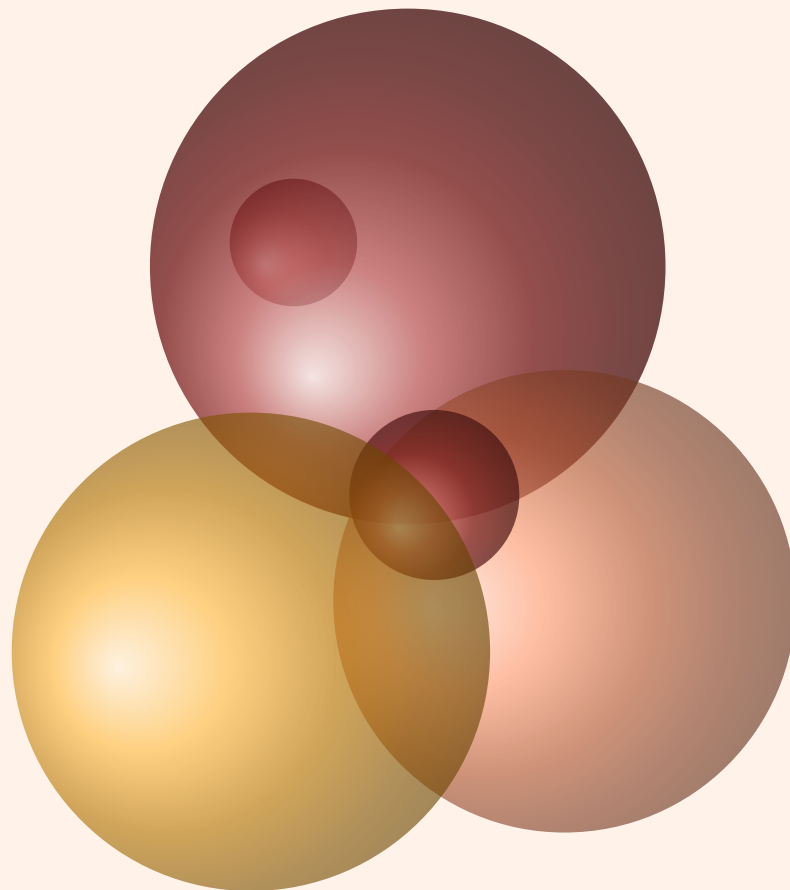


tkz-graph ctan v1.00 c

AlterMundus



Alain Matthes

4 juin 2011

<http://altermundus.fr> <http://altermundus.com>

tkz-graph

AlterMundus

Alain Matthes

*Le package **tkz-graph.sty** est un package pour créer à l'aide de **TikZ** des graphes le plus simplement possible. Il fera partie d'une série de modules ayant comme point commun, la création de dessins utiles dans l'enseignement des mathématiques. La lecture de cette documentation va, je l'espère, vous permettre d'apprécier la simplicité d'utilisation de **TikZ** et vous permettre de commencer à le pratiquer. Il est accompagné du package **tkz-berge.sty** qui permet de tracer des graphes particuliers de la théorie des graphes.*

☞ Je souhaite remercier **Till Tantau** pour avoir créé le merveilleux outil **TikZ**, ainsi que **Michel Bovani** pour **fourier**, dont l'association avec **utopia** est excellente.

☞ Vous trouverez de nombreux exemples sur mes sites : altermundus.com ou altermundus.fr

Vous pouvez envoyer vos remarques, et les rapports sur des erreurs que vous aurez constatées à l'adresse suivante : [Alain Matthes](mailto:Alain.Matthes@univ-lille.fr).

This file can be redistributed and/or modified under the terms of the LATEX Project Public License Distributed from CTAN archives in directory CTAN://macros/latex/base/lpp1.txt.

Table des matières

1	Installation	6
1.1	Avec TeXLive sous OS X, Linux et Windows	6
1.2	Avec MikTeX sous Windows XP	7
2	Premiers graphes avec tkz-graph.sty	8
2.1	Exemple simple avec tkz-graph	8
2.2	Exemple classique avec tkz-graph	9
2.3	Modification du style	9
2.4	La ville de Königsberg avec tkz-graph	10
2.5	La ville de Königsberg avec TikZ mais sans tkz-graph	10
3	Vertex	12
3.1	\Vertex	12
3.1.1	Utilisation de coordonnées cartésiennes	12
3.1.2	Utilisation de coordonnées polaires	13
3.1.3	Option Node : utilisation d'une position référencée	13
3.2	Raccourcis pour placement relatif	14
3.2.1	Utilisation des raccourcis avec les valeurs par défaut	14
3.2.2	Modification de l'unité avec \SetGraphUnit	15
3.2.3	Modification des unités de TikZ : x=2 cm,y=1 cm	15
3.2.4	Exemple classique	15
3.2.5	Autre exemple classique	15
3.2.6	Modication locale de unit avec l'option	16
3.2.7	Modication locale de unit avec l'environnement scope	16
3.2.8	Modication locale de unit avec un groupe \TeX	16
4	Placement de sommets sur une forme géométrique	17
4.1	\Vertices à partir d'un sommet défini par des coordonnées	17
4.2	\Vertices à partir d'une position donnée.	17
4.3	Exemples avec une direction	18
4.4	Placement sur un triangle	18
4.5	Utilisation d'un carré	18
4.6	Utilisation d'un cercle	19
4.7	Utilisation d'un cercle et positionnement des labels	19
4.8	Rotation et labels externes	19
4.9	Placement sur un cercle	19
5	Les labels	21
5.1	Options concernant les labels	21
5.1.1	Option L	21
5.1.2	Option Math	21
5.1.3	Suppression d'un label, Option NoLabel	21
5.1.4	Option LabelOut , Lpos et Ldist	21
5.2	\SetVertexNoLabel	22
5.2.1	Suppression des labels	22
5.3	\SetVertexMath	22
5.4	\SetVertexLabel	22
5.4.1	Labels supprimés puis autorisés.	22
5.4.2	Label en dehors du sommet \SetVertexLabelOut	23
6	Edge avec tkz-graph	24
6.1	Utilisation de \Edge	24
6.2	Arête particulière la boucle : Loop	25
6.2.1	Exemple avec \Loop	25

6.3	Multiple arêtes <code>\Edges</code>	26
6.3.1	Exemple avec <code>\Edges</code>	26
7	Modification des styles des sommets	27
7.1	Modification de <code>vstyle=Art</code>	30
7.2	Modification du style <code>VertexStyle</code> par défaut	31
7.3	Modification d'un style <code>VertexStyle</code>	31
7.4	Autre style <code>\SetVertexSimple</code>	32
7.5	<code>\SetVertexSimple</code> , <code>inner sep</code> et <code>outer sep</code>	32
7.6	Autre style <code>\SetVertexNormal</code>	33
7.7	<code>\SetUpVertex</code>	34
7.8	<code>\SetUpVertex</code> et <code>\tikzset</code>	34
8	Modification des styles des arêtes	35
8.1	Utilisation de l'option <code>style</code> de la macro <code>\Edge</code>	35
8.1.1	Exemple 1	35
8.1.2	Exemple 2	35
8.1.3	Exemple 3	35
8.2	Modification des styles par défaut <code>\SetUpEdge</code>	36
8.2.1	Utilisation de <code>\SetUpEdge</code> Exemple 1	36
8.2.2	Utilisation de <code>\SetUpEdge</code> Exemple 2	37
8.3	Arête avec label <code>LabelStyle</code>	37
8.4	Utiliser un style intermédiaire	37
9	Changement de couleurs dans les styles prédéfinis	39
9.1	<code>\SetGraphShadeColor</code>	39
9.1.1	Exemple	39
9.2	<code>\SetGraphArtColor</code>	40
9.2.1	Exemple	40
9.3	<code>\SetGraphColor</code>	41
9.3.1	Exemple avec <code>\SetGraphColor</code>	41
9.4	Variation I autour des styles	42
9.5	Variation II autour des styles	43
9.6	Variation III autour des styles	44
9.7	Variation IV autour des styles	45
9.8	Variation V autour des styles	46
10	Graphes probabilistes	47
10.1	La macro <code>\grProb</code>	47
10.2	Utilisation de <code>\grProb</code>	47
10.3	<code>\grProb</code> et le style par défaut	48
10.4	<code>\grProb</code> et le style « Simple »	48
10.5	Utilisation d'un style personnalisé	48
10.6	La macro <code>\grProbThree</code>	49
10.6.1	Graphe probabiliste d'ordre 3	49
11	Colorisation Welsh	50
11.1	La macro <code>\AddVertexColor</code>	50
11.2	Exemple d'utilisation	50
	Index	54

Liste des macros dans l'ordre d'apparition :

- `\SetVertexLabelOut`
- `\SetVertexLabelIn`
- `\SetVertexMath`
- `\SetVertexNoMath`
- `\SetUpVertex`
- `\Vertex`
- `\EA`
- `\WE`
- `\NO`
- `\SO`
- `\NOEA`
- `\NOWE`
- `\SOEA`
- `\SOWE`
- `\Vertices`
- `\SetUpEdge`
- `\Edge`
- `\Edges`
- `\Loop`
- `\grProb`
- `\SetGraphShadeColor`
- `\SetGraphArtColor`
- `\SetGraphColor`
- `\AddVertexColor`

Installation

Il est possible que lorsque vous lirez ce document, **tkz-graph** soit présent sur les serveurs du CTAN¹. Si **tkz-graph** ne fait pas encore partie de votre distribution, cette section vous montre comment l'installer.

1.1 Avec TeXLive sous OS X, Linux et Windows

Créer un dossier `tkz` avec comme chemin : `texmf/tex/latex/tkz`.

`texmf` est un dossier personnel, voici les chemins de ce dossier sur mes deux ordinateurs :

– sous OS X `/Users/ego/Library/texmf` ;

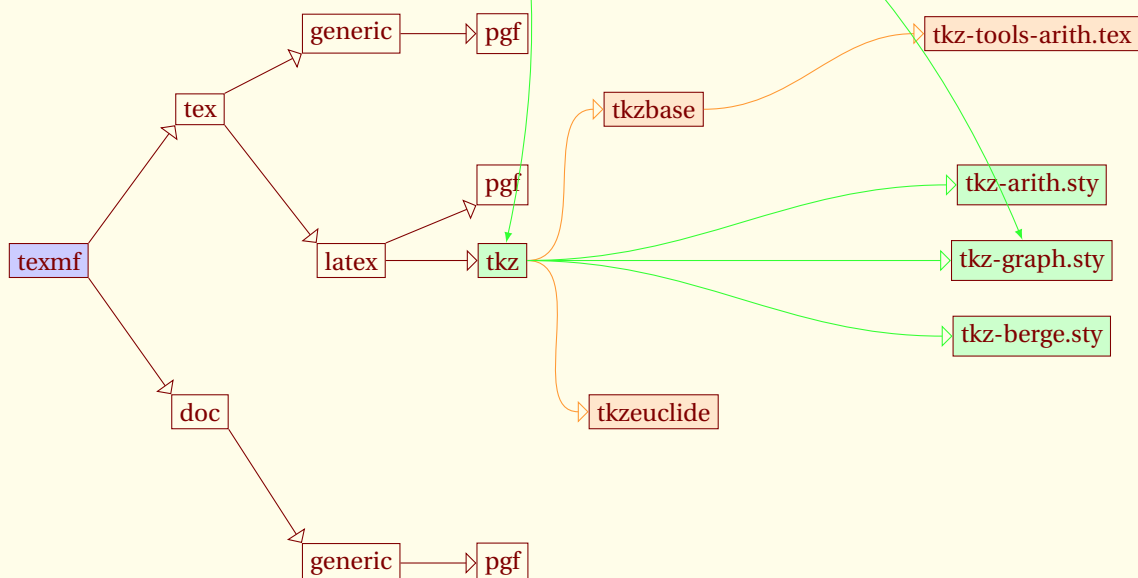
– sous Ubuntu `/home/ego/texmf`.

Sous Windows je ne connais pas cette distribution sous ce système, mais je suppose que l'installation doit ressembler à ce qui se passe sous Linux et OS X.

1. Placez `tkz-graph.sty` dans le dossier `tkz`.
2. Ouvrir un terminal, puis faire `sudo texhash`
3. Vérifier que **xkeyval** version 2.5 minimum et **TikZ 2.1** sont installés car ils sont obligatoires, pour le bon fonctionnement de **tkz-graph**.

Mon dossier `texmf` est structuré ainsi :

*Attention, la présence dans mon dossier `texmf`, des fichiers de **PGF**, s'explique par l'utilisation de la version CVS de **PGF**.*



1. **tkz-graph** ne fait pas encore partie de TeXLive mais il sera bientôt possible de l'installer avec `tlmgr`

1.2 Avec MikTeX sous Windows XP

Il est fort possible que lorsque vous lirez ces lignes, il soit possible d'installer **tkz-graph** automatiquement à l'aide du manager de MikTeX.

Un utilisateur de mes packages **Wolfgang Buechel** a eu la gentillesse de me faire parvenir ce qui suit, et cela permet d'installer manuellement mon package :

Pour ajouter **tkz-graph.sty** à MikTeX² :

- ajouter un dossier **tkz** dans le dossier `[MiKTeX-dir]/tex/latex` ;
- copier **tkz-graph.sty** dans ce dossier ;
- mettre à jour MikTeX, pour cela dans shell DOS lancer la commande `mktexlsr -u` ou bien encore, choisir `Start/Programs/Miktex/Settings/General` puis appuyer sur le bouton `Refresh FNDB` .

2. Essai réalisé avec la version 2.7

Premiers graphes avec tkz-graph.sty

TikZ est un outil que je trouve très agréable à utiliser pour la création de graphes. J'ai trouvé si simple son utilisation que je me suis demandé si cela avait un sens de créer un package pour la création de graphes. Pas de théorie des graphes dans ce package, seulement des outils pour leur construction. Trois arguments peuvent intervenir pour soutenir mon effort :

1. Certains utilisateurs n'ont pas envie d'apprendre quoi que ce soit sur **TikZ** cela est respectable et une simplification du code par l'intermédiaire d'un package peut avoir une certaine utilité. La syntaxe n'est plus tout à fait celle de **TikZ** mais celle de \LaTeX .
2. Il est possible finalement de jouer avec les styles et d'optimiser certaines situations, ainsi la création d'un graphe sans la moindre coordonnée est possible. On peut obtenir des variantes du graphe, simplement en jouant avec les styles.
3. La création de ce que l'on peut appeler les graphes classiques de la théorie des graphes.
4. Et pour terminer, cela peut être une approche en douceur de l'utilisation de **TikZ** par l'intermédiaire des options.

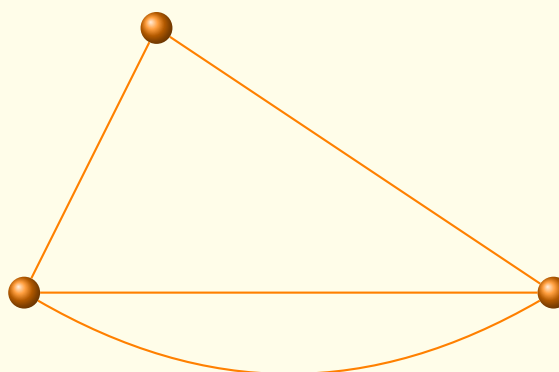
Que peut apporter **tkz-graph.sty** ? Il facilite la gestion des styles des sommets et des arêtes, et également le positionnement de ceux-ci.

2.1 Exemple simple avec tkz-graph

Avant d'expliquer le fonctionnement des différentes macros, il est possible de tester si le package est bien installé avec l'exemple simple suivant. Le code complet est donné. Le préambule peut évidemment être modifié.

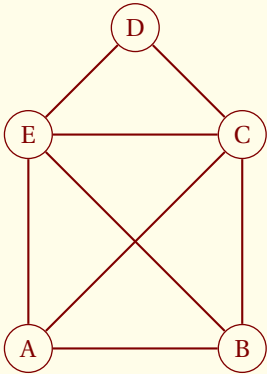
```
% Author : Alain Matthes
% Encoding : UTF8
% Engine : PDFLaTeX
\documentclass[]{\scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[upright]{fourier}
% you can change the line above
\usepackage{tkz-graph}
\thispagestyle{empty}
\begin{document}

\begin{tikzpicture}[scale=1.75]
  \GraphInit[vstyle=Art]
  \Vertex{A}
  \Vertex[x=4,y=0]{B}
  \Vertex[x=1,y=2]{C}
  \Edge[style={bend left}](B)(A)
  \Edges(A,B,C,A)
\end{tikzpicture}
\end{document}
```



2.2 Exemple classique avec tkz-graph

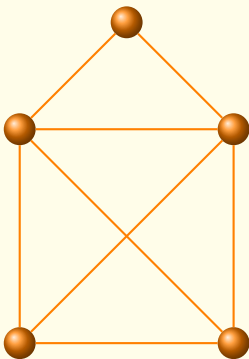
Voyons un exemple classique. Nous allons utiliser un style scolaire **vstyle=Normal** ainsi que les macros **\Vertices**, **\NOEA** et **\Edges** qui permet de créer une "chaîne" d'arêtes (edges). L'environnement **scope** fait partie de **TikZ**, il est utilisé ici afin d'appliquer une rotation.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Normal]
  \SetGraphUnit{2}
  \begin{scope}[rotate=-135]
    \Vertices{circle}{A,B,C,E}
  \end{scope}
  \NOEA[unit=1.414](E){D}
  \Edges(A,B,E,D,C,E,A,C,B)
\end{tikzpicture}
```

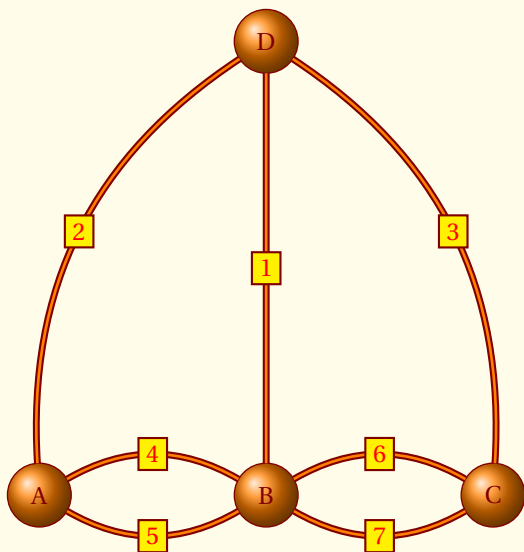
2.3 Modification du style

Un style plus esthétique peut être choisi avec **\GraphInit**. J'ai choisi **Art** parmi une liste que vous découvrirez plus tard.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \begin{scope}[rotate=-135]
    \Vertices[unit=2]{circle}{A,B,C,E}
  \end{scope}
  \NOEA[unit=1.414](E){D}
  \Edges(A,B,E,D,C,E,A,C,B)
\end{tikzpicture}
```

2.4 La ville de Königsberg avec tkz-graph



```
\begin{tikzpicture}
\SetGraphUnit{3}
\GraphInit[vstyle=Shade]
\tikzset{LabelStyle/.style= {draw,
                             fill = yellow,
                             text = red}}

\Vertex{A}
\EA(A){B}
\EA(B){C}
\SetGraphUnit{6}
% modifie la distance entre les nodes
\NO(B){D}
\Edge[label=1](B)(D)
\tikzset{EdgeStyle/.append style = {bend left}}
\Edge[label=4](A)(B)
\Edge[label=5](B)(A)
\Edge[label=6](B)(C)
\Edge[label=7](C)(B)
\Edge[label=2](A)(D)
\Edge[label=3](D)(C)
\end{tikzpicture}
```

Ce dernier exemple était important sur un plan historique, mais il était un peu compliqué car on doit modifier des styles.

2.5 La ville de Königsberg avec TikZ mais sans tkz-graph

Voyons l'exemple précédent, sans l'utilisation du package **tkz-graph**. L'exemple peut être vu sur cet excellent site <http://www.texample.net/tikz/examples/bridges-of-konigsberg/>, voici le code complet. The result is on the next page. D'abord le préambule

```
% The seven bridges of Königsberg
% Author : Alain Matthes
% Encoding : UTF8
% Engine : PDFLaTeX
\documentclass[ ]{article}
\usepackage[utf8]{inputenc}
\usepackage{fullpage}
\usepackage{fourier}
\usepackage{tikz}
\usetikzlibrary{arrows,shapes,positioning}
\begin{document}
```

Ensuite les styles principaux

```
\begin{center}
\begin{tikzpicture}
\useasboundingbox (-1,-1) rectangle (11,11);
\tikzset{VertexStyle/.style = {shape
                               = circle,
                               ball color = orange,
                               text = black,
                               inner sep = 2pt,
                               outer sep = 0pt,
                               minimum size = 24 pt}}

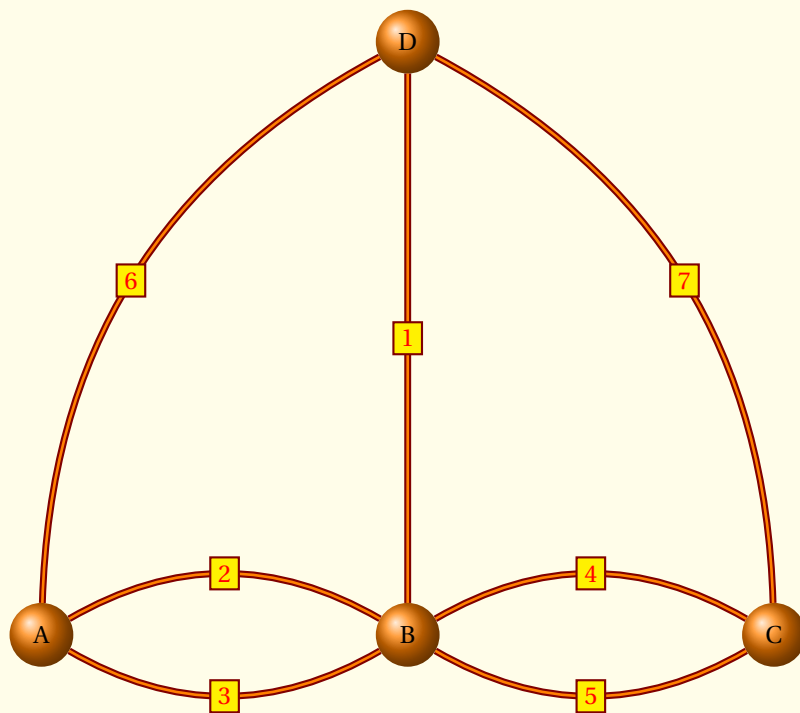
\tikzset{EdgeStyle/.style = {thick,
                              double = orange,
                              double distance = 1pt}}

\tikzset{LabelStyle/.style = {draw,
                              fill = yellow,
```

```
text = red}}
```

enfin, le tracé

```
\node[VertexStyle](A){A};
\node[VertexStyle,right=of A](B){B};
\node[VertexStyle,right=of B](C){C};
\node[VertexStyle,above= 7 cm of B](D){D};
\draw[EdgeStyle](B) to node[LabelStyle]{1} (D) ;
\tikzset{EdgeStyle/.append style = {bend left}}
\draw[EdgeStyle](A) to node[LabelStyle]{2} (B);
\draw[EdgeStyle](B) to node[LabelStyle]{3} (A);
\draw[EdgeStyle](B) to node[LabelStyle]{4} (C);
\draw[EdgeStyle](C) to node[LabelStyle]{5} (B);
\draw[EdgeStyle](A) to node[LabelStyle]{6} (D);
\draw[EdgeStyle](D) to node[LabelStyle]{7} (C);
\end{tikzpicture}
\end{center}
\end{document}
```



Vertex

C'est bien évidemment la macro essentielle qui permet de placer des sommets. Les sommets peuvent être placés avec un système de coordonnées rectangulaires ou bien polaires ou encore relativement les uns par rapport aux autres. Quelques dispositions particulières sont également possibles.

3.1 `\Vertex`

`\Vertex[⟨local options⟩]{⟨Name⟩}`

Un sommet se caractérise par :

- sa référence,
- sa position,
- son label,
- et le style.

Un argument non vide **Name** est obligatoire. Cet argument définit le nom de référence du node. C'est celui que l'on doit utiliser dans toute création de sommet (`\Vertex`) Il ne faut pas le confondre avec le **label** (étiquette) qui sera utilisé pour l'affichage. On peut vouloir afficher M_1 alors que le nom lui sera $M1$.

Des options sont utilisées pour définir les quatre premières caractéristiques. Les styles texte et graphique sont traités séparément.

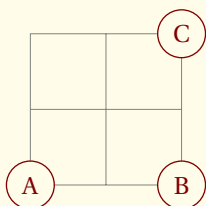
Options	Défaut	Définition
x	{}	abscisse
y	{}	ordonnée
a	{}	angle
d	{}	distance
Node	false	utilisation d'une référence déjà définie
position	{}	style qui permet un positionnement relatif
dir	\EA	direction pour un positionnement relatif
empty	false	booléen permettant de ne pas afficher le sommet
NoLabel	false	booléen supprime le label
LabelOut	false	booléen Label extérieur au node
L	{}	Le label
Math	false	booléen qui affiche le label en mode math
Ldist	0cm	distance du label au node
Lpos	0	position du label par rapport au node

Cette macro permet de définir un sommet qui a un nom **name** et un label.

Si **L=** alors **label = Name** sinon **label = L**.

3.1.1 Utilisation de coordonnées cartésiennes

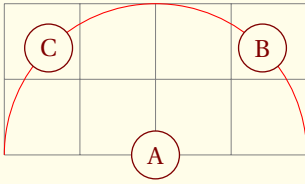
`\Vertex[x=⟨number⟩, y=⟨number⟩]{⟨name⟩}`. Coordonnées cartésiennes x et y .



```
\begin{tikzpicture}
  \GraphInit[vstyle=Normal]
  \draw[help lines] (0,0) grid (2,2);
  \Vertex{A} % par défaut x = 0 et y = 0
  \Vertex[x=2 , y=0]{B} \Vertex[x=2 , y=2]{C}
\end{tikzpicture}
```

3.1.2 Utilisation de coordonnées polaires

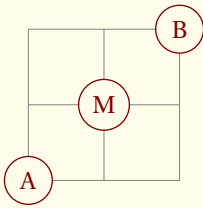
`\Vertex[a=<number>,d=<number>]{<vertex>}` Les coordonnées polaires peuvent être aussi utilisées. J'ai utilisé une grille d'aide afin de constater le placement du sommet.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Normal]
  \draw[help lines] (-2,0) grid (2,2);
  \draw[red] (2,0) arc (0:180: 2 cm);
  \Vertex{A}
  \Vertex[a=45 , d=2 cm]{B}
  \Vertex[a=135 , d=2 cm]{C}
\end{tikzpicture}
```

3.1.3 Option Node : utilisation d'une position référencée

Cette option permet de placer un sommet sur un Node déjà défini ou bien un objet du type « coordinate ».



```
\begin{tikzpicture}
  \GraphInit[vstyle=Normal]
  \draw[help lines] (0,0) grid (2,2);
  \Vertex{A} \Vertex[x=2 , y=2]{B}
  %\tkzActivOff nécessaire avec frenchb et babel
  \tkzActivOff
  \coordinate (M) at ($ (A)!.5!(B) $){};
  \tkzActivOn
  \Vertex[Node]{M}
\end{tikzpicture}
```

3.2 Raccourcis pour placement relatif

Pour effectuer des placements relatifs, il est nécessaire de définir une distance unité entre deux sommets. La macro suivante permet de définir cette distance.

\SetGraphUnit{<nombre>}

Cette macro permet de définir la distance entre deux sommets. La distance se réfère aux centres de ces sommets et le nombre est exprimé en **cm**. Par défaut, l'unité est 1 cm.

utilisation : **\SetGraphUnit{2}**

\ShortCut[<local options>](<vertex A>){<vertex B>}

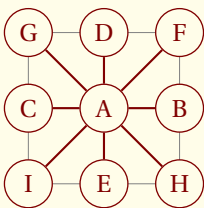
Ces raccourcis permettent de créer un **vertex B** relativement à un **vertex A**. La distance entre les deux sommets est déterminé par la valeur de **unit** et par les unités de **TikZ**. Horizontalement et verticalement la distance est définie par **unit** \times **x** et **unit** \times **y**. La valeur de **unit** peut être redéfinie par la macro **\SetGraphUnit** ou bien avec l'option **unit**. Avec l'option la définition est locale ; avec la macro, la définition est globale mais elle peut être locale si elle est interviens dans un groupe \TeX ou un environnement **scope**. Les raccourcis sont :

Raccourcis	Définition
\EA	à l'est
\WE	à l'ouest
\NO	au nord
\SO	au sud
\NOEA	au nord-est soit "nord" puis "est"
\NOWE	au nord-ouest soit "nord" puis "ouest"
\SOEA	au sud-est soit "sud" puis "est"
\SOWE	au sud-ouest soit "sud" puis "ouest"

\NOEA est un raccourci pour **\NO\EA**. par défaut, la distance entre les sommets avec ce raccourci est $\sqrt{2} \times \text{unit} = \sqrt{2}$. Les options sont celles de la macro **\Vertex**.

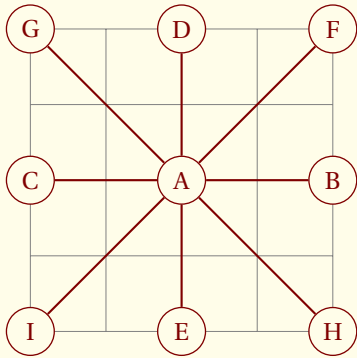
Nous allons d'abord modifier la distance entre deux noeuds d'une façon générale avec **\SetGraphUnit{2}** sinon par défaut **unit =1**.

3.2.1 Utilisation des raccourcis avec les valeurs par défaut



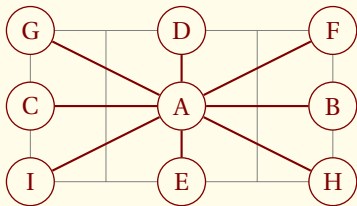
```
\begin{tikzpicture}
  \draw[help lines] (-1,-1) grid (1,1);
  \GraphInit[vstyle=Normal]
  \Vertex{A}
  \EA(A){B} \WE(A){C} \NO(A){D} \SO(A){E}
  \NOEA(A){F} \NOWE(A){G} \SOEA(A){H} \SOWE(A){I}
  \foreach \v in {B,C,D,E,F,G,H,I}{\Edge(A)(\v)};
\end{tikzpicture}
```

3.2.2 Modification de l'unité avec `\SetGraphUnit`



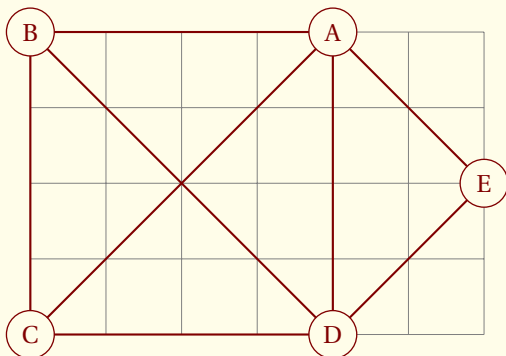
```
\begin{tikzpicture}
  \draw[help lines] (-2,-2) grid (2,2);
  \SetGraphUnit{2}
  \GraphInit[vstyle=Normal]
  \Vertex{A}
  \EA(A){B} \WE(A){C} \NO(A){D} \SO(A){E}
  \NOEA(A){F} \NOWE(A){G} \SOEA(A){H} \SOWE(A){I}
  \foreach \v in {B,C,D,E,F,G,H,I}{\Edge(A)(\v)};
\end{tikzpicture}
```

3.2.3 Modification des unités de TikZ : `x=2 cm,y=1 cm`



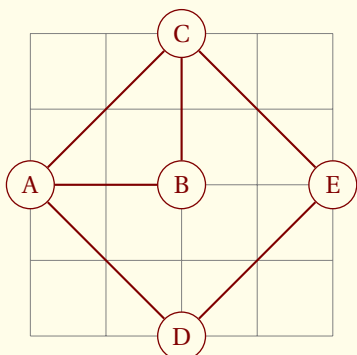
```
\begin{tikzpicture}[x=2 cm,y=1 cm]
  \draw[help lines] (-1,-1) grid (1,1);
  \GraphInit[vstyle=Normal]
  \Vertex{A}
  \EA(A){B} \WE(A){C} \NO(A){D} \SO(A){E}
  \NOEA(A){F} \NOWE(A){G} \SOEA(A){H} \SOWE(A){I}
  \foreach \v in {B,C,D,E,F,G,H,I}{\Edge(A)(\v)};
\end{tikzpicture}
```

3.2.4 Exemple classique



```
\begin{tikzpicture}
  \draw[help lines] (-2,-2) grid (4,2);
  \SetGraphUnit{2}
  \coordinate (0) at (0,0);
  \NOEA(0){A} \NOWE(0){B} \SOEA(0){D}
  \SOWE(0){C} \NOEA(D){E}
  \Edges(B,C,D,A,E,D,B,A,C)
\end{tikzpicture}
```

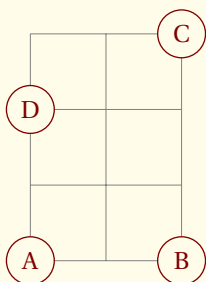
3.2.5 Autre exemple classique



```
\begin{tikzpicture}
  \draw[help lines] (0,-2) grid (4,2);
  \SetGraphUnit{2}
  \GraphInit[vstyle=Normal]
  \Vertex{A}
  \EA(A){B} \NO(B){C} \SO(B){D} \EA(B){E}
  \Edges(A,B,C,A,D,E,C)
\end{tikzpicture}
```

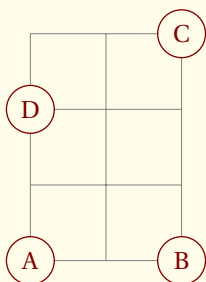
3.2.6 Modication locale de unit avec l'option

Le plus simple :



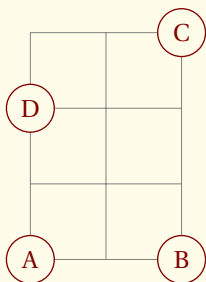
```
\begin{tikzpicture}
  \draw[help lines] (0,0) grid (2,3);
  \SetGraphUnit{2}
  \Vertex{A} \EA(A){B}
  \NO[unit=3](B){C}
  \NO(A){D}
\end{tikzpicture}
```

3.2.7 Modication locale de unit avec l'environnement scope



```
\begin{tikzpicture}
  \draw[help lines] (0,0) grid (2,3);
  \SetGraphUnit{2}
  \Vertex{A} \EA(A){B}
  \begin{scope}
    \SetGraphUnit{3} \NO(B){C}
  \end{scope}
  \NO(A){D}
\end{tikzpicture}
```

3.2.8 Modication locale de unit avec un groupe T_EX



```
\begin{tikzpicture}
  \draw[help lines] (0,0) grid (2,3);
  \SetGraphUnit{2}
  \Vertex{A} \EA(A){B}
  {\SetGraphUnit{3} \NO(B){C}}
  \NO(A){D}
\end{tikzpicture}
```


SECTION 4

Placement de sommets sur une forme géométrique

Il s'agit ici de placer un groupe de sommets suivant une direction donnée ou bien encore suivant une forme prédéfinie. Les sommets sont placés avec comme support une figure géométrique simple. La macro principale utilise une direction définie à l'aide de l'option `dir`, la version étoilée une forme particulière triangulaire, carrée etc...

`\Vertices[⟨local options⟩]{⟨type⟩}{⟨List of vertices⟩}`

Il y a donc plusieurs types de formes géométriques, droite, triangle, carrés et cercles. La macro `\SetGraphUnit` permet de modifier les longueurs. Pour les sommets alignés, ceux-ci sont placés suivant une direction donnée par EA, WE, NO, SO, NOEA, NOWE, SOEA, SOWE.

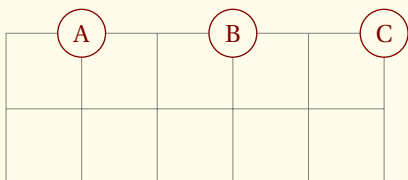
Premier Argument	Définition
line	Sommets alignés, une option détermine la direction
tr1	première forme de triangle
tr2	deuxième forme de triangle
tr3	troisième forme de triangle
tr4	quatrième forme de triangle
square	quatre sommets sur un carré
circle	sommets sur une cercle

Le second argument est une liste de noms pour les sommets.

Options	Défaut	Définition
dir	EA	permet de placer plusieurs sommets alignés

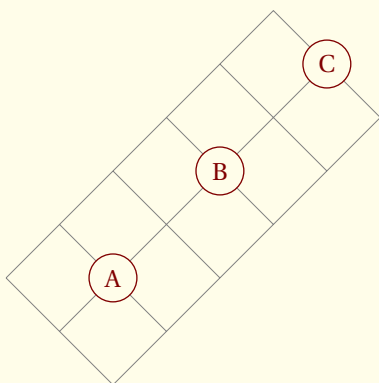
Les options sont celles d'un sommet (*Vertex*).

4.1 `\Vertices` à partir d'un sommet défini par des coordonnées



```
\begin{tikzpicture}
  \SetGraphUnit{2}
  \draw[help lines] (0,0) grid (5,2);
  \Vertices[x=1,y=2]{line}{A,B,C}
\end{tikzpicture}
```

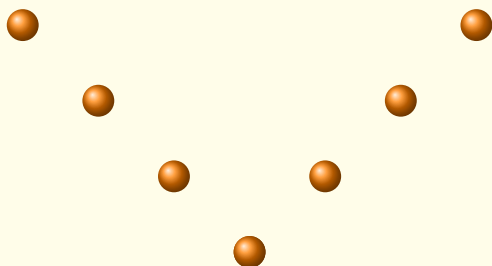
4.2 `\Vertices` à partir d'une position donnée.



```
\begin{tikzpicture}[rotate=45]
  \SetGraphUnit{2}
  \draw[help lines] (0,0) grid (5,2);
  \coordinate (A) at (1,1);
  \Vertices[Node]{line}{A,B,C}
\end{tikzpicture}
```

4.3 Exemples avec une direction

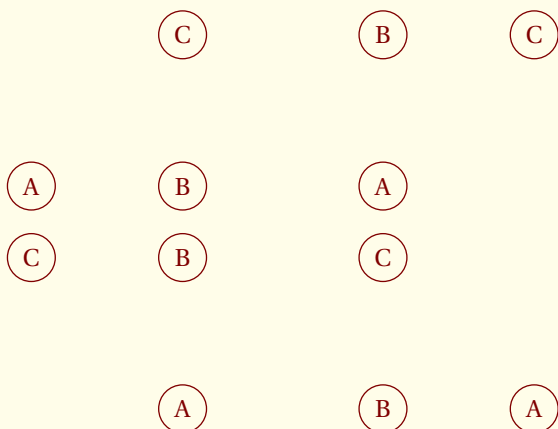
Il s'agit ici de placer une liste de sommets suivant une direction donnée, cette direction est définie à l'aide de l'option `dir`.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \Vertices[dir=\NOEA]{line}{A,B,C,D}
  \Vertices[dir=\NOWE]{line}{A,E,F,G}
\end{tikzpicture}
```

4.4 Placement sur un triangle

Il y a différentes possibilités avec une forme triangulaire, mais les triangles sont isocèles rectangles. Voici dans l'ordre les formes `tr1`, `tr2`, `tr3` et `tr4`

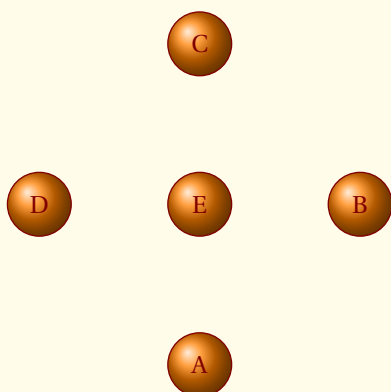


```
\begin{tikzpicture}\SetGraphUnit{2}
  \Vertices{tr1}{A,B,C}
\end{tikzpicture}\hspace*{2cm}
\begin{tikzpicture}\SetGraphUnit{2}
  \Vertices{tr2}{A,B,C}
\end{tikzpicture}
```

```
\begin{tikzpicture}\SetGraphUnit{2}
  \Vertices{tr3}{A,B,C}
\end{tikzpicture}\hspace*{2cm}
\begin{tikzpicture}\SetGraphUnit{2}
  \Vertices{tr4}{A,B,C}
\end{tikzpicture}
```

4.5 Utilisation d'un carré

Deux autres possibilités de placer un node. La première utilise un node obtenu à l'aide d'une intersection (voir le `pgfmanual`). Dans la première, j'ai redéfini la distance unité entre deux sommets à l'aide de `\SetGraphUnit`.



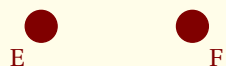
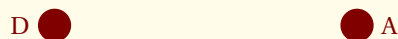
```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Shade]
  \Vertices{square}{A,B,C,D}
  \coordinate (E) at (intersection of A--C and B--D);
  \Vertex[Node]{E}% voir option node
\end{tikzpicture}
```

4.6 Utilisation d'un cercle

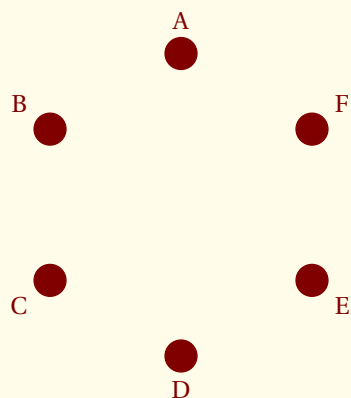
```
\begin{tikzpicture}
  \SetGraphUnit{2}
  \Vertices{circle}{A,B,C,D}
\end{tikzpicture}
```

**4.7 Utilisation d'un cercle et positionnement des labels**

```
\begin{tikzpicture} \SetGraphUnit{2}
  \GraphInit[vstyle=Classic]
  \Vertices{circle}{A,B,C,D,E,F}
\end{tikzpicture}
```

**4.8 Rotation et labels externes**

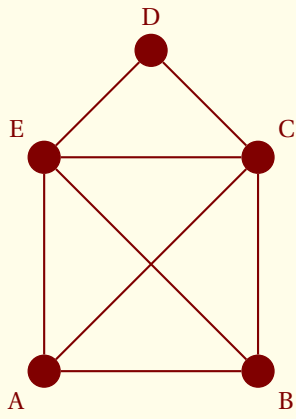
$Lpos = \text{angle de la rotation}$. Cela permet de faire une rotation du label autour du centre de chaque sommet et de suivre la rotation du graphe. Il suffit pour comprendre cette option de compiler l'exemple en l'omettant.



```
\begin{tikzpicture}[rotate=90]
  \GraphInit[vstyle=Classic]
  \Vertices[Lpos=90,unit=2]{circle}{A,B,C,D,E,F}
\end{tikzpicture}
```

4.9 Placement sur un cercle

Avec des labels externes, il faut procéder avec précaution.



```
\begin{tikzpicture}[scale=.5]
  \SetGraphUnit{4}
  \GraphInit[vstyle=Classic]
  \begin{scope}[rotate=45]
    \Vertices[Lpos=45]{circle}{C,E,A,B}
  \end{scope}
  \NOEA[Lpos=90,unit=2.828](E){D}
  \Edges(A,B,E,D,C,E,A,C,B)
\end{tikzpicture}
```

Les labels

Rappel : Si aucun label n'est donné alors l'affichage du label est celui de la référence du **vertex**. Il est possible de modifier localement le comportement des labels

5.1 Options concernant les labels

L'option suivante permet de définir un label, celui-ci peut être en mode texte ou bien en mode math.

5.1.1 Option L



```
\begin{tikzpicture}
  \Vertex[L=$\alpha$] {a}
  \EA[unit=4](a){b}
\end{tikzpicture}
```

5.1.2 Option Math

Le label est en mode math. Il est inutile de placer L en mode math si l'option est utilisée.



```
\begin{tikzpicture}
  \Vertex[Math] {A_1}
  \Vertex[Math,L=\alpha,x=4,y=0] {a}
\end{tikzpicture}
```

5.1.3 Suppression d'un label, Option NoLabel

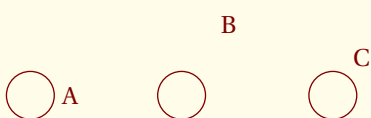
Cette option supprime l'affichage du label. Il est préférable d'utiliser **SetVertexNoLabel** si on veut généraliser à tous les sommets.



```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \Vertex[NoLabel]{A}
  \EA[NoLabel](A){B}
\end{tikzpicture}
```

5.1.4 Option LabelOut, Lpos et Ldist

La première option permet de placer le label hors du node, la deuxième positionne le label autour du sommet et la dernière spécifie la distance entre le label et le sommet.



```
\begin{tikzpicture}
  \Vertex[LabelOut]{A}
  \Vertex[LabelOut,Lpos=60,
    Ldist=.5cm,x=2,y=0]{B}
  \Vertex[LabelOut,Lpos=60,x=4,y=0]{C}
\end{tikzpicture}
```

On peut souhaiter appliquer une option pour tous les sommets.

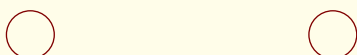
5.2 \SetVertexNoLabel

On peut souhaiter ne pas avoir de label pour tous les sommets avec un style prédéfini.

\SetVertexNoLabel

Cette macro permet de supprimer les labels sur tous les sommets. Elle agit globalement sur tous les sommets. Elle correspond à l'option **NoLabel**.

5.2.1 Suppression des labels

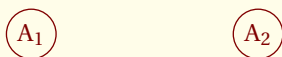


```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \SetVertexNoLabel
  \Vertex{A}\EA(A){B}
\end{tikzpicture}
```

5.3 \SetVertexMath

\SetVertexMath

Cette macro permet d'appliquer l'option **Math** à plusieurs sommets. Elle agit globalement sur tous les sommets. Elle correspond à l'option **Math**.



```
\begin{tikzpicture}
  \SetVertexMath
  \Vertex {A_1} \EA[unit=3] (A_1) {A_2}\texttt{}
\end{tikzpicture}
```

5.4 \SetVertexLabel

\SetVertexLabel

Cette macro autorise les labels. Elle agit globalement sur tous les sommets.

5.4.1 Labels supprimés puis autorisés.

Dans l'exemple qui suit, les labels sont supprimés puis autorisés.



```
\begin{tikzpicture}
  \SetVertexNoLabel
  \SetGraphUnit{2}
  \Vertex {A} \EA(A){B}
  \SetVertexLabel \EA(B){C}
\end{tikzpicture}
```

5.4.2 Label en dehors du sommet \SetVertexLabelOut

\SetVertexLabelOut

\SetVertexLabelOut Dans les exemples précédents, les sommets sont des petits disques colorés, généralement en noir et dans ce cas par défaut le label est à l'extérieur. On peut contrôler la position à l'aide des labels avec **Ldist** et **Lpos**.

\SetVertexLabelIn

\SetVertexLabelIn permet d'écrire le label dans le sommet.

Cette macro permet d'appliquer l'option à plusieurs sommets. **\SetVertexLabelIn** annule l'effet.



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \SetVertexLabelOut
  \Vertex {A} \EA(A){B}
  \SetVertexLabelIn \SO[unit=3](B){C}
\end{tikzpicture}
```



Edge avec tkz-graph

`\Edge[⟨local options⟩](⟨Vertex A⟩)(⟨Vertex B⟩)`

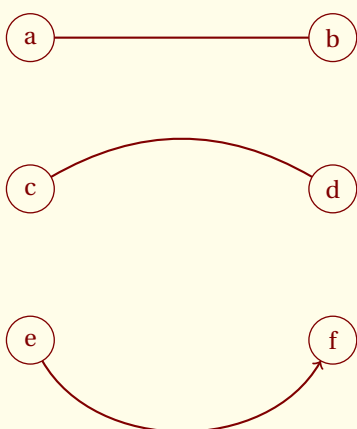
options	défaut	définition
local	false	booléen désactive EdgeStyle
color	\EdgeColor	couleur de l'arête
lw	\EdgeLineWidth	épaisseur de l'arête.
label	{}	le label
labeltext	black	couleur du texte
labelcolor	white	couleur du fond du label
labelstyle	{}	modification du style du label
style	pos=.5	modification du style général

Cette macro permet de tracer une arête entre deux sommets. Dans les exemples et dans le chapitre sur les styles, l'usage des styles est expliqué.

6.1 Utilisation de \Edge

On peut remarquer qu'il y a deux sortes d'arêtes au niveau de la forme : les segments et les arcs. De plus, ces arêtes peuvent avoir un label. La notion de style est importante car on peut définir pour toutes les arêtes un même style dès le début.

par défaut :



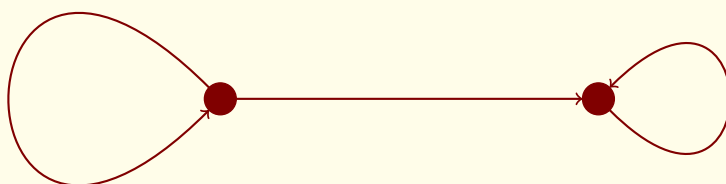
```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \Vertex{a}
  \EA(a){b}
  \SO[unit=2](a){c}
  \EA(c){d}
  {\SetGraphUnit{2}
  \SO(c){e}}
  \EA(e){f}
  \Edge(a)(b)
  \tikzset{EdgeStyle/.style = {-,bend left}}
  \Edge(c)(d)
  \tikzset{EdgeStyle/.style = {->,bend right=60}}
  \Edge(e)(f)
\end{tikzpicture}
```


6.2 Arête particulière la boucle : Loop

`\Loop[⟨local options⟩](⟨Vertex⟩)`

options	défaut	définition
color	black	
lw	0.8pt	
label	{}	
labelstyle	{}	
style	{}	

6.2.1 Exemple avec \Loop



```
\begin{tikzpicture}
  \useasboundingbox (-1,-2) rectangle (8,2);
  \SetVertexSimple
  \SetGraphUnit{5}
  \Vertex{A}
  \EA(A){B}
  \Edge[style={->}](A)(B)
  \Loop[dist=3cm,dir=EA,style={thick,->}](B)
  \Loop[dist=5cm,dir=WE,style={thick,->}](A)
\end{tikzpicture}
```

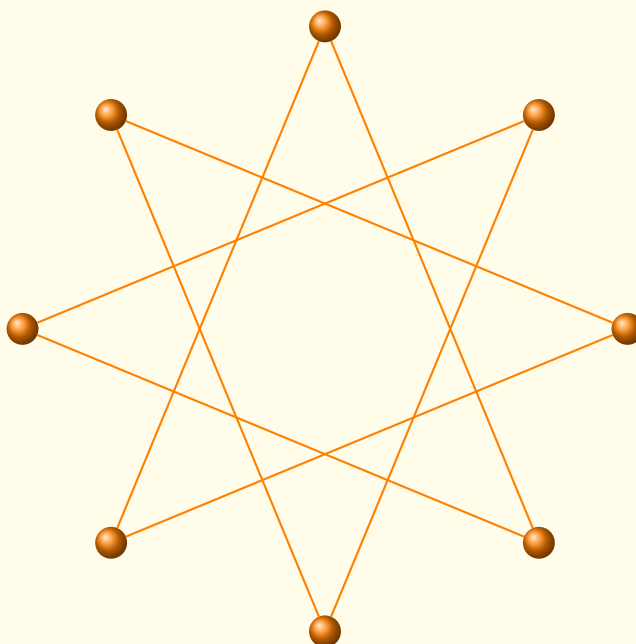
6.3 Multiple arêtes \Edges

`\Edges[⟨local options⟩](⟨Vertex A,Vertex B,...⟩)`

options	défaut	définition
color	black	
lw	thick	
label	{}	
labelstyle	{}	
style	{}	

Cette macro permet de définir une série d'arêtes en une seule fois.

6.3.1 Exemple avec \Edges



```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \GraphInit[vstyle=Art]
  \Vertices{circle}{a0,a1,a2,a3,a4,a5,a6,a7}
  \Edges(a0,a3,a6,a1,a4,a7,a2,a5,a0)
\end{tikzpicture}
```

Modification des styles des sommets

Différentes méthodes sont possibles mais il faut distinguer une utilisation globale ou locale.

Les trois principaux styles sont **VertexStyle**, **EdgeStyle** et **LabelStyle**. Le dernier est attaché aux étiquettes que peuvent avoir les arêtes.

1. `\GraphInit` permet de choisir un style prédéfini et il est possible de retoucher ces styles en modifiant les valeurs choisies par défaut.
2. Les styles des sommets, des arêtes et étiquettes peuvent être personnalisés avec **VertexStyle**, **EdgeStyle** et **LabelStyle**. On peut redéfinir ces styles avec `\tikzset{VertexStyle/.append style = { ... }}` ou bien `\tikzset{VertexStyle/.style = { ... }}`. La première méthode modifie un style existant alors que la seconde définit un style.
3. On peut utiliser les anciennes macros : `\SetVertexSimple`, `\SetVertexNormal`, `\SetUpVertex` et `\SetUpEdge`.

Il est possible de mélanger tout cela en sachant que la dernière définition d'un style l'emporte.

`\GraphInit[⟨local options⟩]`

Options	Défaut	Définition
vstyle	Normal	

Les possibilités pour **vstyle** sont :

1. Empty,
2. Hasse,
3. Simple,
4. Classic,
5. Normal,
6. Shade,
7. Dijkstra
8. Welsh,
9. Art,
10. Shade Art.

Il y a pour le moment 10 styles pré-définis. Il est possible de modifier les valeurs par défaut.

Utilisation des styles pré-définis

1. GraphInit par défaut



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Normal]
  \Vertex{A}\EA(A){B}
  \Edge(A)(B)
\end{tikzpicture}
```

2. GraphInit et vstyle=Empty



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Empty]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

3. GraphInit et vstyle=Hasse



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Hasse]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

4. GraphInit et vstyle=Simple



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Simple]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

5. GraphInit et vstyle=Classic



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Classic]
  \Vertex[Lpos=-90]{A}
  \EA[Lpos=-90](A){B}\Edge(A)(B)
\end{tikzpicture}
```

6. GraphInit et vstyle=Normal



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Normal]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```



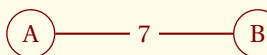
```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Classic]
  \Vertex[Lpos=-90]{Paris}
  \EA[Lpos=-90](Paris){Berlin}
  \Edge (Paris)(Berlin)
\end{tikzpicture}
```

7. GraphInit et vstyle=Shade



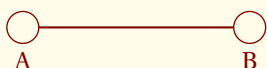
```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Shade]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

8. GraphInit et vstyle=Dijkstra



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Dijkstra]
  \Vertex{A}\EA(A){B}\Edge[label=$7$](A)(B)
\end{tikzpicture}
```

9. GraphInit et vstyle=Welsh



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Welsh]
  \Vertex[Lpos=-90]{A}
  \EA[Lpos=-90](A){B}\Edge(A)(B)
\end{tikzpicture}
```

10. GraphInit et vstyle=Art



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Art]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

11. GraphInit et vstyle=Shade Art

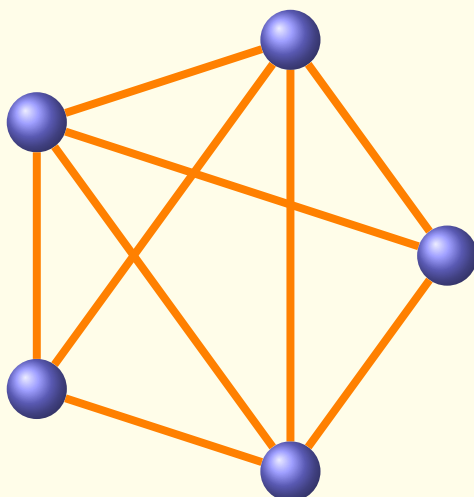


```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Shade Art]
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

`vstyle` est basé sur les macros suivantes qui peuvent être redéfinies.

Commandes pour les styles	utilisation
<code>\newcommand*{\VertexInnerSep}{0pt}</code>	
<code>\newcommand*{\VertexOuterSep}{0pt}</code>	
<code>\newcommand*{\VertexDistance}{3cm}</code>	
<code>\newcommand*{\VertexShape}{circle}</code>	
<code>\newcommand*{\VertexLineWidth}{0.8pt}</code>	
<code>\newcommand*{\VertexLineColor}{black}</code>	
<code>\newcommand*{\VertexLightFillColor}{white}</code>	
<code>\newcommand*{\VertexDarkFillColor}{black}</code>	
<code>\newcommand*{\VertexTextColor}{black}</code>	
<code>\newcommand*{\VertexFillColor}{black}</code>	
<code>\newcommand*{\VertexBallColor}{orange}</code>	
<code>\newcommand*{\VertexBigMinSize}{24pt}</code>	
<code>\newcommand*{\VertexInterMinSize}{18pt}</code>	
<code>\newcommand*{\VertexSmallMinSize}{12pt}</code>	
<code>\newcommand*{\EdgeFillColor}{orange}</code>	
<code>\newcommand*{\EdgeArtColor}{orange}</code>	
<code>\newcommand*{\EdgeColor}{black}</code>	
<code>\newcommand*{\EdgeDoubleDistance}{1pt}</code>	
<code>\newcommand*{\EdgeLineWidth}{0.8pt}</code>	

7.1 Modification de `vstyle=Art`



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \GraphInit[vstyle=Art]
  \renewcommand*{\VertexInnerSep}{8pt}
  \renewcommand*{\EdgeLineWidth}{3pt}
  \renewcommand*{\VertexBallColor}{blue!50}
  \Vertices{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```

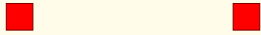
7.2 Modification du style `VertexStyle` par défaut

Il est possible de redéfinir le style `\SetVertexSimple`.

Par défaut :

```
\tikzset{VertexStyle/.style = {
    shape      = circle,
    fill       = black,
    inner sep  = 0pt,
    outer sep  = 0pt,
    minimum size = 8pt,
    draw}]
```

maintenant si on utilise ceci :



```
\begin{tikzpicture}
  \SetVertexSimple
  \tikzset{VertexStyle/.style = {
    shape      = rectangle,
    fill       = red,%
    inner sep  = 0pt,
    outer sep  = 0pt,
    minimum size = 10pt,
    draw}}
  \SetGraphUnit{3}
  \Vertex{A}\EA(A){B}
\end{tikzpicture}
```

7.3 Modification d'un style `VertexStyle`

C'est le style par défaut pour les sommets mais on peut le modifier. Voici quelques exemples utilisés plus tard dans ce document

par défaut :



```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \tikzset{VertexStyle/.style = {%
    shape      = circle,
    shading    = ball,
    ball color = Orange,
    minimum size = 20pt,draw}}
  \SetVertexNoLabel
  \Vertex{A}\EA[unit=3](A){B}
\end{tikzpicture}
```

ou bien encore :

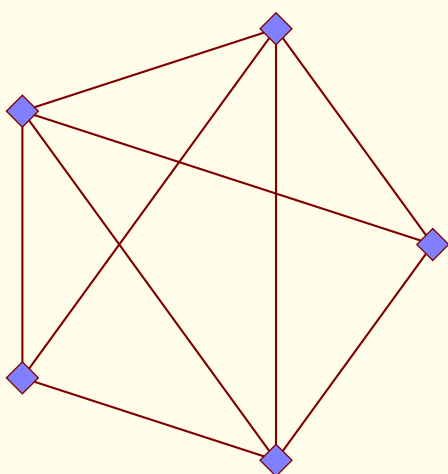


```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \tikzset{VertexStyle/.style = {%
    shape      = circle,
    shading    = ball,
    ball color = green!40!black,%
    minimum size = 30pt,draw}}
  \SetVertexNoLabel
  \Vertex{A}\EA[unit=3](A){B}
\end{tikzpicture}
```

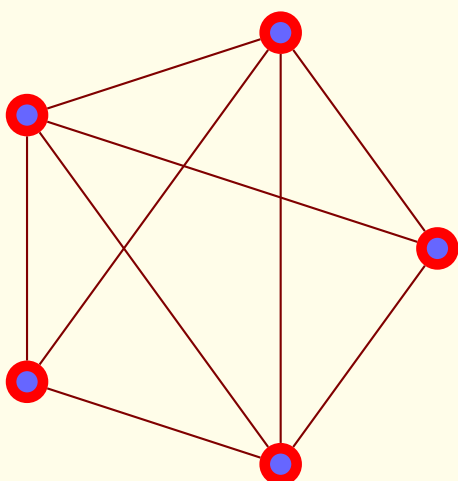
`\SetVertexSimple[⟨local options⟩]`

Il est possible de modifier les styles prédéfinis. La macro `\SetVertexSimple` permet d'affiner le style « Simple » des

	options	default	definition
sommets.	Shape	<code>\VertexShape</code>	
	MinSize	<code>\VertexSmallMinSize</code>	
	LineWidth	<code>\VertexLineWidth</code>	
	LineColor	<code>\VertexLineColor</code>	
	FillColor	<code>\VertexFillColor</code>	

7.4 Autre style `\SetVertexSimple`

```
\begin{tikzpicture}
  \SetVertexSimple[Shape=diamond,
                  FillColor=blue!50]
  \Vertices[unit=3]{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```

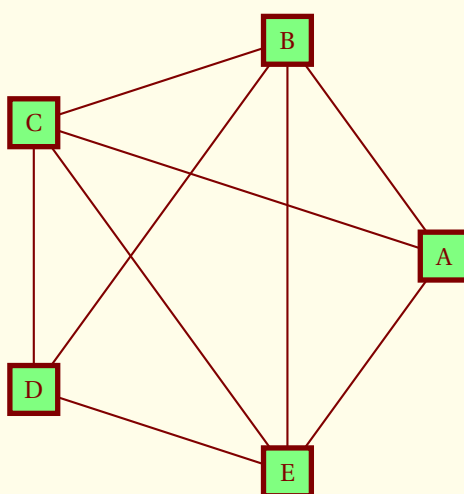
7.5 `\SetVertexSimple, inner sep et outer sep`

```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \SetVertexSimple[MinSize = 12pt,
                  LineWidth = 4pt,
                  LineColor = red,%
                  FillColor = blue!60]
  \tikzset{VertexStyle/.append style =
    {inner sep = 0pt,%
     outer sep = 2pt}}
  \Vertices{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```


`\SetVertexNormal[⟨local options⟩]`

Options	Défaut	Définition
color	<code>\EdgeColor</code>	
label	no default	
labelstyle	no default	
labeltext	<code>\LabelTextColor</code>	
labelcolor	<code>\LabelFillColor</code>	
style	no default	
lw	<code>\EdgeLineWidth</code>	

Macro semblable à la précédente.

7.6 Autre style `\SetVertexNormal`

```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \SetVertexNormal[Shape      = rectangle,%
                  LineWidth = 2pt,%
                  FillColor = green!50]
  \Vertices{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```

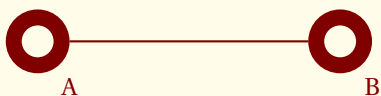
\SetUpVertex[⟨local options⟩]

Options	Défaut	Définition
Lpos	-90	position label externe
Ldist	0cm	distance du label
style		permet d'affiner le style
NoLabel	false	supprime le label
LabelOut	false	Label externe

Cette macro permet de modifier les options précédentes.

7.7 \SetUpVertex

```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \SetUpVertex[Lpos=-60,LabelOut]
  \Vertex{A}\EA(A){B}
\end{tikzpicture}
```

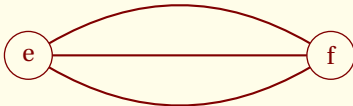
7.8 \SetUpVertex et \tikzset

```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \SetVertexLabel
  \SetUpVertex[Lpos=-60,LabelOut]
  \tikzset{VertexStyle/.append style =
    {outer sep = .5\pgflinewidth}}
  \renewcommand*{\VertexLineWidth}{6pt}
  \Vertex{A}\EA(A){B}\Edge(A)(B)
\end{tikzpicture}
```

Modification des styles des arêtes

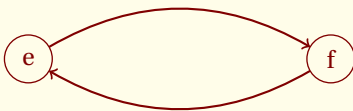
8.1 Utilisation de l'option `style` de la macro `\Edge`

8.1.1 Exemple 1



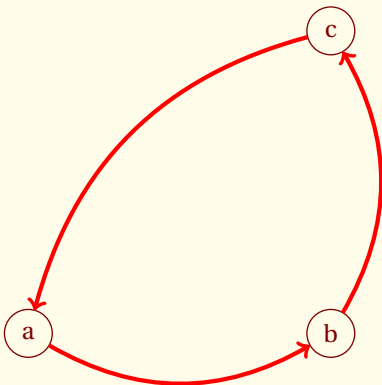
```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \Vertex{e}
  \EA(e){f}
  \Edge(f)(e)
  \Edge[style={bend left}](f)(e)
  \Edge[style={bend right}](f)(e)
\end{tikzpicture}
```

8.1.2 Exemple 2



```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \Vertex{e}
  \EA(e){f}
  \Edge[style={->,bend left}](f)(e)
  \Edge[style={<-,bend right}](f)(e)
\end{tikzpicture}
```

8.1.3 Exemple 3



```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \Vertex{a}
  \EA(a){b}
  \NO(b){c}
  \SetUpEdge[style={->,bend right,ultra thick},
             color=red]
  \Edge(a)(b)
  \Edge(b)(c)
  \Edge(c)(a)
\end{tikzpicture}
```

8.2 Modification des styles par défaut \SetUpEdge

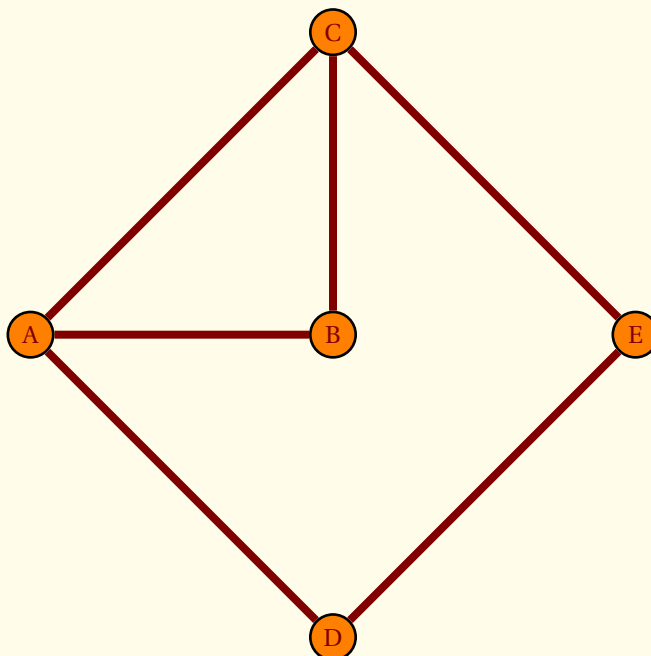
Cette macro a une action globale et permet de rédéfinir un style.

\SetUpEdge[⟨local options⟩]

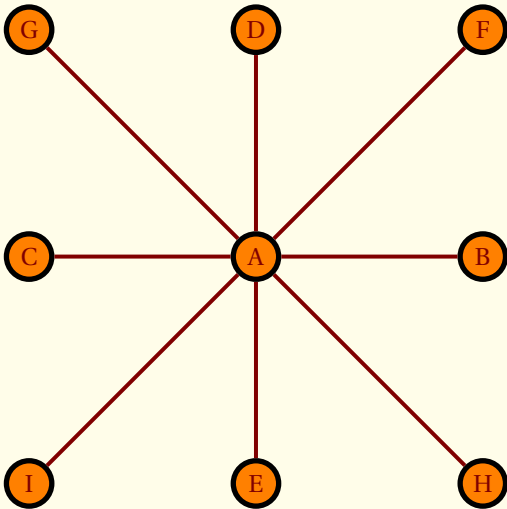
Options	Défaut	Définition
lw	-90	position label externe
color	\EdgeLineWidth	position label externe
label	0cm	distance du label
labelstyle		permet d'affiner le style
labeltext	false	supprime le label
style	false	Label externe

Cette macro permet de modifier les options précédentes.

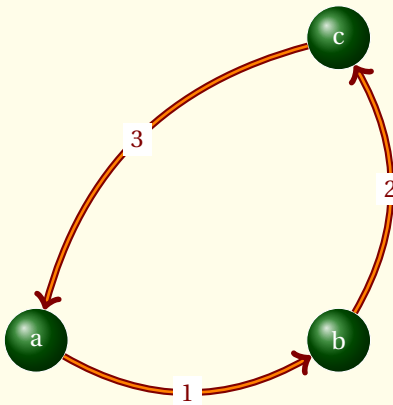
8.2.1 Utilisation de \SetUpEdge Exemple 1



```
\begin{tikzpicture}
  \SetGraphUnit{4}      \SetUpEdge[lw=3pt]
  \Vertex{A}
  \EA (A){B}      \NO (B){C}
  \SO (B){D}      \EA (B){E}
  \Edges(A,B,C,A,D,E,C)
\end{tikzpicture}
```

8.2.2 Utilisation de `\SetUpEdge` Exemple 2

```
\begin{tikzpicture}
  \SetGraphUnit{3}
  \SetUpEdge[lw=1.5pt]
  \Vertex{A}
  \EA(A){B} \WE(A){C} \NO(A){D}
  \SO(A){E} \NOEA(A){F} \NOWE(A){G}
  \SOEA(A){H} \SOWE(A){I}
  \foreach \v in {B,C,D,E,F,G,H,I}{%
    \Edge(A)(\v)};
\end{tikzpicture}
```

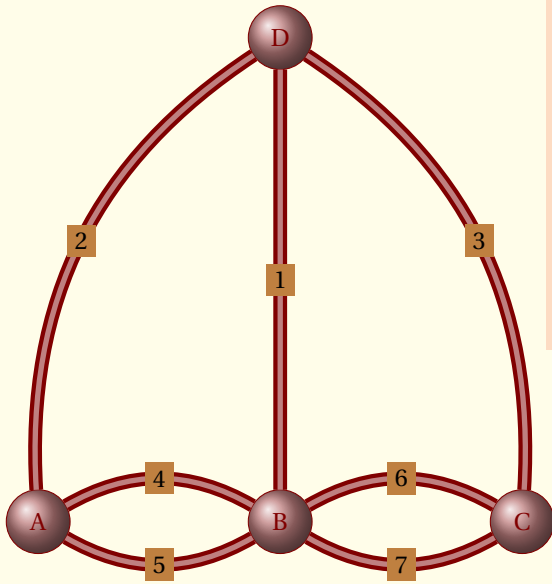
8.3 Arête avec label `LabelStyle`

```
\begin{tikzpicture}
  \SetGraphUnit{4}
  \tikzset{VertexStyle/.style =
    {draw,
      shape = circle,
      shading = ball,
      ball color = green!40!black,
      minimum size = 24pt,
      color = white}}
  \tikzset{EdgeStyle/.style =
    {->,bend right,
      thick,
      double = orange,
      double distance = 1pt}}
  \Vertex{a}
  \EA(a){b}
  \NO(b){c}
  \tikzset{LabelStyle/.style =
    {fill=white}}
  \Edge[label=$1$](a)(b)
  \Edge[label=$2$](b)(c)
  \Edge[label=$3$](c)(a)
\end{tikzpicture}
```

8.4 Utiliser un style intermédiaire

```
\SetGraphUnit{4}
\tikzset{VertexStyle/.style = {shape = circle,
  shading = ball,
  ball color = Maroon!50,
  minimum size = 24pt,
  draw}}
\tikzset{TempEdgeStyle/.style = {ultra thick,
  double = Maroon!50,
  double distance = 2pt}}
\tikzset{LabelStyle/.style = {color = brown,
```

```
text=black}}
```



```
\begin{tikzpicture}[scale=.8]
  \Vertex{A}
  \EA(A){B} \EA(B){C}
  \SetGraphUnit{8}
  \NO(B){D}
  \tikzset{EdgeStyle/.style = {TempEdgeStyle}}
  \Edge[label=1](B)(D)
  \tikzset{EdgeStyle/.style = {TempEdgeStyle,bend left}}
  \Edge[label=4](A)(B) \Edge[label=5](B)(A)
  \Edge[label=6](B)(C) \Edge[label=7](C)(B)
  \Edge[label=2](A)(D) \Edge[label=3](D)(C)
\end{tikzpicture}
```

Changement de couleurs dans les styles prédéfinis

Trois macros sont proposées

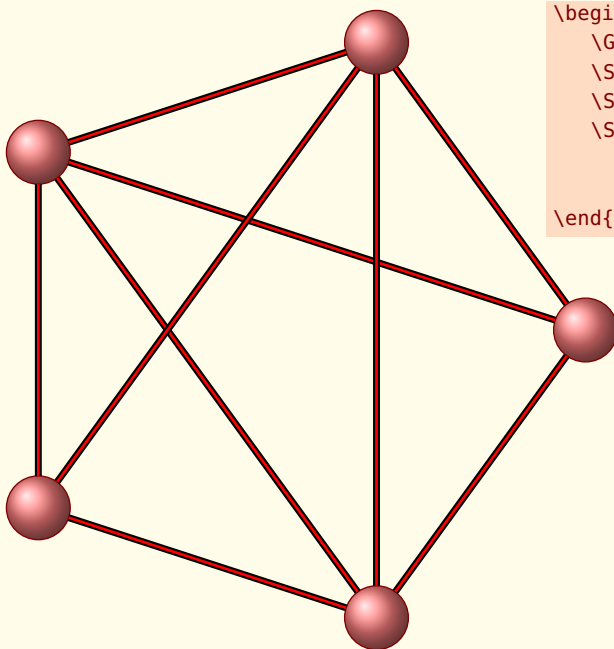
9.1 `\SetGraphShadeColor`

```
\SetGraphShadeColor{<ball color>}{<color>}{<double>}
```

`\SetGraphShadeColor` permet de modifier les couleurs pour le style **Shade**.

9.1.1 Exemple

Cet exemple utilise une macro de **tkz-berge**

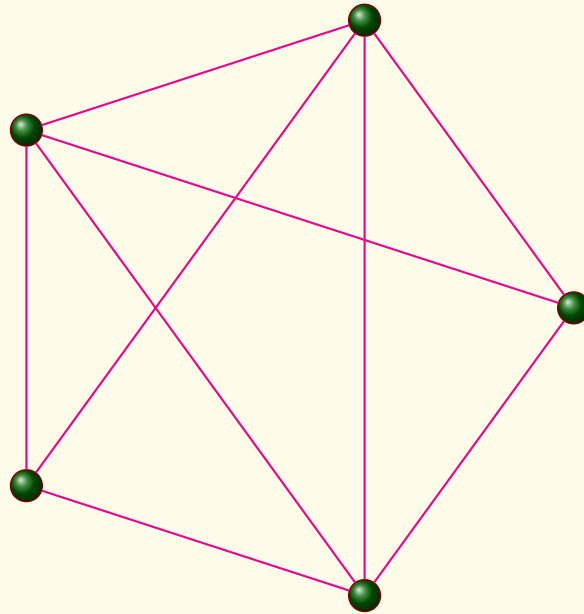


```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \SetGraphUnit{4}
  \SetVertexNoLabel
  \SetGraphShadeColor{red!50}{black}{red}
  \Vertices{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```

9.2 \SetGraphArtColor

```
\SetGraphArtColor{<ball color>}{<color>}
```

`\SetGraphArtColor` permet de modifier les couleurs pour le style **Art**.

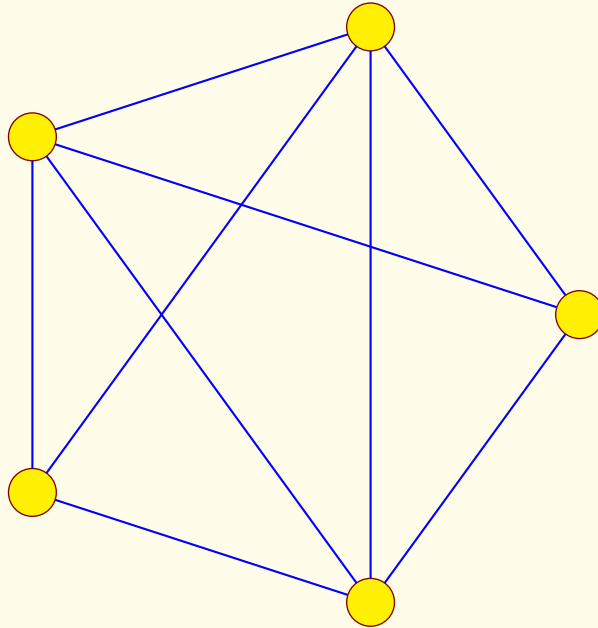
9.2.1 Exemple

```
\begin{tikzpicture}
  \SetVertexArt
  \SetGraphArtColor{green!40!black}{magenta}
  \SetGraphUnit{4}
  \SetVertexNoLabel
  \Vertices{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```


9.3 \SetGraphColor

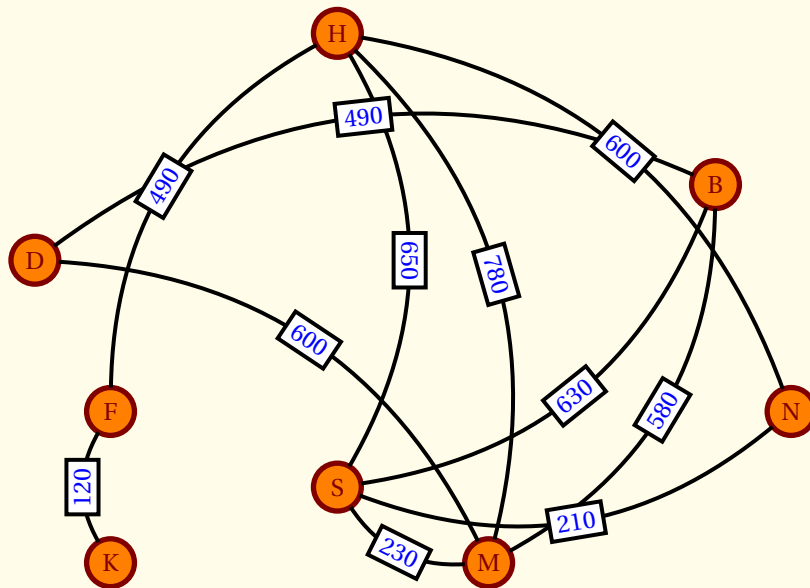
```
\SetGraphColor{<fill color>}{<color>}
```

`\SetGraphColor` permet de modifier les couleurs pour le style **Normal**.

9.3.1 Exemple avec \SetGraphColor

```
\begin{tikzpicture}
  \SetGraphColor{yellow}{blue}{maagenta}
  \SetGraphUnit{4}
  \SetVertexNoLabel
  \Vertices{circle}{A,B,C,D,E}
  \Edges(A,B,C,D,E,A,C,E,B,D)
\end{tikzpicture}
```

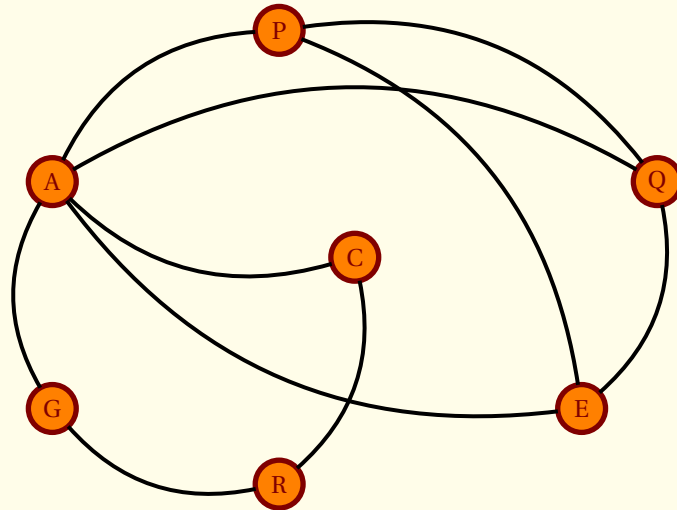
9.4 Variation I autour des styles



```

\begin{tikzpicture}
  \SetVertexNormal[Shape      = circle,
                  FillColor  = orange,
                  LineWidth  = 2pt]
  \SetUpEdge[lw      = 1.5pt,
            color    = black,
            labelcolor = white,
            labeltext = red,
            labelstyle = {sloped,draw,text=blue}]
  \Vertex[x=0 ,y=0]{K}
  \Vertex[x=0 ,y=2]{F}
  \Vertex[x=-1,y=4]{D}
  \Vertex[x=3 ,y=7]{H}
  \Vertex[x=8 ,y=5]{B}
  \Vertex[x=9 ,y=2]{N}
  \Vertex[x=5 ,y=0]{M}
  \Vertex[x=3 ,y=1]{S}
  \tikzset{EdgeStyle/.append style = {bend left}}
  \Edge[label = $120$(K)(F)
  \Edge[label = $650$(H)(S)
  \Edge[label = $780$(H)(M)
  \Edge[label = $490$(D)(B)
  \Edge[label = $600$(D)(M)
  \Edge[label = $580$(B)(M)
  \Edge[label = $600$(H)(N)
  \Edge[label = $490$(F)(H)
  \tikzset{EdgeStyle/.append style = {bend right}}
  \Edge[label = $630$(S)(B)
  \Edge[label = $210$(S)(N)
  \Edge[label = $230$(S)(M)
\end{tikzpicture}

```

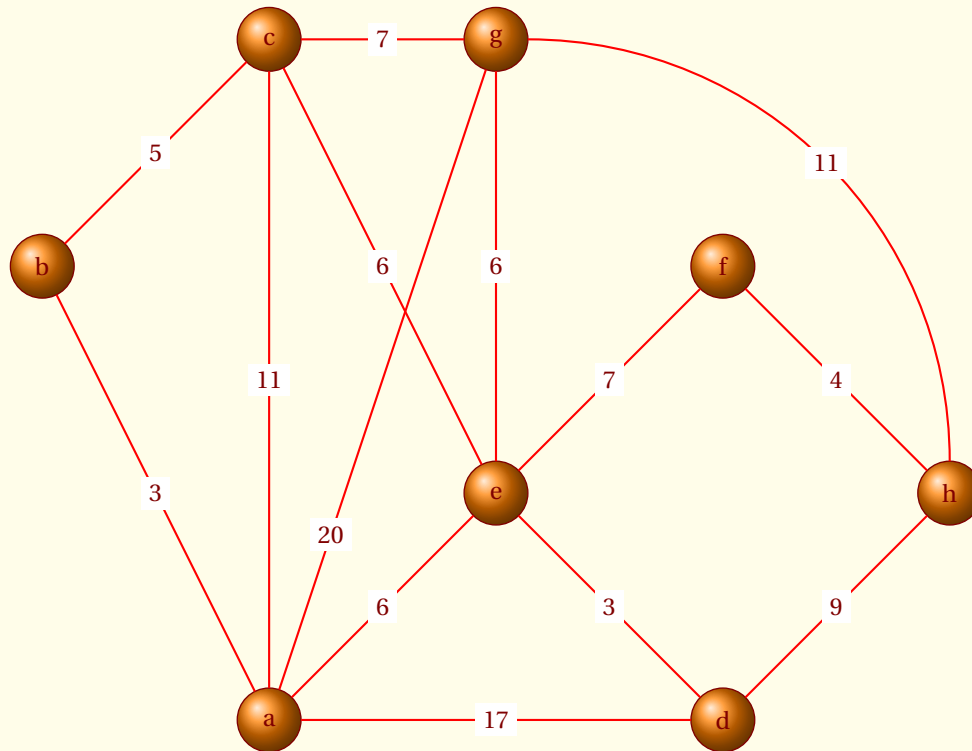
9.5 Variation II autour des styles

```

\begin{tikzpicture}
  \SetVertexNormal[Shape      = circle,
                  FillColor  = orange,
                  LineWidth  = 2pt]
  \SetUpEdge[lw      = 1.5pt,
            color    = black,
            labelcolor = white,
            labeltext = red,
            labelstyle = {sloped,draw,text=blue}]
  \tikzstyle{EdgeStyle}=[bend left]
  \Vertex[x=0, y=0]{G}
  \Vertex[x=0, y=3]{A}
  \Vertex[x=3, y=5]{P}
  \Vertex[x=4, y=2]{C}
  \Vertex[x=8, y=3]{Q}
  \Vertex[x=7, y=0]{E}
  \Vertex[x=3, y=-1]{R}
  \Edges(G,A,P,Q,E) \Edges(C,A,Q) \Edges(C,R,G) \Edges(P,E,A)
\end{tikzpicture}

```

9.6 Variation III autour des styles

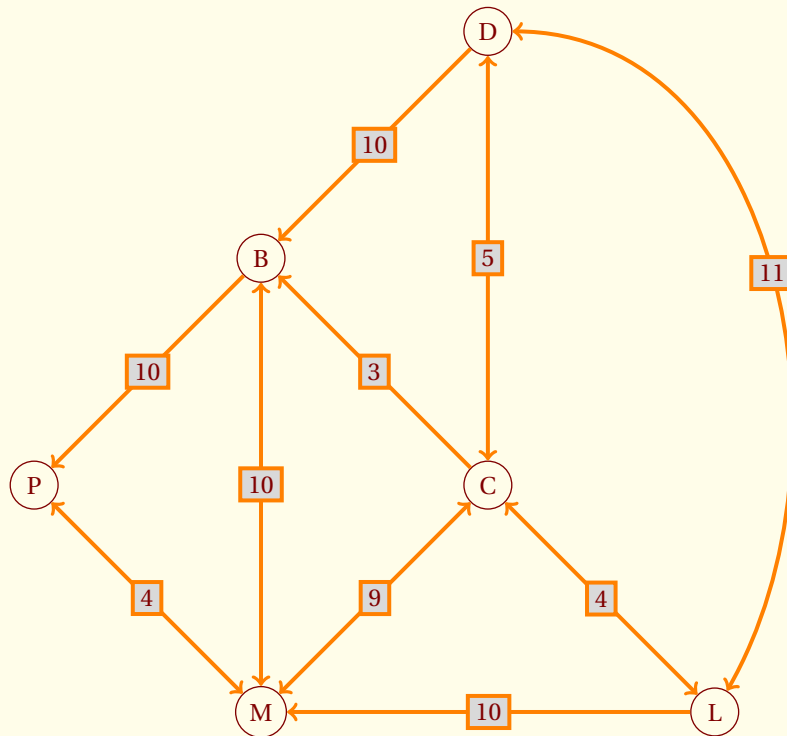


```

\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \SetGraphUnit{3}
  \Vertex{e}
  \NOEA(e){f}\SOEA(e){d}
  \SOEA(f){h}\NOWE(f){g}
  \WE(g){c} \SOWE(e){a} \SOWE(c){b}
  \tikzstyle{LabelStyle}=[fill=white]
  \tikzstyle{EdgeStyle}=[color=red]
  \Edge[label=$3$](a)(b)
  \Edge[label=$11$](a)(c)
  \Edge[label=$6$](a)(e)
  \Edge[label=$17$](a)(d)
  \Edge[style={pos=.25},label=$20$](a)(g)
  \Edge[label=$5$](c)(b)
  \Edge[label=$6$](c)(e)
  \Edge[label=$7$](c)(g)
  \Edge[label=$7$](f)(e)
  \Edge[label=$3$](d)(e)
  \Edge[label=$9$](d)(h)
  \Edge[label=$6$](g)(e)
  \Edge[style={bend left,out=45,in=135},label=$11$](g)(h)
  \Edge[label=$4$](f)(h)
\end{tikzpicture}

```

9.7 Variation IV autour des styles

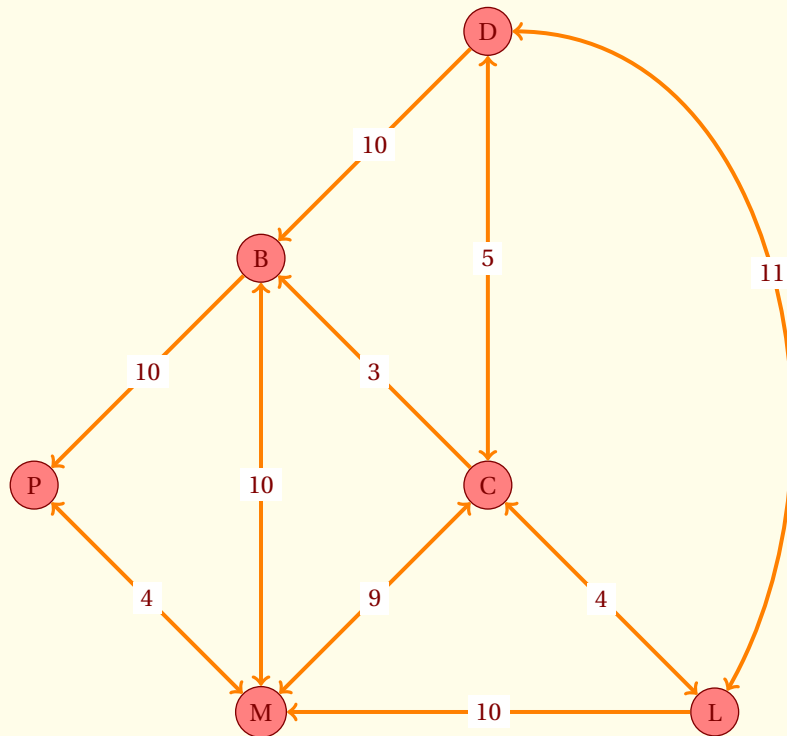


```

\begin{tikzpicture}
\SetUpEdge[lw      = 1.5pt,
           color    = orange,
           labelcolor = gray!30,
           labelstyle = {draw}]
\SetGraphUnit{3}
\GraphInit[vstyle=Normal]
\Vertex{P}
\NOEA(P){B}
\SOEA(P){M}
\NOEA(B){D}
\SOEA(B){C}
\SOEA(C){L}
\tikzset{EdgeStyle/.style={->}}
\Edge[label=$3$](C)(B)
\Edge[label=$10$](D)(B)
\Edge[label=$10$](L)(M)
\Edge[label=$10$](B)(P)
\tikzset{EdgeStyle/.style={<->}}
\Edge[label=$4$](P)(M)
\Edge[label=$9$](C)(M)
\Edge[label=$4$](C)(L)
\Edge[label=$5$](C)(D)
\Edge[label=$10$](B)(M)
\tikzset{EdgeStyle/.style={<->,relative=false,in=0,out=60}}
\Edge[label=$11$](L)(D)
\end{tikzpicture}

```

9.8 Variation V autour des styles



```

\begin{tikzpicture}

\SetUpEdge[lw      = 1.5pt,
           color    = orange,
           labelcolor = white]
\GraphInit[vstyle=Normal] \SetGraphUnit{3}
\tikzset{VertexStyle/.append style={fill = red!50}}
\Vertex{P}
\NOEA(P){B} \SOEA(P){M} \NOEA(B){D}
\SOEA(B){C} \SOEA(C){L}
\tikzset{EdgeStyle/.style={->}}
\Edge[label=$3$](C)(B)
\Edge[label=$10$](D)(B)
\Edge[label=$10$](L)(M)
\Edge[label=$10$](B)(P)
\tikzset{EdgeStyle/.style={<->}}
\Edge[label=$4$](P)(M)
\Edge[label=$9$](C)(M)
\Edge[label=$4$](C)(L)
\Edge[label=$5$](C)(D)
\Edge[label=$10$](B)(M)
\tikzset{EdgeStyle/.style={<->,relative=false,in=0,out=60}}
\Edge[label=$11$](L)(D)
\end{tikzpicture}

```

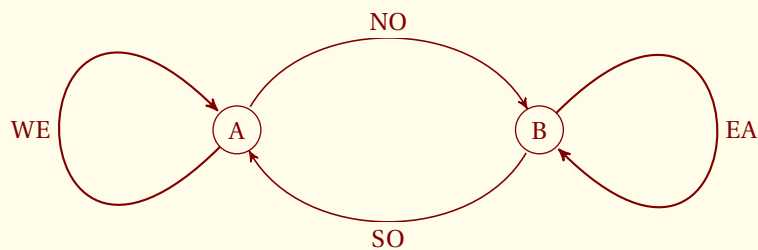
Graphes probabilistes

10.1 La macro `\grProb`

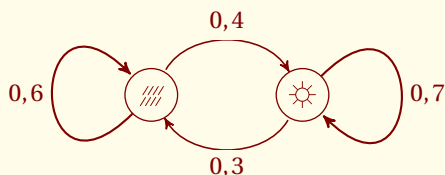
```
\grProb[local options] {<left>} {<right>} {<N>}{<S>}{<W>}{<E>}
```

Arguments	Définition	
Vertex-left	Nom du sommet à gauche	
Vertex-right	Nom du sommet à droite	
label N	Étiquette située en haut	
label S	Étiquette située en bas	
label W	Étiquette située à gauche	
label E	Étiquette située à droite	
options	défaut	définition
unit	4cm	distance entre les sommets
LposA	180	angle si label extérieur en A
LposB	0	angle si label extérieur en B
Ldist	0cm	écart entre le node et le label
LoopDist	4cm	longueur des boucles

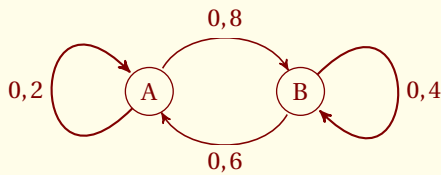
Cette macro permet de créer un graphe probabiliste d'ordre 2.

10.2 Utilisation de `\grProb`

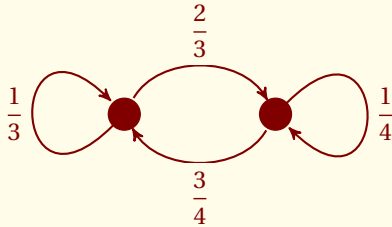
```
\begin{tikzpicture}
  \useasboundingbox (-2.5,-2) rectangle (7.5,2);
  \grProb{A}{B}{NO}{SO}{WE}{EA}
\end{tikzpicture}
```



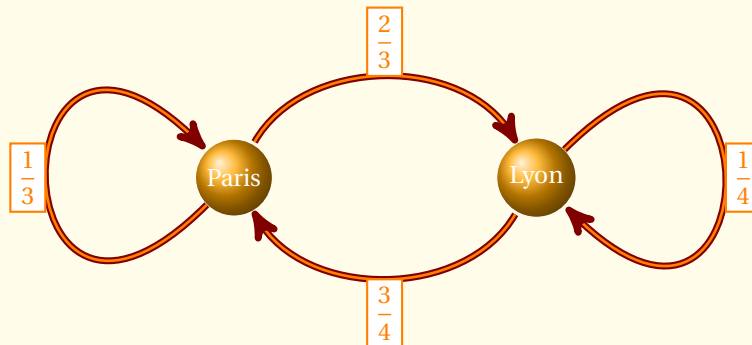
```
\begin{tikzpicture}[scale=.5]
  \useasboundingbox (-2.5,-2) rectangle (5,2);
  \grProb[unit=4]{\Rain}{\Sun}{0,4}{0,3}{0,6}{0,7}
\end{tikzpicture}
```

10.3 \grProb et le style par défaut

```
\begin{tikzpicture}[scale=.5]
  \useasboundingbox (-2.5,-2) rectangle (5,2);
  \grProb{A}{B}{0,8}{0,6}{0,2}{0,4}
\end{tikzpicture}
```

10.4 \grProb et le style « Simple »

```
\begin{tikzpicture}[scale=.5]
  \useasboundingbox (-2.5,-2) rectangle (5,2);
  \SetVertexSimple
  \grProb[Ldist=0.2cm]{Paris}{Lyon}%
    {\scriptstyle\frac{2}{3}}{\scriptstyle\frac{3}{4}}%
    {\scriptstyle\frac{1}{3}}{\scriptstyle\frac{1}{4}}%
\end{tikzpicture}
```

10.5 Utilisation d'un style personnalisé

```
\begin{tikzpicture}
  \useasboundingbox (-2.5,-2.5) rectangle (7.5,2.5);
  \tikzset{VertexStyle/.style = {shape = circle,
    shading = ball,
    ball color = Orange,
    minimum size = 20pt,
    draw,color=white}}
  \tikzset{LabelStyle/.style = {draw,color=orange,fill=white}}
  \tikzset{EdgeStyle/.style = {->, thick,
    double = orange,
    double distance = 1pt}}

  \grProb[Ldist=0.1cm,LposA=0,LposB=180]%
    {Paris}{Lyon}%
    {\scriptstyle\frac{2}{3}}{\scriptstyle\frac{3}{4}}%
    {\scriptstyle\frac{1}{3}}{\scriptstyle\frac{1}{4}}%
\end{tikzpicture}
```


10.6 La macro `\grProbThree`

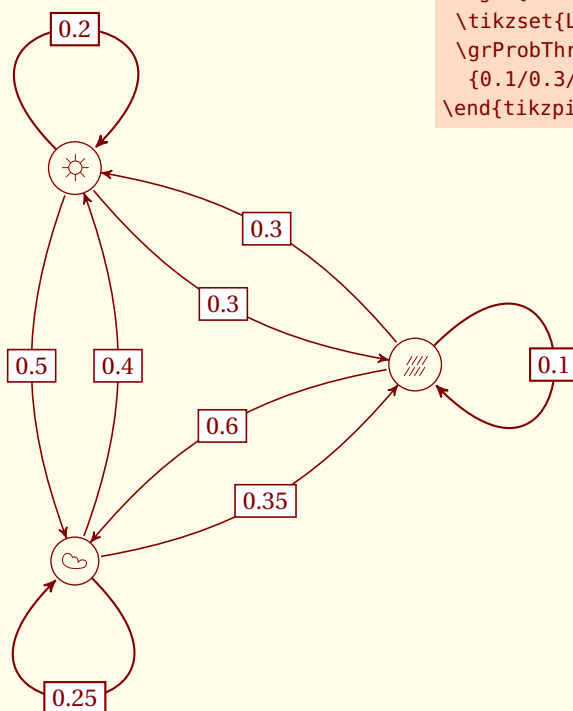
```
\grProbThree[⟨local options⟩] {⟨right⟩} {⟨up⟩}{⟨down⟩} {⟨rr/ru/rd⟩}{⟨uu/ud/ur⟩}{⟨dd/dr/du⟩}
```

Arguments		Définition
Vertex-right		Nom du sommet à droite
Vertex-up		Nom du sommet en haut
Vertex-down		Nom du sommet en bas
rr/ru/rd		arête partant de r vers r etc...
uu/ud/ur		arête partant de u vers u etc...
dd/dr/du		arête partant de d vers d etc...

Options	Défaut	Définition
unit	4cm	distance entre les sommets
LposA	180	angle si label extérieur en A
LposB	0	angle si label extérieur en B
Ldist	0cm	écart entre le node et le label
LoopDist	4cm	longueur des boucles

Cette macro permet de créer un graphe probabiliste d'ordre 3.

10.6.1 Graphe probabiliste d'ordre 3



```
\begin{tikzpicture}[scale=.75]
  \tikzset{LabelStyle/.style = {draw,fill=white}}
  \grProbThree[unit=4]{\Rain}{\Sun}{\Cloud}
  {0.1/0.3/0.6}{0.2/0.3/0.5}{0.25/0.35/0.4}
\end{tikzpicture}
```

Colorisation Welsh

Ce chapitre montre comment colorer des sommets. Le plus simple est d'utiliser le style `Welsh` et la macro `\AddVertexColor` afin de colorer les sommets.

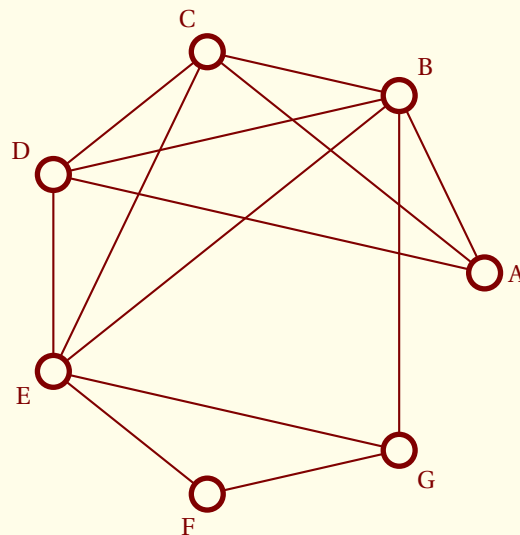
11.1 La macro `\AddVertexColor`

```
\AddVertexColor{<color>}{<List of vertices>}
```

Cette macro permet de colorer des sommets. Le premier argument est la couleur, le second une liste de sommets.

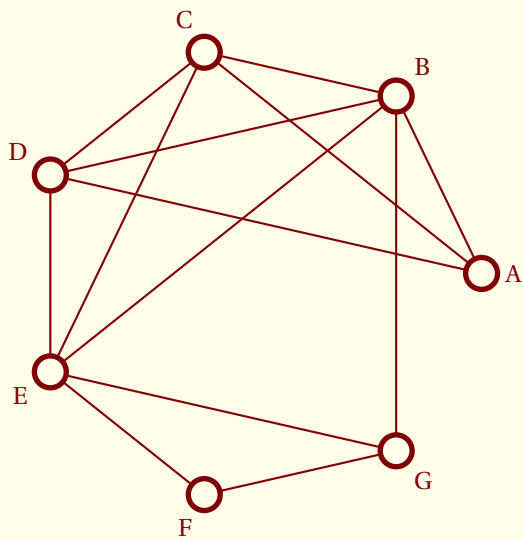
11.2 Exemple d'utilisation

Une compagnie aérienne propose des vols directs entre certaines villes, notées A, B, C, D, E, F et G. Cela conduit au graphe \mathcal{G} suivant, dont les sommets sont les villes et les arêtes représentent les liaisons aériennes :



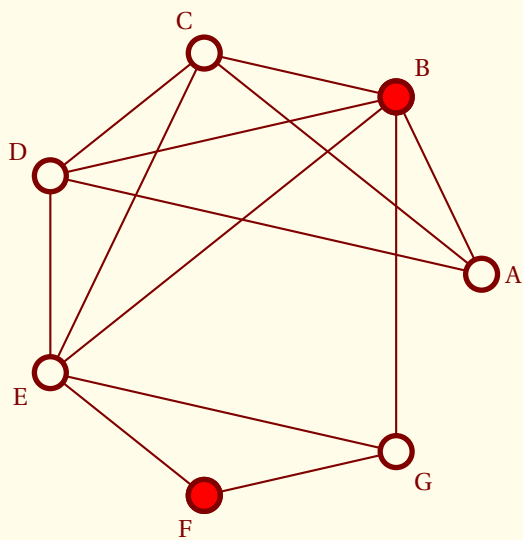
1. Sur les cartes d'embarquement, la compagnie attribue à chaque aéroport une couleur, de sorte que deux aéroports liés par un vol direct aient des couleurs différentes.
Proposer un coloriage adapté, cette condition.
2. Que peut-on en déduire sur le nombre chromatique de \mathcal{G} ?

```
\begin{tikzpicture}
  \renewcommand*{\VertexLineWidth}{2pt}
  \GraphInit[vstyle=Welsh]
  \Vertices[unit=3]{circle}{A,B,C,D,E,F,G}
  \Edges(G,E,F,G,B,D,E,C,D,A,C,B,A) \Edges(B,E)
\end{tikzpicture}
```



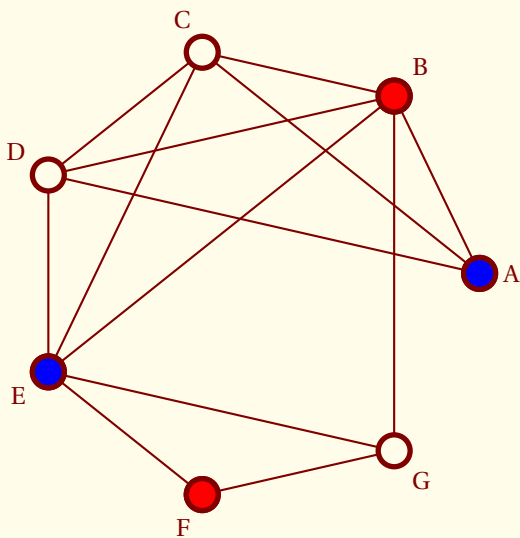
<i>Sommet</i>	<i>Degré</i>
B	5
E	5
C	4
D	4
A	3
G	3
F	2

```
\begin{tikzpicture}
\renewcommand*{\VertexLineWidth}{2pt}
\GraphInit[vstyle=Welsh]
\Vertices[unit=3]{circle}{A,B,C,D,E,F,G}
\Edges(G,E,F,G,B,D,E,C,D,A,C,B,A) \Edges(B,E)
\end{tikzpicture}
```



<i>Sommet</i>	<i>Degré</i>	<i>Couleur</i>
B	5	rouge
E	5	
C	4	
D	4	
A	3	
G	3	
F	2	rouge

```
\begin{tikzpicture}
\renewcommand*{\VertexLineWidth}{2pt}
\GraphInit[vstyle=Welsh]
\Vertices[unit=3]{circle}{A,B,C,D,E,F,G}
\SetVertexNoLabel
\AddVertexColor{red}{B,F}
\Edges(G,E,F,G,B,D,E,C,D,A,C,B,A) \Edges(B,E)
\end{tikzpicture}
```

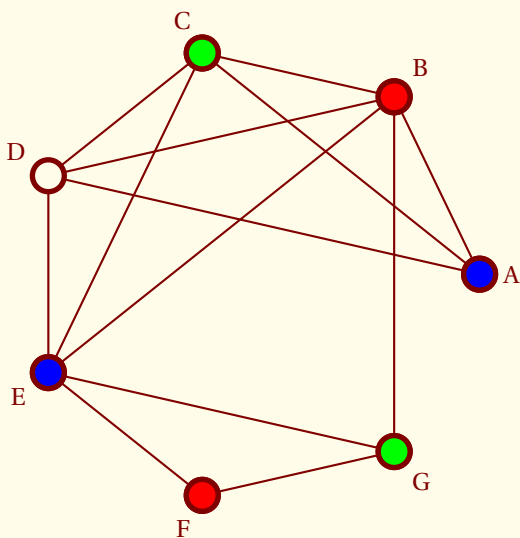


<i>Sommet</i>	<i>Degré</i>	<i>Couleur</i>
B	5	rouge
E	5	bleu
C	4	
D	4	
A	3	bleu
G	3	
F	2	rouge

```

\begin{tikzpicture}
  \renewcommand*{\VertexLineWidth}{2pt}
  \GraphInit[vstyle=Welsh]
  \Vertices[unit=3]{circle}{A,B,C,D,E,F,G}
  \SetVertexNoLabel
  \AddVertexColor{red}{B,F}
  \AddVertexColor{blue}{E,A}
  \Edges(G,E,F,G,B,D,E,C,D,A,C,B,A)
  \Edges(B,E)
\end{tikzpicture}

```

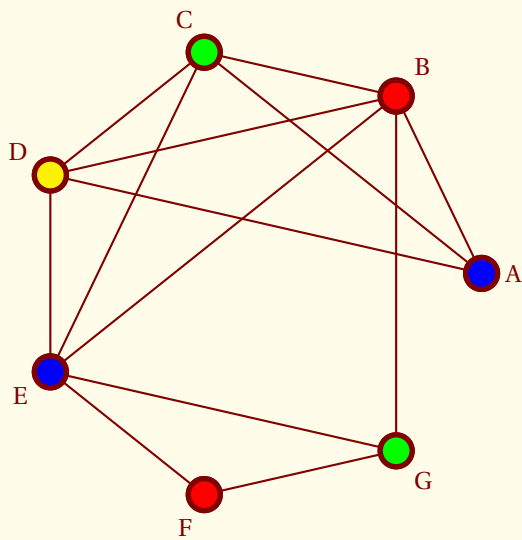


<i>Sommet</i>	<i>Degré</i>	<i>Couleur</i>
B	5	rouge
E	5	bleu
C	4	vert
D	4	
A	3	bleu
G	3	vert
F	2	rouge

```

\begin{tikzpicture}
  \renewcommand*{\VertexLineWidth}{2pt}
  \GraphInit[vstyle=Welsh]
  \Vertices[unit=3]{circle}{A,B,C,D,E,F,G}
  \SetVertexNoLabel
  \AddVertexColor{red}{B,F} \AddVertexColor{blue}{E,A}
  \AddVertexColor{green}{C,G}
  \Vertex[Node]{D}
  \Edges(G,E,F,G,B,D,E,C,D,A,C,B,A) \Edges(B,E)
\end{tikzpicture}

```



<i>Sommet</i>	<i>Degré</i>	<i>Couleur</i>
B	5	rouge
E	5	bleu
C	4	vert
D	4	jaune
A	3	bleu
G	3	vert
F	2	rouge

```

\begin{tikzpicture}
  \renewcommand*{\VertexLineWidth}{2pt}
  \GraphInit[vstyle=Welsh]
  \Vertices[unit=3]{circle}{A,B,C,D,E,F,G}
  \SetVertexNoLabel
  \AddVertexColor{red}{B,F}  \AddVertexColor{blue}{E,A}
  \AddVertexColor{green}{C,G}\AddVertexColor{yellow}{D}
  \Vertex[Node]{D}
  \Edges(G,E,F,G,B,D,E,C,D,A,C,B,A)\Edges(B,E)
\end{tikzpicture}

```

Index

A

<code>\AddVertexColor{<i>color</i>}{<i>List of vertices</i>}</code>	50
<code>\AddVertexColor</code>	5, 50

E

<code>\EA</code>	5, 14
<code>\Edge</code>	5, 24, 35
<code>\Edge: options</code>	
color.....	24
labelcolor.....	24
labelstyle.....	24
labeltext.....	24
label.....	24
local.....	24
lw.....	24
style.....	24
<code>\Edges</code>	5, 9, 26
<code>\Edges: options</code>	
color.....	26
labelstyle.....	26
label.....	26
lw.....	26
style.....	26

G

<code>\GraphInit</code>	9, 27
<code>\GraphInit: options</code>	
vstyle.....	27
<code>\GraphInit[<i>local options</i>]</code>	27
<code>\grProb</code>	5, 47, 48
<code>\grProb: arguments</code>	
Vertex-left.....	47
Vertex-right.....	47
label E.....	47
label N.....	47
label S.....	47
label W.....	47
<code>\grProb: options</code>	
Ldist.....	47
LoopDist.....	47
LposA.....	47
LposB.....	47
unit.....	47
<code>\grProbThree</code>	49
<code>\grProbThree: arguments</code>	
Vertex-down.....	49
Vertex-right.....	49
Vertex-up.....	49
dd/dr/du.....	49
rr/ru/rd.....	49
uu/ud/ur.....	49
<code>\grProbThree: options</code>	
Ldist.....	49
LoopDist.....	49

LposA.....	49
LposB.....	49
unit.....	49
\grProbThree[<i>{local options}</i>] <i>{<right>}</i> <i>{<up>}{<down>}</i> <i>{<rr/ru/rd>}{<uu/ud/ur>}{<dd/dr/du>}</i>	49
\grProb[<i>{local options}</i>] <i>{<left>}</i> <i>{<right>}</i> <i>{<N>}{<S>}{<W>}{<E>}</i>	47

L

\Loop.....	5, 25
\Loop: options	
color.....	25
labelstyle.....	25
label.....	25
lw.....	25
style.....	25

N

\NO.....	5, 14
\NOEA.....	5, 9, 14
\NOWE.....	5, 14

P

Package	
tkz-berge.....	39

S

\SetGraphArtColor{ <i>{ball color}</i> }{ <i>{color}</i> }.....	40
\SetGraphArtColor.....	5, 40
\SetGraphColor{ <i>{fill color}</i> }{ <i>{color}</i> }.....	41
\SetGraphColor.....	5, 41
\SetGraphShadeColor{ <i>{ball color}</i> }{ <i>{color}</i> }{ <i>{double}</i> }.....	39
\SetGraphShadeColor.....	5, 39
\SetGraphUnit.....	15
\SetGraphUnit{2}.....	14
\SetGraphUnit{ <i>{nombre}</i> }.....	14
\SetGraphUnit.....	14, 17, 18
\SetUpEdge.....	5, 27, 36, 37
\SetUpEdge: options	
color.....	36
labelstyle.....	36
labeltext.....	36
label.....	36
lw.....	36
style.....	36
\SetUpEdge[<i>{local options}</i>].....	36
\SetUpVertex.....	5, 27, 34
\SetUpVertex: options	
LabelOut.....	34
Ldist.....	34
Lpos.....	34
NoLabel.....	34
style.....	34
\SetUpVertex[<i>{local options}</i>].....	34
\SetVertexLabel.....	22
\SetVertexLabelIn.....	5, 23
\SetVertexLabelOut.....	5, 23
\SetVertexMath.....	5, 22
\SetVertexNoLabel.....	22

<code>\SetVertexNoMath</code>	5
<code>\SetVertexNormal</code>	27, 33
<code>\SetVertexNormal: options</code>	
<code>color</code>	33
<code>labelcolor</code>	33
<code>labelstyle</code>	33
<code>labeltext</code>	33
<code>label</code>	33
<code>lw</code>	33
<code>style</code>	33
<code>\SetVertexNormal[<i>local options</i>]</code>	33
<code>\SetVertexSimple</code>	27, 31, 32
<code>\SetVertexSimple: options</code>	
<code>FillColor</code>	32
<code>LineColor</code>	32
<code>LineWidth</code>	32
<code>MinSize</code>	32
<code>Shape</code>	32
<code>\SetVertexSimple[<i>local options</i>]</code>	32
<code>\ShortCut</code>	14
<code>\SO</code>	5, 14
<code>\SOEA</code>	5, 14
<code>\SOWE</code>	5, 14
Systeme d'exploitation	
Linux Ubuntu.....	6
OS X.....	6
Windows XP.....	7

T

TeX Distributions	
MikTeX.....	7
TeXLive.....	6
TikZ.....	6
<code>\tikzset{VertexStyle/.append style = { ... }}</code>	27
<code>\tikzset{VertexStyle/.style = { ... }}</code>	27
<code>\tikzset</code>	34

V

<code>\Vertex[a=<i>number</i>,d=<i>number</i>]{<i>vertex</i>}</code>	13
<code>\Vertex[x=<i>number</i>,y=<i>number</i>]{<i>name</i>}</code>	12
<code>\Vertex</code>	5, 12, 14
<code>\Vertex: arguments</code>	
<code>Name</code>	12
<code>\Vertex: options</code>	
<code>LabelOut</code>	12
<code>Ldist</code>	12
<code>Lpos</code>	12
<code>L</code>	12
<code>Math</code>	12
<code>NoLabel</code>	12
<code>Node</code>	12
<code>a</code>	12
<code>dir</code>	12
<code>d</code>	12
<code>empty</code>	12
<code>position</code>	12
<code>x</code>	12

y.....	12
\Vertex[<i>{local options}</i>]{ <i>{Name}</i> }.....	12
\Vertices.....	5, 9, 17
\Vertices: arguments	
circle.....	17
line.....	17
square.....	17
tr1.....	17
tr2.....	17
tr3.....	17
tr4.....	17
\Vertices: options	
dir.....	17
\Vertices[<i>{local options}</i>]{ <i>{type}</i> }{ <i>{List of vertices}</i> }.....	17
W	
\WE.....	5, 14
X	
xkeyval.....	6