

LightsOut!

Vincent van der Velden *s4494059*, Jasper Vos *s2911159*

10 november 2025

1 Uitleg

Hoe het spel werkt

LightsOut is een spel waarbij we een rooster hebben met een aantal lampen die aan of uit staan [1]. Het is de bedoeling om elke lamp uit te krijgen, waarbij de speler een lamp kan "togglen". Elke "toggle" zorgt er echter voor dat de desbetreffende lamp wordt omgedraaid en de 4 directe burens ook.

Structuur van ons programma

Het programma wat we hebben gemaakt heeft drie verschillende menu's:

Parameter menu:

Instellen van hoogte, breedte, karakters, proportie lampen aan, torus-modus, pen-modus, en animatie.

Puzzel menu:

Verschillende manieren om puzzels op te lossen,

- *volg()* Lost automatisch op behalve de onderste rij.
- *losOp()* Lost volledige puzzel op door volg steeds aan te roepen, en als het geen oplossing bevat wordt er een bepaalde combinatie in de bovenste rij aangeklikt om vervolgens *volg()* weer te proberen. Dit blijft doorgaan tot alle mogelijke lamp combinaties zijn geprobeerd.
- *speelOplossing()* Lost een volledige puzzel vanuit de geregistreerde "klikken" van een gegenereerde puzzel.
- *spelen()* Hiermee kun je zelf de puzzel spelen waarbij je de cursor kan bewegen met WASD, en klikken met E.

Teken menu:

- *maakSchoon()* Zet alle arrays die te maken hebben met de puzzel naar false.
- *randomBord()* Zet random lampen aan met een bepaalde proportie die je kan instellen bij het parameter menu.
- *toggle()* Hiermee kun je bepaalde lampen aan/uit zetten met de cursor die beweegt met WASD, en klikken met E.
- *genereerBord()* Dit zet random zetten waarmee we een puzzel genereren. De zetten worden opgeslagen en kan gebruikt worden voor *speelOplossing*.

Het spel kan ook wiskundig benaderd worden via lineaire algebra over $\text{GF}(2)$ [2], maar wij gebruiken eenvoudigere algoritmes.

Plaatje:

Hier is een plaatje van het spel:

```
|                               LightsOut                               |
-----
Vincent van der Velden - s4494059 - Wiskunde 2024
Jasper Vos - s2911159 - Wiskunde 2025
Programmeeropgave 3 - 10/11/2025

-----
Functies:
- Speel LightsOut! Met of zonder torus modus.
- Genereer oplosbare puzzels.
- Teken modus met pen functie.
- Speel oplossingen af met animaties!
-----

0      .      .      0      .
0      0      0      .      0
0      .      0      .      0
.      .      .      0      0
.      0      0      0      .

[P]arameters | p[U]zzelmenu | [T]ekenmenu | [S]toppen
|
```

Figuur 1: Plaatje van het programma

Compiler en overige problemen:

We hebben een probleem met het runnen van de code op windows-machines. Op Linux werkt de code prima en het programma crashed bij *genereerBord()* of *randomBord()*.

2 Tijd

Hieronder is een overzicht van de tijd die we gespendeerd hebben aan LightsOut.

Week	Vincent	Jasper	Totaal
13-17 okt	5u	4u	9u
20-24 okt	6u	7u	13u
27-31 okt	8u	8u	16u
3-7 nov	6u	5u	11u
10 nov	3u	4u	7u
Totaal	28u	28u	56u

Referenties

- [1] Wikipedia contributors, *Lights Out (game)*, Wikipedia, The Free Encyclopedia, [https://en.wikipedia.org/wiki/Lights_Out_\(game\)](https://en.wikipedia.org/wiki/Lights_Out_(game))
- [2] Anderson, M. and Feil, T., *Turning Lights Out with Linear Algebra*, Mathematics Magazine, https://people.sc.fsu.edu/~jburkardt/classes/imps_2017/11_28/2690705.pdf Vol. 71, No. 4 (1998), pp. 300-303

3 Code

```
1 #include <cstdlib>
2 #include <ctime>
3 #include <iostream>
4 #include <unistd.h>
5
6 using namespace std;
7
8 void infoblok() {
9     cout << "_____ " << endl
10         << "|               LightsOut               | " << endl
11         << "_____ " << endl
12         << endl
13         << "Vincent van der Velden - s4494059 - Wiskunde 2024" << endl
14         << "Jasper Vos - s2911159 - Wiskunde 2025" << endl
15         << "Programmeeropgave 3 - 10/11/2025" << endl
16         << endl
17         << "_____ " << endl
18         << "Functies:" << endl
19         << "- Speel LightsOut! Met of zonder torus modus." << endl
20         << "- Genereer oplosbare puzzels." << endl
21         << "- Teken modus met pen functie." << endl
22         << "- Speel oplossingen af met animaties!" << endl
23         << "_____ " << endl;
24 }
25
26 void wissel(int &x, int &y) {
27     // Verwisseld twee variabelen
28     int temp = x;
29     x = y;
30     y = temp;
31 }
32
33 char leesOptie() {
34     // Leest input van gebruiker per karakter
35     char resultaat;
36     do {
37         cin.get(resultaat);
38     } while (resultaat == '\n');
39     if (resultaat >= 'a' && resultaat <= 'z') {
40         resultaat = (resultaat - 'a') + 'A';
41     }
42     return resultaat;
43 }
44
```

```

45 int leesGetal(int max) {
46     // Controleert of de input een getal is
47     char resultaat = 0;
48     char karakter = leesOptie();
49     if (karakter >= '0' && karakter <= '9') {
50         resultaat = karakter - '0';
51     }
52     while (cin.get(karakter) && karakter != '\n') {
53         if (karakter >= '0' && karakter <= '9') {
54             resultaat = resultaat * 10 + (karakter - '0');
55             if (resultaat >= max)
56                 resultaat = max;
57         }
58     }
59     return resultaat;
60 }
61
62 class Puzzel {
63 private:
64     // Constante variabelen
65     const static int MAX_HOOGTE = 20;
66     const static int MAX_BREEDTE = 20;
67     const static int MAX_PROPORTIE = 100;
68     // States voor het scheiden van inputs en overige functies
69     enum State {
70         HOOFDMENU,
71         PARAMETER,
72         TEKEN,
73         TOGGLE,
74         PUZZEL,
75         SPELEN,
76     };
77     // Parameters
78     int hoogte = 5;
79     int breedte = 5;
80     int proportieLampenAan = 50;
81     int pen = 0;
82     char karakterLampAan = '0';
83     char karakterLampUit = '.';
84     bool isTorus = false;
85     bool isAnimatie = false;
86     // Nodig voor de random getallen
87     long seed = time(0);
88     // Om bij te houden waar de cursor is
89     int posX = 0;
90     int posY = 0;
91     // Arrays die nodig zijn voor de puzzel
92     bool lampen[MAX_HOOGTE][MAX_BREEDTE];
93     bool lampenStart[MAX_HOOGTE][MAX_BREEDTE];
94     bool oplossing[MAX_HOOGTE][MAX_BREEDTE];
95     // Overige
96     bool startProgramma = true;
97     bool eindigSpel = false;
98     State state = HOOFDMENU;
99
100 public:

```

```

101     Puzzel() { genereerBord(); }
102     // Starten van het programma
103     void start();
104     // Menu's
105     void inputHandler(char input);
106     void inputHoofdmenu(char input);
107     void inputParameterMenu(char input);
108     void inputPuzzelMenu(char input);
109     void inputTekensMenu(char input);
110     // Parameter menu
111     void veranderParameter(string naam, int &variabele, int max);
112     void veranderParameter(string naam, char &variabele);
113     void veranderParameter(string naam, bool &variabele);
114     // Tekens menu
115     void maakSchoon();
116     void randomBord();
117     void toggle(char input);
118     void genereerBord();
119     // Puzzel menu
120     void volg();
121     void losOp();
122     void speelOplossing();
123     void spelen(char input);
124     // Overige hulp functies
125     void randomArray(int grootte, int array[]);
126     void kopieArray(bool array1[MAX_HOOGTE][MAX_BREEDTE],
127                     bool array2[MAX_HOOGTE][MAX_BREEDTE]);
128     int randomGetal();
129     void bordTekenen();
130     void updateScherm();
131     void beweeg(int dx, int dy);
132     void klik(int x, int y);
133     void flipLamp(int x, int y);
134     void menuText();
135     void setState(State newState) { state = newState; }
136 };
137
138 void Puzzel::start() {
139     // Start het programma
140     system("clear");
141     bordTekenen();
142     char input = '\0';
143
144     while (!eindigSpel) {
145         input = leesOptie();
146
147         inputHandler(input);
148         updateScherm();
149     }
150 }
151 void Puzzel::veranderParameter(string naam, int &variabele, int max) {
152     // Functie nodig voor het veranderen van de parameters, merk op dat we 3
153     // varianten hebben gemaakt zodat elke soort type variabele compatibel is.
154     cout << "Geef nieuwe " << naam << ": ";
155     variabele = leesGetal(max);
156     maakSchoon();

```

```

157 }
158
159 void Puzzel::veranderParameter(string naam, char &variabele) {
160     cout << "Geef nieuwe " << naam << ": ";
161     variabele = leesOptie();
162     maakSchoon();
163 }
164
165 void Puzzel::veranderParameter(string naam, bool &variabele) {
166     cout << naam << "modus (j/n): ";
167     char keuze = leesOptie();
168     if (keuze == 'J') {
169         variabele = true;
170     } else if (keuze == 'N') {
171         variabele = false;
172     }
173     maakSchoon();
174 }
175
176 int Puzzel::randomGetal() {
177     // Geeft een random getal
178     seed = (221 * seed + 1) % 1000;
179     return seed;
180 }
181
182 void Puzzel::randomArray(int grootte, int array[]) {
183     // Maakt een random array met unieke getallen
184     for (int i = 0; i < grootte; i++) {
185         int r = i + randomGetal() % (grootte - i);
186         wissel(array[i], array[r]);
187     }
188 }
189
190 void Puzzel::randomBord() {
191     // Maak een bord door random lampen aan te zetten. Let op dat deze puzzels
192     // onoplosbaar kunnen zijn.
193     maakSchoon();
194     int aantalLampen = breedte * hoogte;
195     int getallen[aantalLampen];
196     for (int i = 0; i < aantalLampen; i++) {
197         getallen[i] = i;
198     }
199     int aantalLampenAan = (aantalLampen * proportieLampenAan) / 100;
200     randomArray(aantalLampen, getallen);
201
202     for (int i = 0; i < aantalLampenAan; i++) {
203         int rij = getallen[i] / breedte;
204         int kolom = getallen[i] % breedte;
205         lampen[rij][kolom] = true;
206         lampenStart[rij][kolom] = true;
207     }
208 }
209
210 void Puzzel::kopieArray(bool array1[MAX_HOOGTE][MAX_BREEDTE],
211                        bool array2[MAX_HOOGTE][MAX_BREEDTE]) {
212     // Maakt een kopie van een array

```

```

213     for (int i = 0; i < hoogte; i++) {
214         for (int j = 0; j < breedte; j++) {
215             array1[i][j] = array2[i][j];
216         }
217     }
218 }
219
220 void Puzzel::speelOplossing() {
221     // Lost puzzel op met de opgeslagen oplossing van genereerBord()
222     kopieArray(lampen, lampenStart);
223     for (int i = 0; i < hoogte; i++) {
224         for (int j = 0; j < breedte; j++) {
225             if (oplossing[i][j]) {
226                 klik(j, i);
227                 if (isAnimatie) {
228                     sleep(1);
229                     updateScherm();
230                 }
231             }
232         }
233     }
234 }
235
236 void Puzzel::genereerBord() {
237     // Maakt een bord door zetten te zetten. Deze puzzels zijn altijd oplosbaar.
238     maakSchoon();
239     int aantalLampen = breedte * hoogte;
240     int getallen[aantalLampen];
241     int moeilijkheidsgraad = randomGetal() % aantalLampen;
242     for (int i = 0; i < aantalLampen; i++) {
243         getallen[i] = i;
244     }
245     randomArray(aantalLampen, getallen);
246     if (startProgramma) {
247         startProgramma = false;
248     } else {
249         cout << "Moeilijkheidsgraad (1-" << aantalLampen << ") : ";
250         moeilijkheidsgraad = leesGetal(aantalLampen);
251     }
252
253     for (int i = 0; i < moeilijkheidsgraad; i++) {
254         int rij = getallen[i] / breedte;
255         int kolom = getallen[i] % breedte;
256         oplossing[rij][kolom] = true;
257         klik(kolom, rij);
258     }
259     kopieArray(lampenStart, lampen);
260 }
261
262 void Puzzel::inputTekensMenu(char input) {
263     // Switch voor alle verschillen opties
264     switch (input) {
265     case 'S':
266         maakSchoon();
267         break;
268     case 'R':

```

```

269     randomBord();
270     break;
271 case 'O':
272     setState(TOGGLE);
273     break;
274 case 'G':
275     genereerBord();
276     break;
277 case 'T':
278     setState(HOOFDMENU);
279     break;
280 }
281 }
282
283 void Puzzel::menuText() {
284     // Alle verschillende soorten menu's
285     switch (state) {
286     case HOOFDMENU:
287         cout << "[P]arameters | [U]zzelmenu | [T]ekenmenu | [S]toppen" << endl;
288         break;
289     case TEKEN:
290         cout << "[S]choon | [R]andom | [O]ggle | [G]enereer | [T]erug" << endl;
291         break;
292     case PUZZEL:
293         cout << "[V]olg | [L]os 5x5 | [S]peel oplossing | [P]elen | [T]erug"
294             << endl;
295         break;
296     case SPELEN:
297         cout << "[W]omhoog | [A]links | [S]omlaag | [D]rechts | [E]klik | [T]erug"
298             << endl;
299         break;
300     case TOGGLE:
301         cout << "[W]omhoog | [A]links | [S]omlaag | [D]rechts | [E]klik | [T]erug"
302             << endl;
303         break;
304     case PARAMETER:
305         cout << "[H]oogte | [B]reedte | [R]andom | [A]an | [U]it | [O]rus | [P]en "
306             << "[A]nimatie | [T]erug"
307             << endl;
308         break;
309     }
310 }
311
312 void Puzzel::inputHoofdmenu(char input) {
313     // Switch van alle opties in het hoofdmenu
314     switch (input) {
315     case 'P':
316         setState(PARAMETER);
317         break;
318     case 'U':
319         setState(PUZZEL);
320         break;
321     case 'T':
322         setState(TEKEN);
323         break;
324     case 'S':

```



```

325     eindigSpel = true;
326     break;
327 }
328 }
329
330 void Puzzel::inputParameterMenu(char input) {
331     // Switch van alle opties in het parameter menu
332     switch (input) {
333     case 'H':
334         veranderParameter("hoogte", hoogte, MAX_HOOGTE);
335         break;
336     case 'B':
337         veranderParameter("breedte", breedte, MAX_BREEDTE);
338         break;
339     case 'R':
340         veranderParameter("proportie", proportieLampenAan, MAX_PROPORTIE);
341         break;
342     case 'A':
343         veranderParameter("karakter", karakterLampAan);
344         break;
345     case 'U':
346         veranderParameter("karakter", karakterLampUit);
347         break;
348     case 'O':
349         veranderParameter("Torus ", isTorus);
350         break;
351     case 'N':
352         veranderParameter("Animatie ", isAnimatie);
353         break;
354     case 'P':
355         veranderParameter("pen modus(0 := uit, 1 := alles aan, 2 := alles uit)",
356                             pen, 2);
357         break;
358     case 'T':
359         setState(HOOFDMENU);
360         break;
361     }
362 }
363
364 void Puzzel::volg() {
365     // Lost alleen de laatste rij niet op
366     for (int i = 1; i < hoogte; i++) {
367         for (int j = 0; j < breedte; j++) {
368             if (lampen[i - 1][j]) {
369                 klik(j, i);
370                 if (isAnimatie) {
371                     updateScherm();
372                     sleep(1);
373                 }
374             }
375         }
376     }
377 }
378
379 void Puzzel::losOp() {
380     // Gebruikt volg en als er geen oplossing is proberen we verschillende

```

```

381 // combinaties van lampen in de bovenste rij aan te zetten, volg() wordt dan
382 // opnieuw aangeroepen en dit proces blijft herhalen tot alle rijen leeg zijn.
383 if (hoogte != 5 || breedte != 5) {
384     return;
385 }
386 bool isLeeg = true;
387 for (int p = 0; p < 32; p++) {
388     for (int j = 0; j < breedte; j++) {
389         if (p & (1 << j)) {
390             klik(j, 0);
391         }
392     }
393     volg();
394     isLeeg = true;
395     for (int j = 0; j < breedte; j++) {
396         if (lampen[hoogte - 1][j] == true) {
397             isLeeg = false;
398             break;
399         }
400     }
401     if (isLeeg) {
402         return;
403     } else {
404         for (int i = 0; i < hoogte; i++) {
405             for (int j = 0; j < breedte; j++) {
406                 lampen[i][j] = lampenStart[i][j];
407             }
408         }
409     }
410 }
411 }
412
413 void Puzzel::inputPuzzelMenu(char input) {
414     // Switch van alle opties in het puzzel menu
415     switch (input) {
416         case 'V':
417             volg();
418             break;
419         case 'L':
420             losOp();
421             break;
422         case 'S':
423             speelOplossing();
424             break;
425         case 'P':
426             setState(SPELEN);
427             break;
428         case 'T':
429             setState(HOOFDMENU);
430             break;
431     }
432 }
433 void Puzzel::inputHandler(char input) {
434     // Stuur de input naar de goede state
435     switch (state) {
436         case HOOFDMENU:

```

```

437     inputHoofdmenu(input);
438     break;
439 case PARAMETER:
440     inputParameterMenu(input);
441     break;
442 case PUZZEL:
443     inputPuzzelMenu(input);
444     break;
445 case TEKEN:
446     inputTekkenMenu(input);
447     break;
448 case SPELEN:
449     spelen(input);
450     break;
451 case TOGGLE:
452     toggle(input);
453     break;
454 }
455 }
456
457 void Puzzel::bordTekenen() {
458     // Tekent het bord en menu
459     infoblok();
460     char karakter = '\0';
461     for (int i = 0; i < hoogte; i++) {
462         for (int j = 0; j < breedte; j++) {
463             if (lampen[i][j]) {
464                 karakter = karakterLampAan;
465             } else {
466                 karakter = karakterLampUit;
467             }
468             if (posX == j && posY == i && (state == SPELEN || state == TOGGLE)) {
469
470                 cout << " [" << karakter << "]" << " ";
471             } else {
472                 cout << " " << karakter << " ";
473             }
474         }
475         cout << "\n";
476     }
477     menuText();
478 }
479
480 void Puzzel::maakSchoon() {
481     // Zet alle elementen in de arrays naar "false"
482     for (int i = 0; i < MAX_HOOGTE; i++) {
483         for (int j = 0; j < MAX_BREEDTE; j++) {
484             lampen[i][j] = false;
485             lampenStart[i][j] = false;
486             oplossing[i][j] = false;
487         }
488     }
489 }
490
491 void Puzzel::toggle(char input) {
492     // Toggle modus waarbij we lampen kunnen aan/uit zetten

```

```

493     switch (input) {
494     case 'D':
495         beweeg(1, 0);
496         break;
497     case 'A':
498         beweeg(-1, 0);
499         break;
500     case 'W':
501         beweeg(0, -1);
502         break;
503     case 'S':
504         beweeg(0, 1);
505         break;
506     case 'E':
507         if (pen == 0) {
508             lampen[posY][posX] = !lampen[posY][posX];
509         }
510         break;
511     case 'T':
512         setState(TEKEN);
513         break;
514     }
515     if (pen == 1) {
516         lampen[posY][posX] = true;
517     } else if (pen == 2) {
518         lampen[posY][posX] = false;
519     }
520 }
521
522 void Puzzel::spelen(char input) {
523     // Het spel spelen
524     switch (input) {
525     case 'D':
526         beweeg(1, 0);
527         break;
528     case 'A':
529         beweeg(-1, 0);
530         break;
531     case 'W':
532         beweeg(0, -1);
533         break;
534     case 'S':
535         beweeg(0, 1);
536         break;
537     case 'E':
538         klik(posX, posY);
539         break;
540     case 'T':
541         setState(PUZZEL);
542         break;
543     }
544 }
545
546 void Puzzel::updateScherm() {
547     // Maakt de terminal schoon, en tekent het bord
548     system("clear");

```

```

549     bordTekenen();
550 }
551
552 void Puzzel::beweeg(int x, int y) {
553     // Berekent juiste positie
554     posX = (posX + x + breedte) % breedte;
555     posY = (posY + y + hoogte) % hoogte;
556 }
557
558 void Puzzel::flipLamp(int x, int y) {
559     // Flipt de desbetreffende lamp
560     if (isTorus) {
561         if (x < 0) {
562             x = (x + breedte) % breedte;
563         }
564         if (y < 0) {
565             y = (y + hoogte) % hoogte;
566         }
567     }
568     if (y >= 0 && y < hoogte && x >= 0 && x < breedte) {
569         lampen[y][x] = !lampen[y][x];
570     }
571 }
572
573 void Puzzel::klik(int x, int y) {
574     // Flipt de lamp en de buren
575     flipLamp(x, y);
576     flipLamp(x - 1, y);
577     flipLamp(x + 1, y);
578     flipLamp(x, y - 1);
579     flipLamp(x, y + 1);
580 }
581
582 int main() {
583     Puzzel puzzel = Puzzel();
584     puzzel.start();
585     return 0;
586 }

```