

```

1  /*
2  -----
3  Nom du fichier      : Galton.cpp
4  Nom du labo        : Labo08_Galton Groupe L
5  Auteur(s)          : Jeremiah Steiner, Ylli Fazlija
6  Date creation       : 11.01.2022
7  Description (But)   : Définition de la classe Galton. Contient les différentes
8                        fonctions utiles à l'utilisation d'un plateau.
9  Remarque(s)        :
10 Compileur           : Mingw-w64 g++ 8.1.0
11 -----
12 */
13
14
15 #include <iostream>
16 #include <algorithm> // Utilisé pour trouver le plus grand élément dans un tableau.
17 #include "Galton.h"
18
19 using namespace std;
20
21 // Création de la seed pour le random.
22 std::random_device Galton::rd;
23
24 /// Constructeur paramétrique de l'objet Galton
25 /// \param nbBille le nombre de billes à lancer dans la planche
26 /// \param h la hauteur de la planche
27 Galton::Galton(const unsigned nbBille, const unsigned h) :
28     nbrDeBilles(nbBille),
29     hauteur(h)
30 {
31     tableauBilles.resize(h + 1); // on s'assure d'avoir la place
32
33     // Remplissage du tableau de billes.
34     LancerBilles();
35 }
36
37 /// Constructeur par copie de la classe Galton
38 /// Il crée une nouvelle instance à partir des valeurs d'une autre instance et
39 /// lance une nouvelle fois la simulation. Les résultats de celle-ci ne seront
40 /// donc pas les mêmes.
41 /// \param g instance à copier.
42 Galton::Galton(Galton& g) :
43     nbrDeBilles(g.nbrDeBilles),
44     hauteur(g.hauteur)
45 {
46     tableauBilles.resize(g.hauteur + 1); // on s'assure d'avoir la place
47     LancerBilles();
48 }
49
50 /// Cette fonction permet de remplir le tableau de billes des valeurs
51 /// On lance chaque bille et on détermine, en utilisant un générateur de
52 /// chiffres aléatoire, son emplacement.
53 void Galton::LancerBilles()
54 {
55     std::mt19937 gen(rd()); // Moteur d'aléatoire qui utilise le seed rd.
56
57     // On règle notre moteur d'aléatoire
58     // pour nous donner des chiffres aléatoires entre 0 et 1.
59     std::uniform_int_distribution<> distrib(0, 1);
60
61     size_t indice;
62
63     // Boucle qui itère sur le nombre de billes.
64     // Chaque itération correspond à une bille.
65     for(size_t i = 0; i < nbrDeBilles; ++i) {
66         indice = 0;
67
68         // Deuxième boucle qui itère sur la hauteur.
69         // à chaque itération, deux issues sont possibles : droite ou gauche.
70         // Si distrib retourne 1, on ajoute 1 à indice.
71         for (size_t n = 0; n < hauteur; ++n) {
72             indice += distrib(gen);
73         }
74
75         // Une fois la deuxième boucle terminée, on a trouvé l'indice de
76         // la case dans laquelle la bille tombe.

```

```
77     tableauBilles[indice]++;
78 }
79 }
80
81 /// Permits d'afficher une représentation graphique du tableau
82 /// à l'aide d'une double boucle.
83 /// \param CARACTERE Le caractère qui représentera une bille.
84 /// \param ESPACE Le caractère qui représentera les espaces.
85 void Galton::AfficherTableauGraphique(const char& CARACTERE,
86                                       const char& ESPACE) const
87 {
88     for (size_t i = ((size_t) *max_element(tableauBilles.begin(),
89                                           tableauBilles.end())); i > 0; --i)
89     {
90         for (size_t val : tableauBilles)
91         {
92             cout << (val >= i ? CARACTERE : ESPACE);
93         }
94         cout << endl;
95     }
96 }
97 }
98
99 /// Simple getter afin d'accéder au tableau de billes d'une instance Galton.
100 vector<int> Galton::getTableauBilles() const {
101     return tableauBilles;
102 }
103
```