

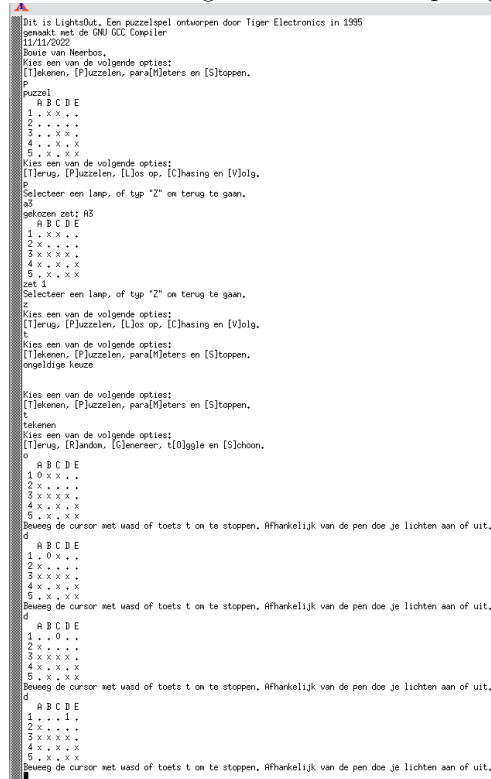
# Programmeeropgave 3: Lights out

Bowie van Neerbos - s2033755

11 januari 2023

## 1 Uitleg

Het spel Lights out is een puzzelspel waarbij het doel is alle lampen uit te doen. Echter gaan bij het omzetten van de schakelaar van een lamp ook nog zijn omstanders om [2]! Deze code simuleert dit spel in een terminal. bij het starten van de code wordt de gebruiker begroet en gepresenteerd met een menu. Een kort overzicht van alle opties: Parameters aanpassen: Zo kunnen worden aangepast: de gebruikte symbolen, de afmetingen van het bord, de vorm van het bord en de pen gebruikt bij het tekenen en het percentage lampen dat aangaat als een nieuw bord wordt aangemaakt. Teken: Hier kan er handmatig een bord uitgetekend worden met een bewegende cursor, een random bord aangemaakt worden (geheel willekeurig of door middel van het zetten van een aantal stappen, dit aantal is dan de moeilijkheid) of het bord kan leeggemaakt worden. Puzzelen: Met de opties; puzzelen op het huidige bord (het echte spel dus), het volg-algoritme (waarbij alleen de laatste rij onopgelost blijft) en het oplossen van de puzzel (op twee manieren, namelijk door de lampen weer in omgekeerde volgorde om te schakelen, en door het zogenaamde light chasing [1]). Ik heb het programma gecheckt op veel punten, en zo veel mogelijk bugs weggewerkt. Hier en daar gebruik ik memberfuncties waar ik beter normale functies had kunnen gebruiken en vice versa, maar dit schaadt de werking van de code niet. Ik heb mijn tijd vooral geïnvesteerd in de werking van de code. Een plaatje van de interface:



```
Dit is LightsOut. Een puzzelspel ontworpen door Tiger Electronics in 1995
gemaakt met de GNU GCC Compiler
11/11/2022
Bowie van Neerbos.
Kies een van de volgende opties:
[T]ekenen, [P]uzzelen, para[M]eters en [S]toppen.
p
puzzel
1 . X X . .
2 . . . . .
3 . . X X .
4 . . X X .
5 . X X X X
Kies een van de volgende opties:
[T]erug, [P]uzzelen, [L]os op, [C]hasing en [V]olg.
p
Selecteer een lamp, of typ "Z" om terug te gaan.
a5
gekozen zet: A5
1 . X X . .
2 X . . . .
3 X X X X .
4 X . X X X
5 . X X X X
zet 1
Selecteer een lamp, of typ "Z" om terug te gaan.
z
Kies een van de volgende opties:
[T]erug, [P]uzzelen, [L]os op, [C]hasing en [V]olg.
t
Kies een van de volgende opties:
[T]ekenen, [P]uzzelen, para[M]eters en [S]toppen.
ongeldige keuze
Kies een van de volgende opties:
[T]ekenen, [P]uzzelen, para[M]eters en [S]toppen.
t
tekenen
Kies een van de volgende opties:
[T]erug, [R]andom, [C]reëer, t[O]ggel en [S]choon.
o
1 0 X X . .
2 X . . . .
3 X X X X .
4 X . X X X
5 . X X X X
Beweg de cursor met waad of toets t om te stoppen, Afhankelijk van de pen doe je lichten aan of uit.
d
1 0 X X . .
2 X . . . .
3 X X X X .
4 X . X X X
5 . X X X X
Beweg de cursor met waad of toets t om te stoppen, Afhankelijk van de pen doe je lichten aan of uit.
d
1 . . . . .
2 X . . . .
3 X X X X .
4 X . X X X
5 . X X X X
Beweg de cursor met waad of toets t om te stoppen, Afhankelijk van de pen doe je lichten aan of uit.
d
1 . . . . .
2 X . . . .
3 X X X X .
4 X . X X X
5 . X X X X
Beweg de cursor met waad of toets t om te stoppen, Afhankelijk van de pen doe je lichten aan of uit.
```

## 2 Tijd

datum	uren	wat
20-10	2	menu en leesgetal
27-10	4	de wereld vullen en printen random en leeg klassen, memberfuncties etc.
3-11	3	niks, mental block
8-11	5	basisspel werkt, parameter menu klaar.
10-11	3.5	alles behalve toggle functie af
11-11	4.5	code af en verslag.
totaal	22	

## 3 opmerkingen

Allereerst, het gebruik van de random generator zoals gepresenteerd in het collegedictaat leidde steeds tot een voorspelbare set puzzel permutaties. Bij zowel `maakRandom` als `genereer` waren er slechts een tiental mogelijke puzzels. Deze methode is aangepast om het puzzelen leuk en onvoorspelbaar te houden.

opmerkingen in de code met een asterisk "\*" zijn aangepast door `randomgetal()` te vermenigvuldigen met `rand()`. Ook wordt aan het begin van de code `srand(time(0))` gebruikt, zodat elke run van het programma andere getallen oplevert. Tot slot is in `randomgetal()` de "static long getal" random gegenereerd, in plaats van een vaste waarde. Als deze aanpassingen ongedaan worden gemaakt, worden de random getallen gegenereerd zoals de bedoeling was, maar wordt het puzzelen ook een stuk minder leuk. Ook is de layout van het bord bij mij iets anders: letters boven en getallen links. Dit omdat dit voor mij natuurlijker aanvoelde bij het spelen.

Dit zijn voor zover ik mij kan herinneren de enige punten waar ik afwijk van de opdracht.

## Referenties

- [1] Matthew Baker. Solving lights out.
- [2] Tony Delgado. 'beyond tetris' - lights out. 2007.

# Code

Dit is de volledige code:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib>
4  #include <stdlib.h>      /* srand, rand */
5  #include <time.h>
6
7
8
9  using namespace std;
10
11 //klassen=====
12
13 class puzzel
14 {
15 public:
16     puzzel(int hoogteWaarde, int breedteWaarde, char aanKeuze,
17           char uitKeuze, int percentKeuze, bool torusKeuze, int penKeuze);
18
19     puzzel();
20     int hoogte, breedte;
21     char aankarakter, uitkarakter; //de symbolen voor de lampen
22     bool torus; //verbindt de linker- met de rechterkant vd puzzel; een ring
23     bool gegenereerd; //houdt bij op welke manier de puzzel is gegenereerd
24     int percentage; // het percentage lampen dat aangaat bij maakRandom
25     int pen; //het gedrag van de pen bij tekenen (0,1,2)
26     int moeilijkheid; //hoe veel zetten worden gedaan bij genereer
27
28     void maakSchoon();
29     void drukAf();
30     void maakWereld();
31     void maakRandom(); //gegenereerd = false
32     void genereer(); //gegenereerd = true
33     void volg();
34     void laatsteRij();
35     void losop();
36     void zet(int, int);
37     void eenlamp(int, int);
38     bool check();
39     bool zetofniet(int, int);
40     bool cursor[26][26];
41 private:
42     char deWereld[26][26]; //max 26, zodat we in ieder geval niet voorbij z gaan
43     bool oplossing[26][26];
44     void binnenDeGrenzen();
45 };
46
47 //functies=====
48
49 void infoblok(){
50     cout << "Dit is LightsOut. Een puzzelspel ontworpen door "
51     << "Tiger Electronics in 1995" << endl
52     << "gemaakt met de GNU GCC Compiler" << endl << "11/11/2022" << endl
53     << "Bowie van Neerbos." << endl;
```

```

54 }//infoblok
55
56 // Pseudo-random-getal tussen 0 en 999: aangepast van dictaat.
57 long randomgetal ( ) {
58     static long getal = rand()%100; //rand() zodat niet elke run hetzelfde doet
59     getal = ( 221 * getal + 1 ) % 10000;
60     return getal;
61 } // randomgetal
62
63 //returnt de eerste niet-enter. bij herhaald aanroepen geeft hij
64 //opeenvolgende losse chars uit de invoer
65 char eerste(){
66     char invoer = '\n';
67     while (invoer == '\n'){
68         invoer = cin.get();
69     }//while
70     return invoer;
71 }//eerste
72
73 //geeft van een invoer (bijv "/n 6fgdh73sjiod30) een getal tussen minimum
74 //en maximum.
75 int leesGetal(int minimum, int maximum){
76     int getal = 0;
77     char invoer = eerste(); //skipt de enters
78     while (invoer != '\n'){
79         if ( '0' <= invoer && invoer <= '9' ){
80             getal = 10 * getal + (invoer - '0');
81             if (getal > maximum){getal = maximum;}
82         }//if
83         invoer = cin.get();
84     }//while
85     if (getal < minimum){getal = minimum;}
86     return getal;
87 }//leesGetal
88
89 //checkt of de gekozen karakters voldoen (geen 0 of 1, en ongelijk aan elkaar)
90 void checkKarakter(char &aan, char &uit){
91     if (aan == uit || aan == '1' || aan == '0' || uit == '1' || uit == '0'){
92         aan = 'x'; uit = '.';
93         cout << "Ongeldige invoer. Standaardkarakters worden gebruikt" << endl;
94     }//if
95 }//checkKarakter
96
97 //doet eigenlijk niks, een zogenaamde easter egg
98 void lebowski(){
99     cout << "Well, that's just, like, your opinion, man..." << endl;
100 }//lebowski
101
102 //klapt een enkele lamp om in de puzzel
103 void toggle(char &lamp){
104     if (lamp == 'x'){lamp = 'o';}
105     else if (lamp == 'o'){lamp = 'x';}
106 }//toggle
107
108 //klapt een enkele waarde om in de oplossing
109 void booltoggle(bool &lamp){

```

```

110     if (lamp){lamp = false;}
111     else {lamp = true;}
112 }//booltoggle
113
114 //lost de puzzel op mbv de oplossing, mits beschikbaar
115 void losop(puzzel &mijnpuzzel){
116     int i, j;
117     for (i=1; i < mijnpuzzel.hoogte; i++){
118         for (j=1; j < mijnpuzzel.breedte; j++){
119             if (mijnpuzzel.zetofniet(i,j)){//is op deze positie een zet gedaan?
120                 mijnpuzzel.zet(i, j);//draai deze dan terug
121                 mijnpuzzel.drukAf();
122                 cout << endl;
123             }//if
124         }//for
125     }//for
126 }//losop
127
128 //het tekenen
129 void tekenen(puzzel &mijnpuzzel){
130     int i = 1; int j = 1;
131     mijnpuzzel.cursor[i][j]=true; mijnpuzzel.drukAf();
132     char richting;
133     bool stoppen = false;
134     while(!stoppen){
135         cout << "Beweeg de cursor met wasd of toets t om te stoppen. "
136         << "Afhankelijk van de pen doe je lichten aan of uit." << endl;
137         if (richting != 't' && richting != 'T'){
138             mijnpuzzel.cursor[i][j] = false;
139             richting = eerste();
140             if ((richting == 's' || richting == 'S')
141                 && i != mijnpuzzel.hoogte - 1){i++;}
142             if ((richting == 'w' || richting == 'W')
143                 && i != 1){i--;}
144             if ((richting == 'd' || richting == 'D')
145                 && j != mijnpuzzel.breedte - 1){j++;}
146             if ((richting == 'a' || richting == 'A')
147                 && j != 1){j--;}
148             mijnpuzzel.cursor[i][j]=true; mijnpuzzel.eenlamp(i, j);
149             mijnpuzzel.drukAf();
150         }
151         else{stoppen = true;}//t getypt, terug naar menu
152     }
153     mijnpuzzel.cursor[i][j]=false; // haalt de cursor weer van de puzzel
154 }
155
156 //doe zetten tot de puzzel is opgelost, of stop eerder met Z
157 bool puzzelen(puzzel &mijnpuzzel){
158     int letter, getal;
159     static int zetten = 0;
160     bool stoppen = false;
161     while (!stoppen){
162         bool foutje = false; //voert geen zet uit als er een invoerfout is
163         cout << "Selecteer een lamp, of typ \"Z\" om terug te gaan." << endl;
164         letter = eerste() - 'a' + 1; //geeft de kolom
165         if (letter == ('Z' - 'a' + 1) || letter == ('z' - 'a' + 1))

```

```

166     {stoppen = true; return false;}
167     if (letter >= mijnpuzzel.breedte || letter <= 0){
168         letter = letter + 'a' - 'A';
169         if (letter >= mijnpuzzel.breedte || letter <= 0){
170             cout << "Ongeldige invoer. Voer bestaande coördinaten in. "
171             << " Deze staan om de puzzel heen. Bijv: A3 of d5 (of k18?)."
172             << endl;
173             foutje = true;
174         }//if
175     }//if
176     if (!foutje){
177         getal = leesGetal(1, mijnpuzzel.hoogte - 1);
178         char kolom = letter + 'A' - 1;
179         cout << "gekozen zet: " << kolom << getal << endl ;
180         mijnpuzzel.zet(getal, letter);
181         mijnpuzzel.drukAf();
182         zetten++; cout << "zet " << zetten << endl;
183     }//if
184     if (mijnpuzzel.check()){
185         cout << "Gefeliciteerd, u heeft de puzzel opgelost" << endl;
186         return true;
187     }//if
188 }//while
189 return false;
190 }//puzzelen
191
192 //het puzzel submenu
193 void puzzelMenu(puzzel &mijnpuzzel){
194     mijnpuzzel.drukAf();
195     bool stoppen = false;
196     while(!stoppen){// stopt pas als de gebruiker dit wil, of wanneer opgelost
197         cout << "Kies een van de volgende opties:" << endl
198         << "[T]erug, [P]uzzelen, [L]os op, [C]hasing en [V]olg." << endl;
199         char keuze; cin >> keuze;
200         switch (keuze){
201             case 'p': case 'P':
202                 //gaat naar een submenu waar het puzzelen plaatsvindt
203                 if (puzzelen(mijnpuzzel)){stoppen = true;}
204                 break;
205             case 'l': case 'L':
206                 if (!mijnpuzzel.gegenereerd){
207                     cout << "Dit werkt alleen voor puzzels die met de optie "
208                     << "\"genereer\" zijn gemaakt uit het tekenmenu." << endl
209                     << "Probeer optie \"C\". " << endl; break;
210                 }//if
211                 losop(mijnpuzzel); stoppen = true; break;
212             case 'v': case 'V':
213                 mijnpuzzel.volg(); break;
214             case 'c': case 'C':
215                 if (mijnpuzzel.hoogte != 6 || mijnpuzzel.breedte != 6
216                 || mijnpuzzel.torus){
217                     cout << "Dit kan alleen voor een standaard 5x5 puzzel!"
218                     << endl;
219                     break;
220                 }//if
221                 mijnpuzzel.volg(); mijnpuzzel.laatsteRij(); mijnpuzzel.volg();

```

```

222         stoppen = true; break;
223     case 't': case 'T':
224         stoppen = true; break;
225     default:
226         cout << "ongeldige keuze \n" << keuze << endl; break;
227     }//switch
228 }//while
229 }//puzzelMenu
230
231 //het submenu waar een selectie eigenschappen van de puzzel kan worden aangepast
232 void parameterMenu(puzzel &mijnpuzzel){
233     bool stoppen=false;
234     while(!stoppen){//stopt pas als de gebruiker klaar is met aanpassen
235         cout << "Kies een van de volgende opties:" << endl
236         << " [T]erug, [D]imensies, [K]arakters, t[O]rus,"
237         << " pe[R]centage en [P]en." << endl;
238         char keuze; cin >> keuze;
239         switch(keuze){
240             case 'K': case 'k':
241                 cout << "Kies het karakter voor een brandende lamp"
242                 << " (geen 0 of 1):" << endl;
243                 mijnpuzzel.aankarakter = eerste();
244                 cout << "Kies een ander karakter voor een niet-brandende lamp"
245                 << " (geen 0 of 1):" << endl;
246                 mijnpuzzel.uitkarakter = eerste();
247                 checkKarakter(mijnpuzzel.aankarakter, mijnpuzzel.uitkarakter);
248                 break;
249             case 'D': case 'd':
250                 cout << "Kies de hoogte van het speelveld:"
251                 << " (3 - 20):" << endl;
252                 mijnpuzzel.hoogte = leesGetal(3, 20)+1;
253                 cout << "Kies de breedte van het speelveld"
254                 << " (3 - 20):" << endl;
255                 mijnpuzzel.breedte = leesGetal(3, 20)+1; break;
256             case 'R': case 'r':
257                 cout << "Zet het percentage lampen dat aangaat bij een random"
258                 << " speelveld (0-100):" << endl;
259                 mijnpuzzel.percentage = leesGetal(0,100); break;
260             case 'O': case 'o':
261                 if (mijnpuzzel.torus){
262                     mijnpuzzel.torus = false;
263                     cout << "Torusmode uit" << endl;
264                 }
265                 else {
266                     mijnpuzzel.torus = true;
267                     cout << "Torusmode aan" << endl;
268                 }
269                 break;
270             case 'P': case 'p':
271                 mijnpuzzel.pen++;
272                 if (mijnpuzzel.pen == 3){mijnpuzzel.pen = 0;}
273                 cout << "Penstand is nu " << mijnpuzzel.pen << endl;
274                 break;
275             case 'T': case 't':
276                 stoppen = true; break;
277             default:

```

```

278         cout << "ongeldige keuze \n" << keuze << endl; break;
279     }//switch
280 }//while
281 mijnpuzzel.maakWereld();//maakt een nieuwe wereld met de parameters
282 mijnpuzzel.maakRandom();
283 }//parameterMenu
284
285 //het menu waar de gebruiker (zelf of automatisch) een nieuwe puzzel kan maken
286 void tekenMenu(puzzel &mijnpuzzel){
287     bool stoppen = false;
288     int moeilijk = 1;
289     while (!stoppen){
290         cout << "Kies een van de volgende opties:" << endl
291         << "[T]erug, [R]andom, [G]enereer, t[O]ggel en [S]choon." << endl;
292         char keuze; cin >> keuze;
293         switch (keuze){
294             case 't': case 'T':
295                 stoppen = true; cout << "stoppen \n"; break;
296             case 'g': case 'G':
297                 cout << "kies moeilijkheidsgraad (een getal >= 1)" << endl;
298                 moeilijk = leesGetal(1,999);
299                 mijnpuzzel.moeilijkheid = moeilijk;
300                 mijnpuzzel.maakSchoon();
301                 mijnpuzzel.genereer();
302                 mijnpuzzel.drukAf();
303                 mijnpuzzel.gegenereerd = true;
304                 break;
305             case 'r': case 'R':
306                 mijnpuzzel.maakRandom();//maakSchoon is hier niet nodig
307                 mijnpuzzel.drukAf();
308                 mijnpuzzel.gegenereerd = false;
309                 break;
310             case 'o': case 'O':{
311                 tekenen(mijnpuzzel);
312                 break;
313             }
314             case 's': case 'S':
315                 mijnpuzzel.maakSchoon(); break;
316             default:
317                 cout << "ongeldige keuze \n" << keuze << endl; break;
318         }//switch
319     }//while
320 }
321
322 //het hoofdmenu
323 void menu(puzzel &mijnpuzzel){
324     bool stoppen = false;
325     while (!stoppen){
326         cout << "Kies een van de volgende opties:" << endl
327         << "[T]ekenen, [P]uzzelen, para[M]eters en [S]toppen." << endl;
328         char keuze; keuze = cin.get();
329         switch (keuze){
330             case 'm': case 'M':
331                 cout << "parameters \n"; parameterMenu(mijnpuzzel); break;
332             case 'p': case 'P':
333                 cout << "puzzel \n"; puzzelMenu(mijnpuzzel); break;

```



```

334         case 't': case 'T':
335             cout << "tekenen \n"; tekenMenu(mijnpuzzel); break;
336         case 's': case 'S':
337             cout << "stoppen \n"; stoppen = true; break;
338         case 'l': case 'L':
339             lebowski(); break;
340         default:
341             cout << "ongeldige keuze \n" << keuze << endl; break;
342     } //switch
343 } //while
344 } //menu
345
346 //main=====
347
348 int main() {
349     infoblok();
350     srand(time(0)); //dit voorkomt dat rand() bij elke run hetzelfde geeft
351     puzzel mijnpuzzel;
352     mijnpuzzel.maakWereld();
353     mijnpuzzel.maakRandom();
354     menu(mijnpuzzel);
355     cout << "Bedankt voor het spelen en tot ziens!";
356     return 0;
357 } //main
358
359 //constructors=====
360
361 //de constructor met gebruikersopties
362 puzzel::puzzel(int hoogteWaarde, int breedteWaarde, char aanKeuze,
363 char uitKeuze, int percentKeuze, bool torusKeuze, int penKeuze):
364 hoogte(hoogteWaarde), breedte(breedteWaarde), aankarakter(aanKeuze),
365 uitkarakter(uitKeuze), percentage(percentKeuze), torus(torusKeuze),
366 pen(penKeuze)
367 {
368     binnenDeGrenzen();
369 }
370
371 //de standaard constructor
372 puzzel::puzzel(): hoogte(6), breedte(6), aankarakter('x'), uitkarakter('.'),
373 percentage(30), torus(false), gegenereerd(false), pen(0)
374 {}
375
376 //memberfuncties=====
377
378 //puzzel moet tussen 3x3 (4 vakjes en coördinaten) en 26x26 zitten
379 void puzzel::binnenDeGrenzen()
380 {
381     if ((hoogte < 3) || (hoogte > 26) || (breedte < 3) || (breedte > 26)) {
382         cout << "Gekozen dimensies vallen niet binnen de grenzen" << endl;
383         exit(1);
384     } //if
385 } //puzzel::binnenDeGrenzen
386
387 void puzzel::drukAf()
388 {
389     int i, j; //nog iets met cursor?

```

```

390     for ( i = 0; i < hoogte; i++ ) { //skip de randen, i=1
391         for ( j = 0; j < breedte; j++ ) { //idem
392             if ( deWereld[i][j] == 'o' && cursor[i][j] )
393                 cout << " " << '0';
394             else if ( deWereld[i][j] == 'x' && cursor[i][j] )
395                 cout << " " << '1';
396             else if ( deWereld[i][j] == 'x' )
397                 cout << " " << aankarakter; // <= later "aankarakter"
398             else if ( deWereld[i][j] == 'o' )
399                 cout << " " << uitkarakter;
400             else if ( (deWereld[i][j] - '0') >= 10
401                 && (deWereld[i][j] - '0') < 20 && j == 0)
402                 cout << "1" << deWereld[i][j] - '0' - 10;
403             else if ( (deWereld[i][j] - '0') >= 20
404                 && (deWereld[i][j] - '0') < 27 && j == 0)
405                 cout << "2" << deWereld[i][j] - '0' - 20;
406             else
407                 cout << " " << deWereld[i][j];
408         } //for j
409         cout << endl;
410     } //for i
411 } //puzzel::drukaf
412
413 //Maakt de wereld/puzzel, inclusief de coördinaten
414 void puzzel::maakWereld()
415 {
416     int i, j;
417     for ( i = 0; i < hoogte; i++ ) {
418         for ( j = 0; j < breedte; j++ ) {
419             if ( i == 0 && j == 0 ) {
420                 deWereld[i][j] = ' '; //de hoek is leeg
421             }
422             else if ( i == 0 ) {
423                 char letter = j + 'A' - 1; //letter as
424                 deWereld[i][j] = letter;
425             }
426             else if ( j == 0 ) {
427                 char getal = i + '1' - 1; //getallen as
428                 deWereld[i][j] = getal;
429             }
430             else {
431                 deWereld[i][j] = 'o'; //lampen (uit)
432             }
433             oplossing[i][j] = false;
434             cursor[i][j] = false;
435         } //for
436     } //for
437 } //puzzel::maakWereld
438
439 //Zet alle lampen uit. Zelfde effect als maakWereld, maar iets sneller.
440 void puzzel::maakSchoon()
441 {
442     int i, j;
443     for ( i = 1; i < hoogte; i++ ) { //skip de randen, i=1
444         for ( j = 1; j < breedte; j++ ) { //idem
445             deWereld[i][j] = 'o';

```

```

446         oplossing[i][j] = false;
447     }//for
448 }//for
449 }//puzzel::maakSchoon
450
451 // Zet "random" een gedeelte van de lampen aan
452 void puzzel::maakRandom()
453 {
454     int percent = percentage;
455     int i, j;
456     for ( i = 1; i < hoogte; i++ ) { //skip de coördinaatassen, i=1
457         for ( j = 1; j < breedte; j++ ) { //idem
458             int TrueFalse = (rand() * randomgetal())%100; /* zie verslag
459             if (TrueFalse <= percent){
460                 deWereld[i][j] = 'x';
461             }
462             else {deWereld[i][j] = 'o';}
463         }
464     }
465     gegenereerd = false;
466 }//puzzel::maakRandom
467
468 //genereer een puzzel op basis van zetten, ipv individuele lampen
469 void puzzel::genereer()
470 {
471     int stap, i, j;
472     for (stap=1; stap <= moeilijkheid; stap++){
473         i = (rand() * randomgetal())%(hoogte-1) + 1; /* zie verslag
474         j = (rand() * randomgetal())%(breedte-1) + 1; /*
475         zet(i, j);
476     }//for
477     gegenereerd = true;
478 }//puzzel::genereer
479
480 //algoritme dat alle rijen behalve de onderste uit zet
481 void puzzel::volg()
482 {
483     int i, j;
484     for (i = 1; i < (hoogte-1); i++){
485         for ( j = 1; j < breedte; j++ ) {
486             if (deWereld[i][j]=='x'){
487                 zet(i+1,j);
488                 drukAf();
489             }//if
490         }//for
491     }//for
492 }//puzzel::volg
493
494 //lost, icm volg, de puzzel op dmv "light chasing". alleen voor 5x5 puzzels!
495 void puzzel::laatsteRij()
496 {
497     if(deWereld[5][1] == 'x' && deWereld[5][2] == 'x' && deWereld[5][3] == 'x'
498     && deWereld[5][4] == 'o' && deWereld[5][5] == 'o'){
499         zet(1,2); drukAf();}
500     else if(deWereld[5][1] == 'x' && deWereld[5][2] == 'x'
501     && deWereld[5][3] == 'o' && deWereld[5][4] == 'x' && deWereld[5][5] == 'x'){

```

```

502     zet(1,3); drukAf();}
503     else if(deWereld[5][1] == 'x' && deWereld[5][2] == 'o'
504     && deWereld[5][3] == 'x' && deWereld[5][4] == 'x' && deWereld[5][5] == 'o'){
505         zet(1,5); drukAf();}
506     else if(deWereld[5][1] == 'x' && deWereld[5][2] == 'o'
507     && deWereld[5][3] == 'o' && deWereld[5][4] == 'o' && deWereld[5][5] == 'x'){
508         zet(1,2); drukAf(); zet(1,1); drukAf();}
509     else if(deWereld[5][1] == 'o' && deWereld[5][2] == 'x'
510     && deWereld[5][3] == 'x' && deWereld[5][4] == 'o' && deWereld[5][5] == 'x'){
511         zet(1,1); drukAf();}
512     else if(deWereld[5][1] == 'o' && deWereld[5][2] == 'x'
513     && deWereld[5][3] == 'o' && deWereld[5][4] == 'x' && deWereld[5][5] == 'o'){
514         zet(1,1); drukAf(); zet(1,4); drukAf();}
515     else if(deWereld[5][1] == 'o' && deWereld[5][2] == 'o'
516     && deWereld[5][3] == 'x' && deWereld[5][4] == 'x' && deWereld[5][5] == 'x'){
517         zet(1,4); drukAf();}
518     else if(deWereld[5][1] == 'o' && deWereld[5][2] == 'o'
519     && deWereld[5][3] == 'o' && deWereld[5][4] == 'o' && deWereld[5][5] == 'o'){
520         cout << "reeds opgelost!" << endl;}
521     else {cout << "onoplosbaar" << endl;}
522 }//puzzel::laatsteRij
523
524 //checkt of de puzzel is opgelost
525 bool puzzel::check()
526 {
527     int i, j;
528     for ( i = 1; i < hoogte; i++ ) { //skip de randen, i=1
529         for ( j = 1; j < breedte; j++ ) { //idem
530             if (deWereld[i][j] == 'x'){return false;} //brandende lamp gevonden
531         } //for
532     } //for
533     return true;
534 } //puzzel::check
535
536 //voert een zet uit, en klapt geselecteerd vakje in een pluspatroon om
537 void puzzel::zet(int i, int j)
538 {
539     booltoggle(oplossing[i][j]); //houdt de zetten bij in oplossing
540     toggle(deWereld[i][j]);
541     toggle(deWereld[i+1][j]);
542     toggle(deWereld[i-1][j]);
543     if( !torus){
544         toggle(deWereld[i][j+1]);
545         toggle(deWereld[i][j-1]);
546     } //if
547     else{ //buitenste kolommen hebben effect op overkant in torusmode
548         if (j == 1){
549             toggle(deWereld[i][j+1]);
550             toggle(deWereld[i][breedte - 1]);
551         }
552         else if (j == (breedte - 1) ){
553             toggle(deWereld[i][j-1]);
554             toggle(deWereld[i][1]);
555         }
556         else {
557             toggle(deWereld[i][j-1]);

```

```

558         toggle(deWereld[i][j+1]);
559     }
560 }//else
561 }//puzzel::zet
562
563 //verandert de lampen bij het tekenen
564 void puzzel::eenlamp(int i, int j)
565 {
566     if (pen == 0){toggle(deWereld[i][j]);}
567     if (pen == 1){deWereld[i][j] = 'x';}
568     if (pen == 2){deWereld[i][j] = 'o';}
569     gegenereerd = false;
570 }//puzzel::eenlamp
571
572 // wordt gebruikt bij het genereren van de oplossing. omdat deze private is
573 bool puzzel::zetofniet(int i, int j){
574     return oplossing[i][j];
575 }//puzzel::zetofniet

```