisent presque tous une technologie RFID, demandant d'installer dans la figurine un peu d'électronique et nécessitant aussi un lecteur de RFID spécifique permettant de reconnaître l'objet. *Skylanders* et *Disney infinity* utilisent un socle spécifique devant se connecter à la console ou tablette de jeu. De plus la technologie RFID ne détecte uniquement la présence d'une puce (et de son objet

associé) mais en aucune manière sa position ni sa rotation.

La technologie Volumique au contraire ne nécessite aucun lecteur spécifique, ni d'électronique embarquée, et en plus permet de reconnaître avec précision la position et la rotation de l'objet sur l'écran en temps réel. Il suffit de poser l'objet sur l'écran pour reconnaître l'objet.

---

**prototyper**

---

---

## prototyper et tester

---

### **Une technologie très facile à tester et à prototyper.**

Techniquement il ne faut qu'un écran capacitif (un téléphone ou une tablette) et un matériau conducteur (plastique conducteur, aluminium, encre...). Le SDK/plugin fonctionne avec la grande majorité des outils à disposition, comme Unity 3D, et des langages comme le C# et Objective C. Cela permet donc de tester une idée très vite afin de la valider dans un processus de développement plus général.

En quelques heures, vous serez capable de tester par vous-même ce nouveau type d'interaction entre objets et application numérique.

### **Technologie brevetée.**

Cette technologie est brevetée par les éditions volumiques sous les numéros WO2013107957A1, WO2012168574A1.

Cela veut donc dire qu'en cas de développement d'un produit commercial, vous êtes protégés par nos brevets. Un contrat sera donc effectué entre votre société et les éditions volumiques concernant l'usage de cette technologie dans un domaine commercial.

Les détails seront fournis dans le contrat de licence en annexes.

## validation industrielle

Différents produits utilisant cette technologie sont déjà sortis commercialement. Cette technologie est donc validée industriellement par de grands acteurs du marché, comme Spin Master (<http://www.appmatestoys.com/>) et Hasbro. Les aspects concernant les normes en vigueur et la sécurité sont aussi validés. La maîtrise des coûts de fabrication est aussi évaluée très précisément, et nous pouvons conseiller des fabricants capables d'utiliser cette technologie dans un processus industriel.

Voici une liste de jeux sortis dans le commerce utilisant cette technologie.

### **SpellShot Zapped | Hasbro**

<http://spellshot.net/>



### **Monopoly Zapped | Hasbro**

<http://www.hasbro.com/en-us/product/monopoly-zapped-edition:C8209658-5056-900B-10FA-9A3DC86C8F76>



### **BattelShip Zapped | Hasbro**

<http://www.hasbro.com/en-us/product/battelship-movie-edition-zapped-game:770717DC-5056-900B-10C4-D6E0450759C6>



### **City ville Skyline | Zynga**

<http://www.amazon.com/Cityville-Skyline-Game-Zapped-Edition/dp/B008KGR1YY>



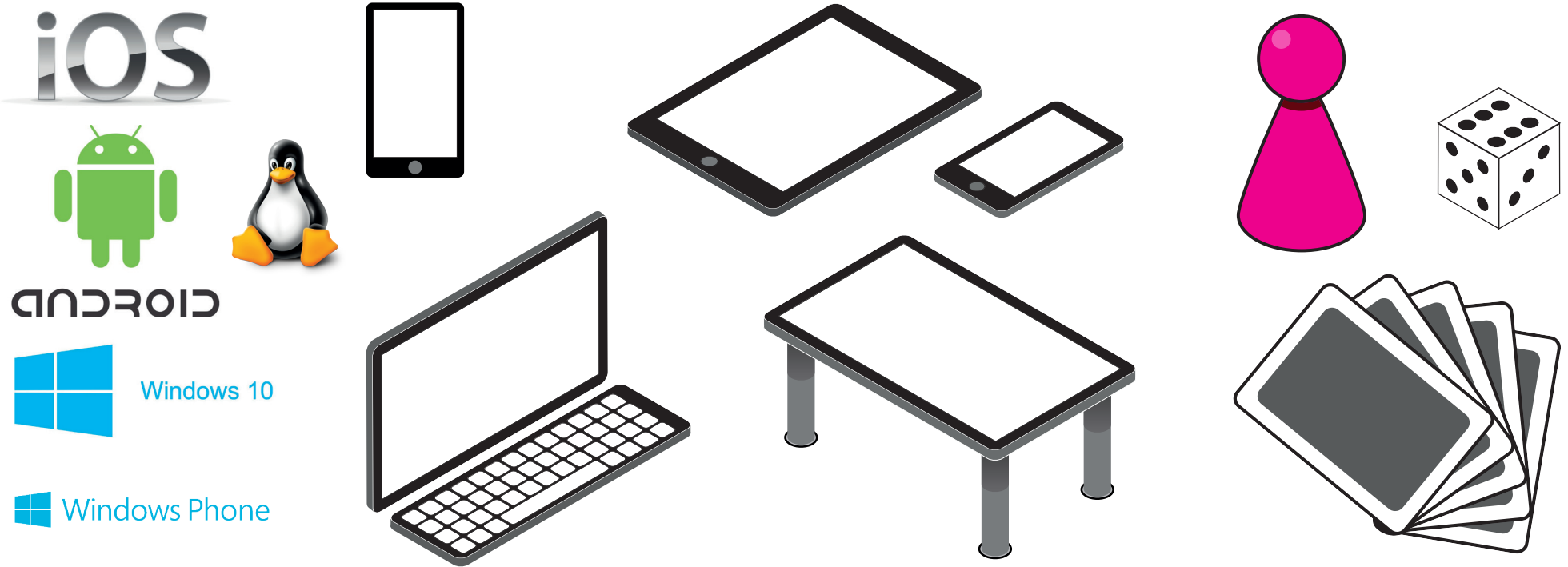
### **Smart Letters , Smart Numbers | Marbotic**

<http://www.marbotic.fr/>





## Description de la technologie

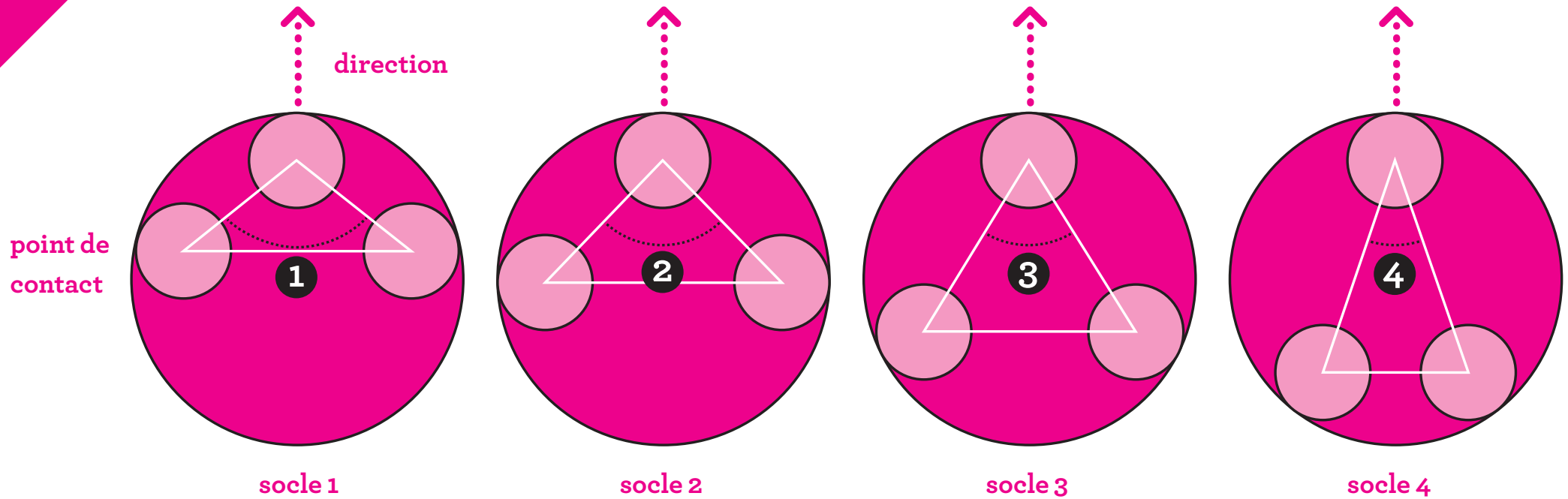


La détection est basée sur la reconnaissance d'un motif positionné sous l'objet en contact avec un écran capacitif.

**Les différents Operating System supportés sont :** iOS, Android, Windows et Linux.

**Les différents types d'écrans :** téléphone / tablette / table interactive / Tablette PC...

les écrans doivent utiliser un système capacitif. Les différents types d'objets peuvent être des socles, des figurines, des cartes à jouer... avec comme seule contrainte d'avoir à un moment donné ses points de reconnaissances en contact avec l'écran. La surface de contact de ces objets doit donc être plane.



Dans le jeu *SpellShot* créé et développé par Volumique, nous avons utilisé des motifs composés de 3 points répartis autour de la base des socles. Les 4 figurines sont reconnues par la forme du triangle composée par ces 3 points. En utilisant 3 points, nous pouvons déduire la position de l'objet (au barycentre des 3 points), l'angle de

rotation (entre deux points du triangle) et son axe de visée (le troisième point étant toujours considéré comme l'avant de la figurine). La reconnaissance de la figurine se fait par l'angle d'un des sommets du triangle reconnu par le système. Chaque figurine possédant sa propre forme de triangle.

Il faut noter que l'algorithme de détection actuel est différent, mais utilise toujours 3 points de contact.



*SpellShot*

---

**SDK**

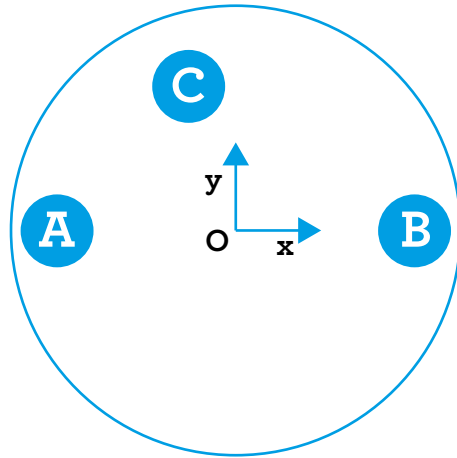
---

---

## Description du SDK / Plugin

---

### Volumique Pattern Detection Patterns



There are some constraints on the definition of triplet  $\{A,B,C\}$  to ensure compatibility with the detection algorithm. Patterns definition are provided by Volumique according to client needs.

#### Detection Library

All library calls are made through a singleton named `VPatternDetector`.

`VPatternDetector` must be configured by registering predefined patterns.

Each pattern is registered by providing the following values:

- **distance** : distance between points A and B
- **position** : position of point C relatively to center of [AB]
- **userData** : value specific to the user which is used as an ID in order to differentiate registered patterns
- **position offset** : if necessary an offset position can be added to detected position when point O does not correspond to the center of the object/card holding the pattern
- **angle offset** : if necessary an offset can be applied to the detected orientation when the object is not aligned with the axis defined by the figure. In order to detect objects/cards placed on screen, user must provide active touches to `VPatternDetector`. This can be done every frame, or only when touch events occur.

After processing active touches, `VPatternDetector` returns detected patterns with the following values:

- **userData** : value provided when the pattern was registered
  - **position** : computed position of the pattern. This corresponds to position of point O with an additional offset if it was requested during pattern registration
  - **direction** : computed direction of the pattern. This corresponds to the vector  $y$  in the previous picture (offseted if necessary).
- Remark: `VPatternDetector` is tolerant to loss of touches. This means that in order to be detected, all 3 points of a pattern must be first detected on screen. But once a pattern is detected, it will remain detected as long as one of the 3 initial points remains on screen. This allows to preserve detection even if the object holding the points is not laid perfectly flat on screen all the time.*

---

## Description du SDK / Plugin

---

### Sample Projects

#### Xcode: [iOS]

*ios/VolumiquePatternDetectorSample* is an Xcode Project that demonstrates the use of the library.

- The library is located in lib folder.

It contains:

- *libVolumiquePatternDetectorLic.a*: the file to link with
- *VPatternDetector.h*: the header file of VPatternDetector Objective-C interface (with documented functions)
- *VPatternDetector* is configured in ViewController viewDidLoad method.
- *VPatternDetector detectPatterns* is called on touch events.
- Detected patterns and unused touches are displayed in MyView drawRect method. Some elements are displayed shifted on Y axis in order to see them on screen.

#### Unity: [iOS & Android]

*unity/VolumiquePatternDetectorSample* is a Unity project for iOS and Android that demonstrates the use of the library.

- the library is located in Assets/Plugins folder. It contains:

- *Android/libVolumiquePatternDetector.so*: the Android library file
- *iOS/libVolumiquePatternDetectorLic.a*: the iOS library file
- *VPatternDetector.cs*: the c# interface that communicates with the library
- *VPatternDetectorDebugger.cs* is a behaviour that :
  - registers patterns in its Start method.
  - detects and show patterns in its OnGUI method (as in the Xcode project some elements are displayed shifted).

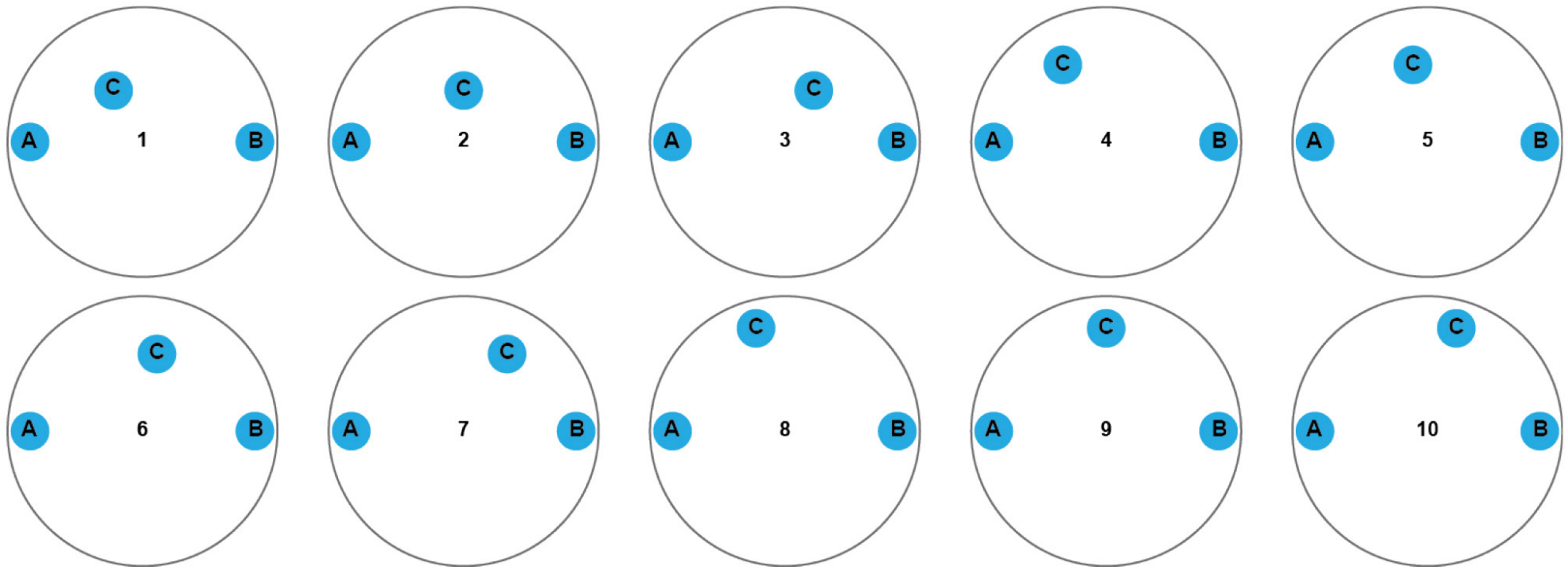
#### License

VPatternDetector uses a license system. The license must be set before any other call of the library.

The license included in the sample projects is compatible with the following 10 patterns:

```
pattern1: distance = 35  position = {-4.5, 8}
pattern2: distance = 35  position = {0, 8}
pattern3: distance = 35  position = {4.5, 8}
pattern4: distance = 35  position = {-6.75, 12}
pattern5: distance = 35  position = {-2.25, 12}
pattern6: distance = 35  position = {2.25, 12}
pattern7: distance = 35  position = {6.75, 12}
pattern8: distance = 35  position = {-4.5, 16}
pattern9: distance = 35  position = {0, 16}
pattern10: distance = 35  position = {4.5, 16}
```

## Description du SDK / Plugin



---

**fabriquer**

---



---

## Fabriquer et tester rapidement

---



### Aluminium

L'aluminium est un matériau conducteur très facile à utiliser. Vous pouvez découper et coller vos motifs en feuille d'aluminium.

Cela marche très bien sur des cartes à jouer par exemple ou tout autre objet.



### Encre conductrice

l'encre conductrice que l'on peut trouver chez Bare Conductive (<http://www.bareconductive.com/>) fonctionne très bien. Vous pouvez la diluer dans de l'eau et l'appliquer sur votre objet. Vous pouvez repeindre par dessus avec de l'encre classique votre objet sans que cela ne perturbe le système de reconnaissance. Il existe aussi des encres argentées pouvant conduire l'électricité.



### Impression 3D

A cette adresse : <http://www....>

vous pouvez trouver des fichiers 3D vous permettant de tester vous-même la reconnaissance.

Ces fichiers sont des socles ayant sous leur base 3 points permettant de créer un motif unique reconnu par notre SDK.

Une fois l'impression 3D faite, il vous suffit de peindre avec de l'encre conductrice dans les emplacements prévus à cet effet (voir le schéma).

---

## Fabriquer et tester rapidement

---

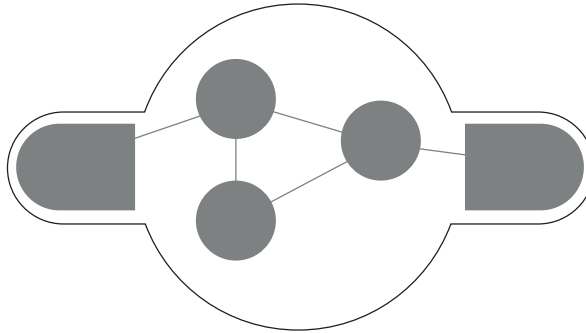


### Socles génériques conducteurs.

Ces socles sont constitués d'un plastique conducteur et possèdent déjà des motifs reconnus par notre SDK.

Il suffit de placer ces socles sous une figurine afin de la rendre détectable.

- la figurine n'est pas conductrice, il faut alors manipuler la figurine en posant ces doigts sur les ergots du socle.
- la figurine est elle même conductrice (encre), la manipulation peut se faire en touchant uniquement la figurine.



### Autocollants conducteurs

Ces auto-collants conducteurs peuvent se coller sous n'importe quelle forme afin de la rendre détectable.



### Fabrication industrielle

Concernant la phase industrielle nous préconisons l'utilisation d'un matériau conducteur, beaucoup plus robuste à long terme que l'encre conductrice.

Il faut utiliser des matériaux ayant une très bonne stabilité dimensionnelle comme l'ABS (acrylonitrile butadiène styrène ). L'ABS peut être facilement rendu conducteur.