

TrivialPursuit [fr]

Créer un plateau type 'Trivial Pursuit',
avec personnalisations.

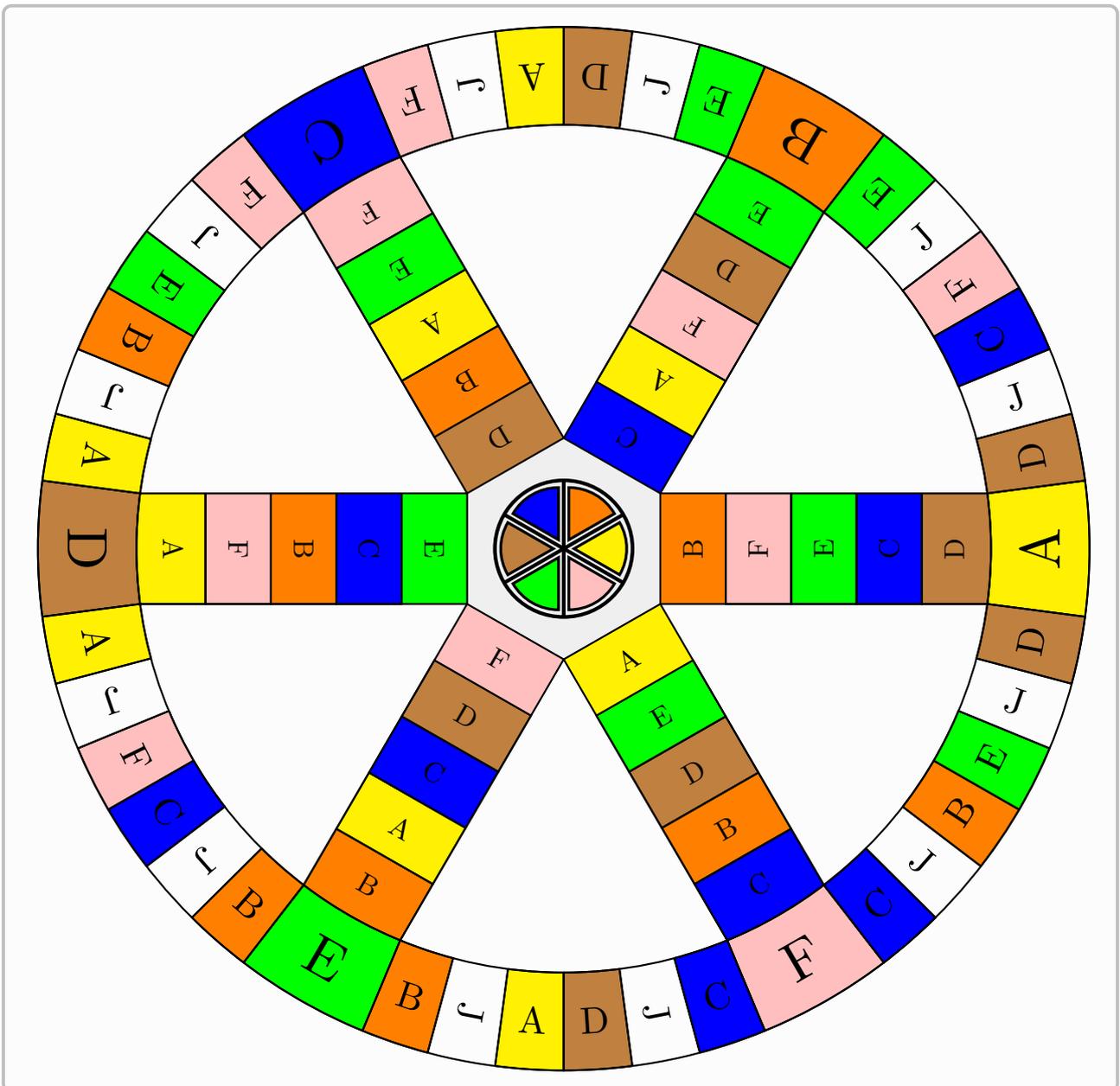
Version 0.1.7 -- 25 août 2025

Cédric Pierquet

c pierquet -- at -- outlook . fr

<https://github.com/cpierquet/latex-packages/tree/main/trivialpursuit>

Trivial Pursuit™, est une marque déposée par Hasbro.



Contents

1	Introduction	2
1.1	Description	2
1.2	Chargement	2
2	Historique	3
3	La commande pour le plateau	4
3.1	Fonctionnement global	4
3.2	Clés et options	5
3.3	Palettes de couleurs prédéfinies, icônes fa prédéfinies	8
4	La commande pour les cartes	10
4.1	Fonctionnement global	10
4.2	Macro(s) de personnalisation	11
4.3	Exemple personnalisé	12

1 Introduction

1.1 Description

L'idée de ce package est de proposer un plateau de jeu de type Trivial Pursuit (marque déposée par Hasbro) avec personnalisations au niveau :

- des dimensions (rayon + hauteur bordure) ;
- des couleurs ;
- des logos (sous forme de caractères, comme `fontawesome`).

L'alternance des cases et des couleurs est reprise du plateau *classique*, donc est fixée globalement.

Pour le moment, les *limitations* sont :

- le nombre de catégories est (pour le moment) fixé à 6 ;
- les logos sont des caractères, comme ceux du package `fontawesome`.

1.2 Chargement

Pour charger le package, il suffit d'ajouter, dans le préambule :

```
\usepackage{TrivialPursuit}           %avec fa5, par défaut
\usepackage[fa6]{TrivialPursuit}     %avec fa6 (test)
\usepackage[fa7]{TrivialPursuit}     %avec fa7 (test)
\usepackage[nonfa]{TrivialPursuit}   %sans fa (test), à charger manuellement
```

Les packages chargés et utilisés par `TrivialPursuit` sont :

- `tikz` avec les librairies `calc`, `positioning` ;
- `calc` et `fontawesome5` ;
- `simplekv` ; `xintexpr` et `listofitems`.

2 Historique

- 0.1.7 : Ajout de cartes
- 0.1.6 : Bugfix + pré-compatibilité avec fa67
- 0.1.5 : Bugfix (compatibilité avec twemoji)
- 0.1.4 : Bugfix + pré-compatibilité avec fa6
- 0.1.2 : Ajout d'une clé [VersionCourte] pour diminuer le nombre de cases
- 0.1.1 : Ajout de palettes de couleurs prédéfinies
- 0.1.0 : Version initiale

3 La commande pour le plateau

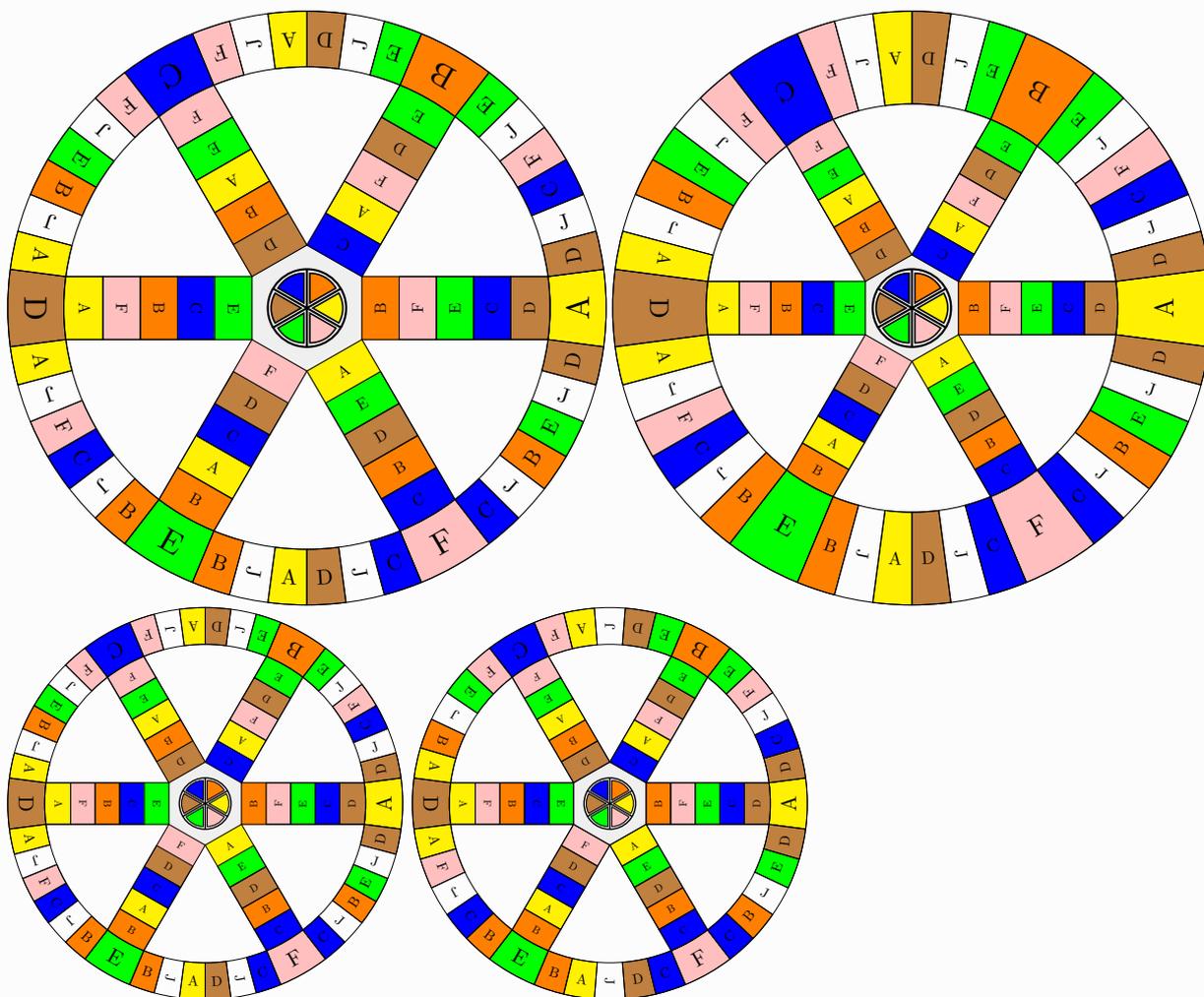
3.1 Fonctionnement global

Le code se charge de tracer le plateau avec les éventuels *ornements*.

Toutes les clés disponibles sont présentées dans le paragraphe suivant, mais concernant les dimensions, des *générales* sont proposées par défaut, mais il est toutefois possible de les modifier :

- soit explicitement par les clés `[Rayon=...]` et `[HauteurCases=...]` ;
- soit en spécifiant une unité particulière grâce à la clé `[Unite=...]`, au quel cas la forme générale du plateau sera conservée.

```
\PlateauTrivialPursuit[Unite=0.5]  
%  
\PlateauTrivialPursuit[Rayon=4,HauteurCases=1.25] %attention au rendu...  
  
\PlateauTrivialPursuit[Unite=0.33]  
%  
\PlateauTrivialPursuit[Unite=0.33,VersionCourte]
```

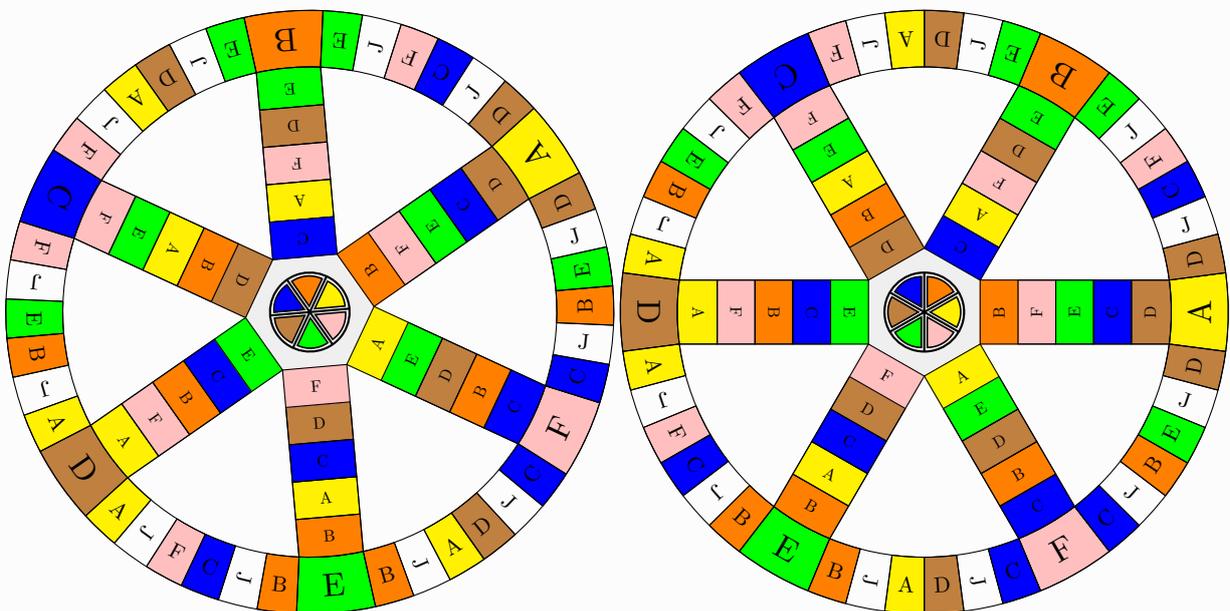


3.2 Clés et options

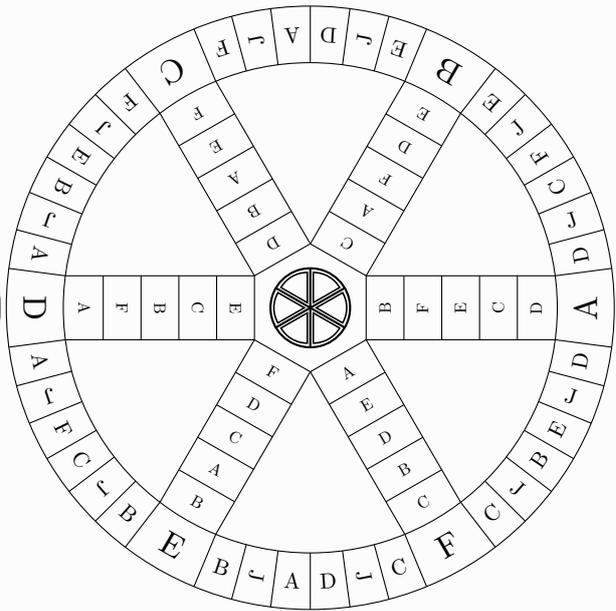
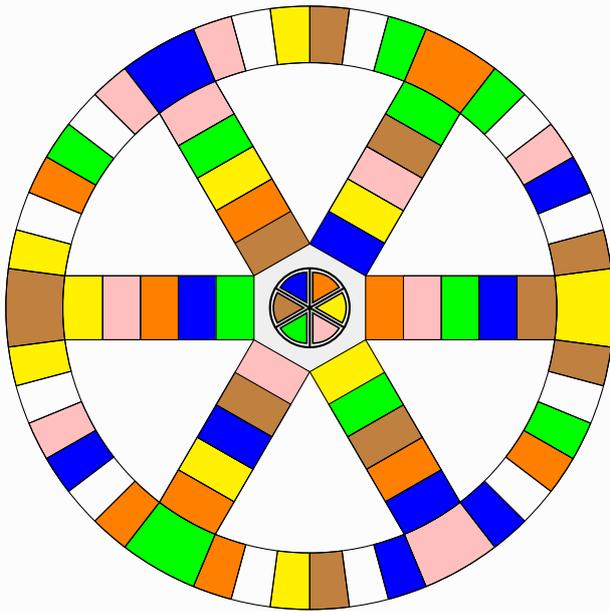
Les clés disponibles sont :

- `Rayon` := rayon du plateau (en unité), et valant `8` par défaut ;
- `HauteurCases` := hauteur des cases extérieures, et valant `1.5` par défaut ;
- `ListeCouleurs` := couleurs des catégories (sens trigo),
et valant `yellow,orange,blue,brown,green,pink` par défaut ;
- `ListeIcones` := icônes des catégories (sens trigo),
`A,B,C,D,E,F` par défaut ;
- `Icones` := booléen pour afficher les icônes, et valant `true` par défaut ;
- `Centre` := booléen pour colorer l'hexagone centrale, et valant `true` par défaut ;
- `CouleurCentre` := couleur du centre hexagonal, et valant `lightgray!25` par défaut ;
- `Couleurs` := booléen pour afficher les couleurs, et valant `true` par défaut ;
- `Logo` := booléen pour afficher un logo central, et valant `true` par défaut ;
- `Jokers` := booléen pour afficher les jokers, et valant `true` par défaut ;
- `IconeJoker` := icône des jokers, et valant `J` par défaut ;
- `Vide` := booléen pour afficher le plateau vide, et valant `false` par défaut ;
- `Unite` := pour préciser l'unité (et gérer un facteur d'échelle), et valant `1` par défaut ;
- `Epaisseur` := épaisseur des tracés, et valant `0.8pt` par défaut ;
- `Rotation` := rotation éventuelle du plateau, et valant `0` par défaut ;
- `VersionCourte` := utiliser 5 cases au lieu de 6 entre les camemberts, et valant `false` par défaut.

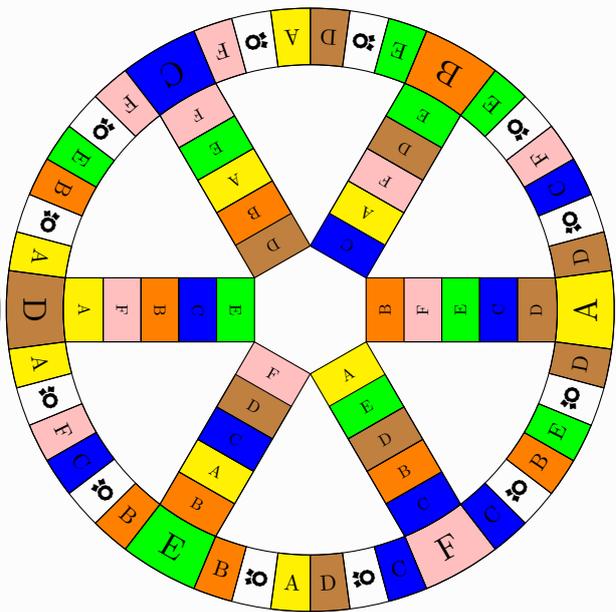
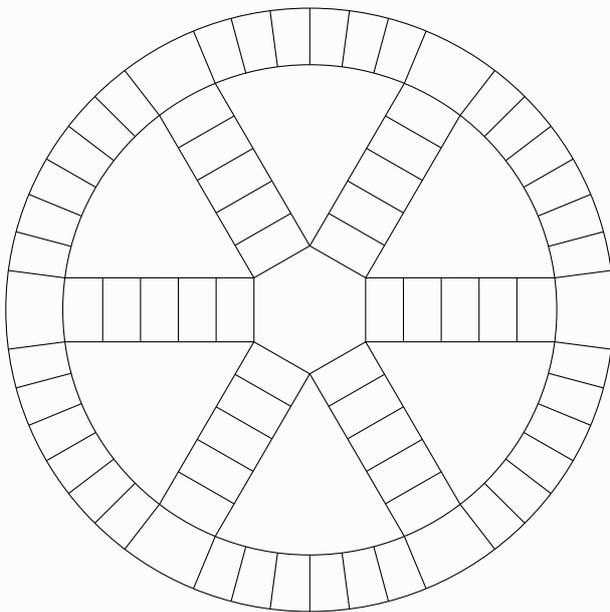
```
\PlateauTrivialPursuit[Rotation=35,Unite=0.5]
\PlateauTrivialPursuit[Unite=0.5]
```



```
\PlateauTrivialPursuit[Unite=0.5,Icones=false]
\PlateauTrivialPursuit[Unite=0.5,Couleurs=false]
```



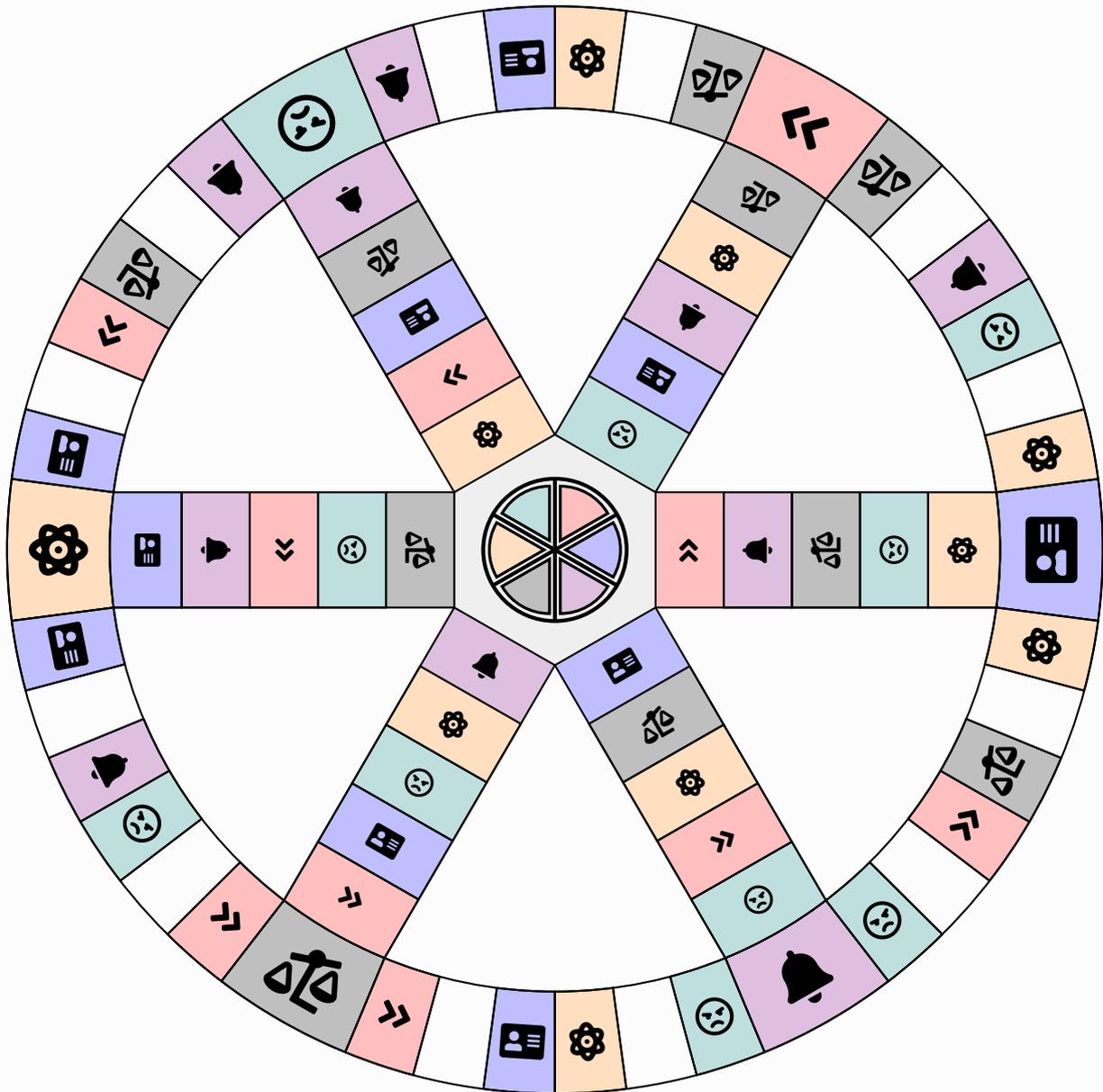
```
\PlateauTrivialPursuit[Unite=0.5,Vide]
\PlateauTrivialPursuit[Unite=0.5,Logo=false,Centre=false,IconeJoker=\faAward]
```



```

\PlateauTrivialPursuit[%
  Jokers=false,%
  ListeCouleurs={%
    blue!25,red!25,teal!25,orange!25,gray!50,violet!25},%
  ListeIcones={%avec fa5
    \faAddressCard,\faAngleDoubleRight,\faAngry[regular],%
    \faAtom,\faBalanceScaleLeft,\faBell}
]

```

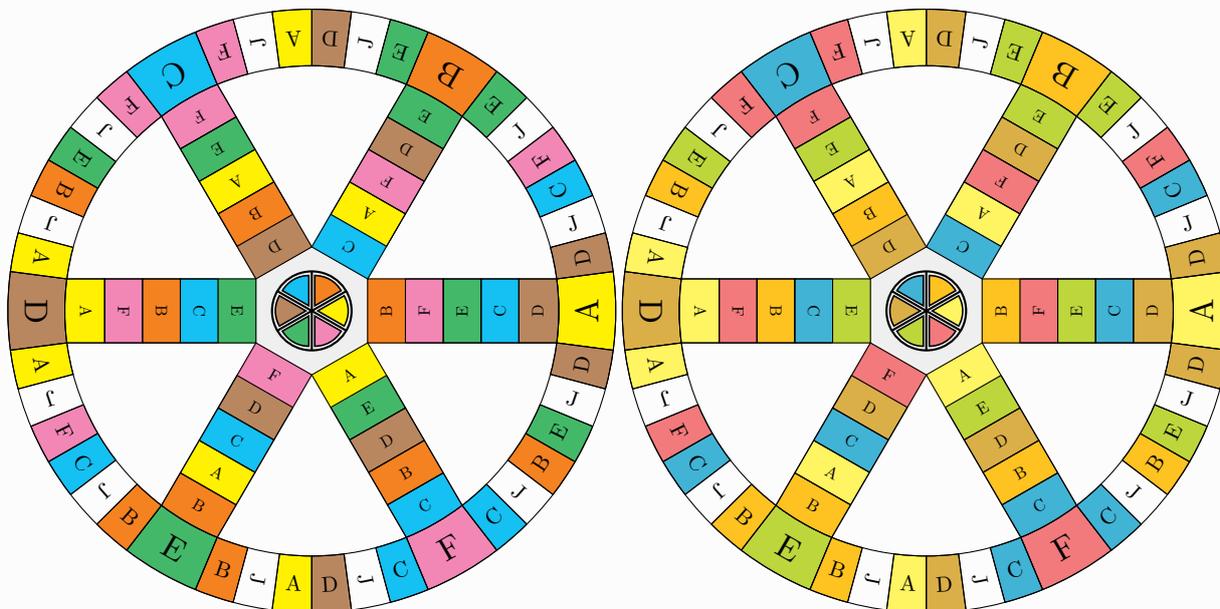


3.3 Palettes de couleurs prédéfinies, icônes fa prédéfinies

En complément (sur une suggestion de quark67), deux styles de couleurs ont été définies en interne :

```
\TPColorsA
\TPColorsB
```

```
\PlateauTrivialPursuit[Unite=0.5,ListeCouleurs=\TPColorsA]
\PlateauTrivialPursuit[Unite=0.5,ListeCouleurs=\TPColorsB]
```



```
\categorialpursuit
```

\PlateauTrivialPursuit%

[ListeIcones=\categorivialpursuit,ListeCouleurs=\TPColorsB,IconeJoker=\faDice]



4 La commande pour les cartes

4.1 Fonctionnement global

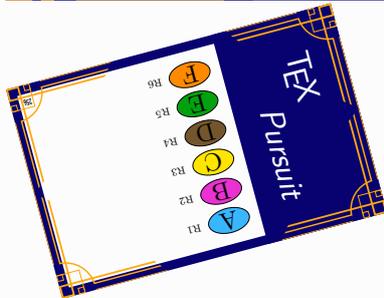
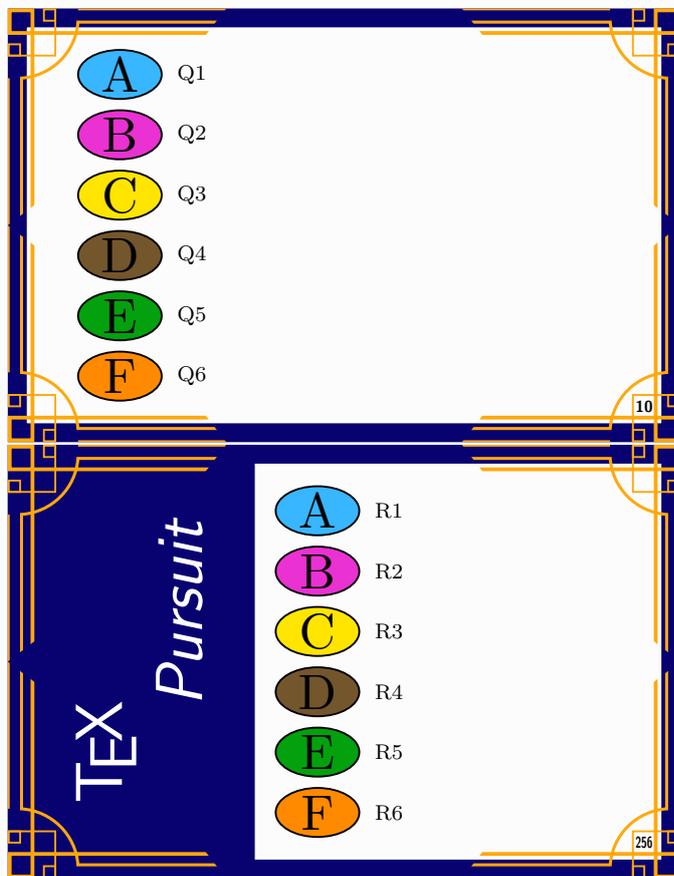
Le code se charge de représenter des cartes, recto ou verso, avec :

- des dimensions *fixées* (mais une échelle globale peut être appliquée) ;
- des couleurs, catégories, questions/réponses personnalisables ;
- des logos éventuels

```
\CardTrivialRecto{Q1}{Q2}{Q3}{Q4}{Q5}{Q6}{10}
```

```
\CardTrivialVerso{R1}{R2}{R3}{R4}{R5}{R6}{256}
```

```
\CardTrivialVerso[scale=-0.5,rotate=15]{R1}{R2}{R3}{R4}{R5}{R6}{256}
```



Les textes (labels + numéro) sont adaptés automatiquement aux éléments graphiques, la taille de la police des Q/R peut-être adaptée le cas échéant.

```
\CardTrivialRecto%
  [options tikzpicture]%
  {Q1}{Q2}{Q3}{Q4}{Q5}{Q6}%
  {numéro de la carte}
```

```
\CardTrivialVerso%
  [options tikzpicture]%
  {R1}{R2}{R3}{R4}{R5}{R6}%
  {numéro de la carte}
```

4.2 Macro(s) de personnalisation

Il existe une macro auxiliaire qui permet de modifier différents paramètres des cartes.

```
\SetParamsCardTrivial{%
  QA font=...,           %police des Q/R
  num font=...,          %police du numéro
  label font=...,       %police des catégories
  categ=...,            %labels des catégories
  border deco color=..., %couleur de bordure
  alt deco color=...,   %couleur secondaire
  color categ A=...,    %couleur catégorie A
  color categ B=...,    %couleur catégorie B
  color categ C=...,    %couleur catégorie C
  color categ D=...,    %couleur catégorie D
  color categ E=...,    %couleur catégorie E
  color categ F=...,    %couleur catégorie F
  alt deco=TF           %booleén pour afficher la bordure
}
```

```
%logo verso
\def\intlogocardversotrivial{%
  \draw[white] (CARD-LOGO) node[rotate=90,text
    width=3.75cm,font=\Huge\sffamily,align=left] {\TeX} \\\hfill{\itshape Pursuit}} ;
}
```

```
%logo recto (fond)
\def\intlogocardrectotrivial{}
```

4.3 Exemple personnalisé

```

%\usepackage{openmoji}

\SetParamsCardTrivial{%
  border deco color=teal,
  alt deco color=orange,
  QA font={\sffamily\fontsize{8pt}{10pt}\selectfont},%
  label font={\bfseries},
  categ={No,H,Ma,Oo,A,G}
}
\def\intlogocardversotrivial{%
  \draw (CARD-LOGO) node[rotate=90]
  {\openmoji[height=2.25cm]{game die}\openmoji[height=2.25cm]{glowing star}} ;
}
\def\intlogocardrectotrivial{%
  \draw (CARD-LOGO) node[opacity=0.05]
  {\openmoji[height=4.25cm]{green apple}} ;
}

```

```

\CardTrivialRecto{Q1}{Q2}{Q3}{Q4}{Q5}{Q6}{12}
\CardTrivialVerso[scale=-1]{R1}{R2}{R3}{R4}{R5}{R6}{12}

```

