



Роботын тэмцээн

Зегедийн Их сургуулийн Хиймэл оюуны судлаач нар робот программчлах тэмцээн зохион байгуулж байгаа. Таны найз Ханга энэ тэмцээнд оролцохоор шийдсэн. Зорилго нь Унгарын алдарт хоньч нохойн үүлдэр Пулигийн ухаалаг байдлыг тусгасан төгс *Пулиботыг* программчлах явдал юм.

Пулиботыг $(H + 2) \times (W + 2)$ хэмжээтэй нүднүүдээс тогтох төөрдөг байшин дээр тестэлнэ. Мөрүүдийг -1 утгаас H хүртэлх утгуудаар, хойноос урагш чиглэлд дугаарлах ба багануудыг -1 утгаас W хүртэлх утгуудаар баруунаас зүүн тийш дугаарлана. Бид r -р мөрний c -р багана дээр байрлах $(-1 \leq r \leq H, -1 \leq c \leq W)$ нүд рүү (r, c) нүд гэж хандана.

$0 \leq r < H$ ба $0 \leq c < W$ үед (r, c) нүдийг авч үзье. (r, c) нүдтэй **хөрш** 4 нүд байна:

- $(r, c - 1)$ нүдийг (r, c) -гийн **баруун** талын нүд гэнэ;
- $(r + 1, c)$ нүдийг (r, c) -гийн **урд** талын нүд гэнэ;
- $(r, c + 1)$ нүдийг (r, c) -гийн **зүүн** талын нүд гэнэ;
- $(r - 1, c)$ нүдийг (r, c) -гийн **хойд** талын нүд гэнэ.

Хэрэв $r = -1$ эсвэл $r = H$ эсвэл $c = -1$ эсвэл $c = W$ бол (r, c) нүдийг төөрдөг байшингийн **захын** нүд гэнэ. Төөрдөг байшингийн захын биш нүд нэг бол **саад** эсвэл **хоосон** нүд байна. Мөн хоосон нүд бүр 0-ээс Z_{MAX} хүртэлх тоогоор илэрхийлэгдэх **өнгөтэй** байна. Анх хоосон нүд бүрийн өнгө нь 0 байна.

Жишээ нь $H = 4$ ба $W = 5$ байх үеийн, зөвхөн $(1, 3)$ нүдэндээ саад бүхий төөрдөг байшинг авч үзье:

	-1	0	1	2	3	4	5
-1							
0		0	0	0	0	0	
1		0	0	0		0	
2		0	0	0	0	0	
3		0	0	0	0	0	
4							

Саадтай ганц нүдийг хэрээсээр тэмдэглэсэн. Захын нүднүүдийг саарал өнгөөр будсан. Хоосон нүднүүдэд байгаа тоо нь харгалзах өнгийг илэрхийлнэ.

$(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$ гэсэн, хос хосоороо ялгаатай хоосон үүрнүүдийн дарааллын хувьд i ($0 \leq i < \ell$) утга бүрийн хувьд (r_i, c_i) ба (r_{i+1}, c_{i+1}) үүрүүд хөрш байвал түүнийг (r_0, c_0) үүрнээс (r_ℓ, c_ℓ) үүр хүртэлх ℓ ($\ell > 0$) урттай **зам** гэж нэрлэнэ.

ℓ урттай зам нь яг $\ell + 1$ тооны нүдийг агуулна гэдгийг анхаар.

Тэмцээн дээр судлаач нар $(0, 0)$ нүднээс $(H - 1, W - 1)$ нүд хүртэл дор хаяж нэг зам оршин байхаар төөрдөг байшинг зохиосон. Энэ нь $(0, 0)$ ба $(H - 1, W - 1)$ нүднүүд хоосон байх нь баталгаатай гэсэн үг юм.

Ханга төөрдөг байшингийн аль нүднүүд нь хоосон болон аль нүднүүд нь саад гэдгийг мэдэхгүй.

Таны даалгавар бол судлаач нарын зохиосон үл мэдэгдэх төөрдөг байшин дотор $(0, 0)$ нүднээс $(H - 1, W - 1)$ нүд хүрэх *хамгийн богино замыг* (минимум урттай зам) олдог болгож Пулиботыг программчлах явдал юм. Пулиботын тодорхойлолт ба тэмцээний дүрмийг доор үзүүлсэн.

Уг бодлогын сүүлийн хэсэг нь Пулиботыг харуулах дэлгэцийн хэрэгслийн тухай байгаа.

Пулиботын тодорхойлолт

$-1 \leq r \leq H$ ба $-1 \leq c \leq W$ байх үед (r, c) нүдний **төлөвийг** доорх нөхцлүүдийг хангах бүхэл тоо байдлаар тодорхойлъё:

- хэрэв (r, c) нүд нь захын нүд бол түүний төлөв нь -2 ;
- хэрэв (r, c) нүд нь саад бол түүний төлөв нь -1 ;
- хэрэв (r, c) нүд нь хоосон нүд бол түүний төлөв нь нүдний өнгө байна.

Пулиботын программ алхмуудын дараалал байдлаар биелэгдэнэ. Алхам бүрт Пулибот ойрын нүднүүдийн төлөвийг таньж, дараа нь команд биелүүлнэ. Гүйцэтгэх командыг таньсан төлвүүдээр тодорхойлно. Үүнийг доор илүү дэлгэрэнгүй тайлбарласан.

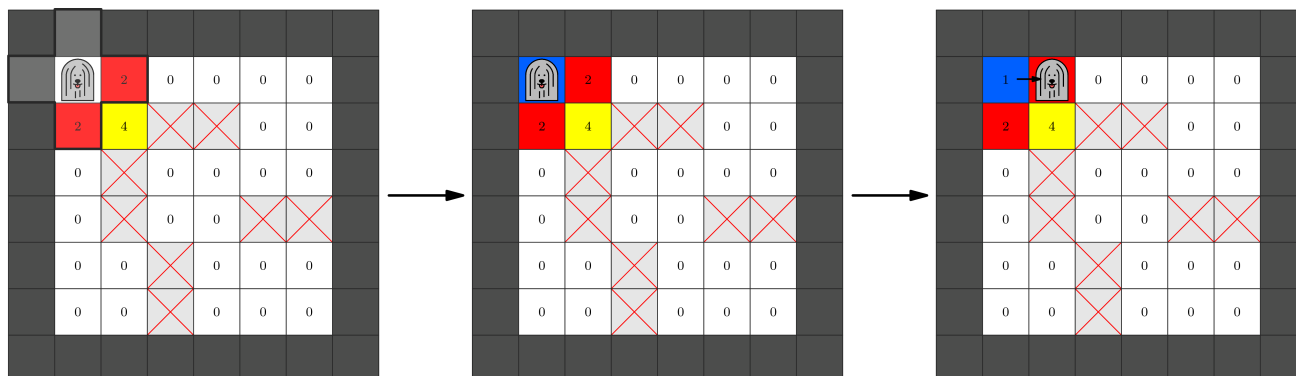
Одоогийн алхмын эхэнд Пулибот (r, c) гэсэн хоосон нүдэнд байна гэж үзье. Уг алхмыг дараах байдлаар гүйцэтгэнэ:

1. Эхлээд Пулибот одоогийн **төлөвийн массив** болох (r, c) нүдний болон түүний бүх хөршүүдийнх нь төлвөөс тогтох $S = [S[0], S[1], S[2], S[3], S[4]]$ массивыг танина:
 - $S[0]$ нь (r, c) нүдний төлөв.
 - $S[1]$ нь баруун нүдний төлөв.
 - $S[2]$ нь урд нүдний төлөв.
 - $S[3]$ нь зүүн нүдний төлөв.
 - $S[4]$ нь хойд нүдний төлөв.
2. Дараа нь Пулибот (Z, A) **командыг** тодорхойлох ба энэ нь таних төлөвийн массивт харгалзана.

3. Эцэст нь Пулибот уг командыг биелүүлнэ: (r, c) үүрийн өнгийг Z болгоод дараа нь доорх үйлдлүүдийн нэг болох A үйлдлийг гүйцэтгэнэ:

- (r, c) нүдэнд *үлдэх*;
- 4 хөрш нүдний нэг рүү *шилжих*;
- *программыг дуусгах*.

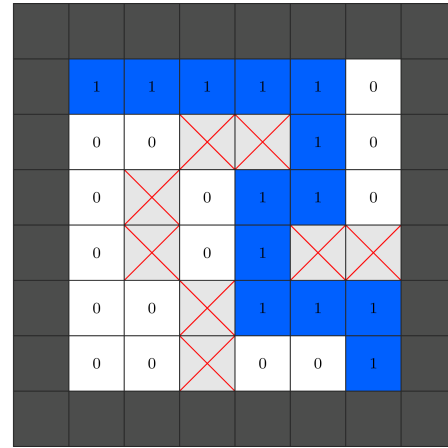
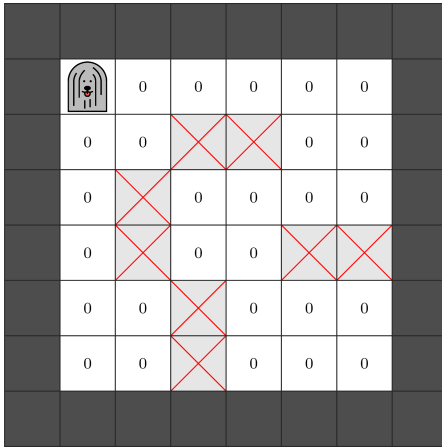
Жишээ нь, доорх зургийн зүүн талд байгаа тохиолдлыг авч үзье. Пулибот одоо 0 өнгөтэй $(0, 0)$ нүдэнд байгаа. Пулибот $S = [0, -2, 2, 2, -2]$ төлөвийн массивыг таньсан. Пулиботын программ энэ массивыг таниад зургийн голын болон баруун хэсэгт үзүүлснээр одоогийн нүдийг $Z = 1$ өнгөтэй болгоод зүүн тийш шилждэг байж болно:



Роботын тэмцээний дүрэм

- Эхэндээ Пулибот $(0, 0)$ нүдэнд байх ба программаа ажиллуулж эхэлнэ.
- Пулибот хоосон биш нүд рүү шилжихгүй.
- Пулиботын программ дээд тал нь 500 000 алхмын дараа дуусах ёстой.
- Пулиботын программ дууссаны дараа төөрдөг байшин дахь хоосон нүднүүд дараах байдлаар будагдсан байна:
 - Дотор нь агуулагдах нүд бүрийн өнгө нь 1 байх $(0, 0)$ -ээс $(H - 1, W - 1)$ хүрэх хамгийн богино зам оршин байна.
 - Бусад бүх хоосон нүдний өнгө нь 0 байна.
- Пулибот өөрийн программаа ямар ч хоосон нүдэнд дуусгаж болно.

Жишээ нь доорх зураг дээр $H = W = 6$ байх нэгэн боломжит төөрдөг байшинг үзүүлжээ. Эхний байрлалыг зүүн талд, дууссаны дараах хоосон нүднүүдийн боломжит нэг будалтыг баруун талд үзүүлсэн:



Хэрэгжүүлэлтийн мэдээлэл

Та дараах процедурыг хэрэгжүүлнэ.

```
void program_pulibot()
```

- Энэ процедур нь Пулиботын программыг үүсгэнэ. Энэ программ нь бодлогын хязгаарлалтыг хангах бүх H ба W утгууд болон төөрдөг байшин дээр зөв ажиллах ёстой.
- Уг процедурыг тест бүрийн хувьд яг нэг удаа дуудна.

Энэ процедур нь Пулиботын программыг үүсгэхийн тулд доорх процедурыг дуудаж болно:

```
void set_instruction(int[] S, int Z, char A)
```

- S : 5 урттай, төлөвийн массивыг илэрхийлэх массив.
- Z : өнгийг илэрхийлэх сөрөг бүхэл тоо.
- A : доорх байдлаар Пулиботын үйлдлийг илэрхийлэх нэг тэмдэгт:
 - H: үлдэх;
 - W: баруун тийш шилжих;
 - S: урагшаа шилжих;
 - E: зүүн тийш шилжих;
 - N: хойшоо шилжих;
 - T: программыг дуусгах.
- Энэ процедурыг дуудсанаар Пулибот таних төлөвийн массив S -т байх үедээ (Z, A) командыг биелүүлнэ.

Энэ процедурыг S массивын ижил төлөв дээр олон удаа дуудвал Output isn't correct гэсэн мэдэгдлийг авна.

Боломжит төлөвийн массив S бүр дээр set_instruction-г дуудах шаардлагагүй. Харин Пулибот өмнө нь команд оноогоогүй төлөвийн массивыг таньсан тохиолдолд та Output

isn't correct мэдэгдлийг авах болно.

program_pulibot дууссаны дараа шалгагч нь Пулиботын программыг нэг эсвэл хэд хэдэн төөрдөг байшин дээр дуудна. Эдгээр дуудалтууд нь таны шийдлийн хугацааны хязгаарлалтанд *тооцогдохгүй*. Шалгагч нь дасан зохицдоггүй ба энэ нь тест бүрийн хувьд төөрдөг байшингийн олонлогийг нь урьдчилан тодорхойлсон байгаа гэсэн үг юм.

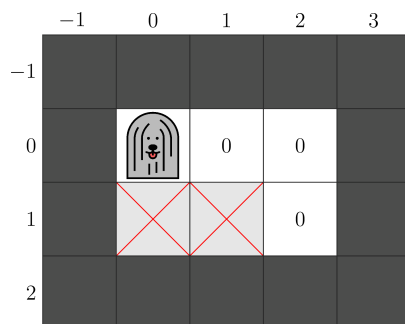
Хэрэв Пулибот программаа дуусахаас өмнө Роботын тэмцээний дүрмийг зөрчвөл та Output isn't correct мэдэгдлийг авах болно.

Жишээ

program_pulibot процедур нь set_instruction-г доорх байдлаар дуудаж болно:

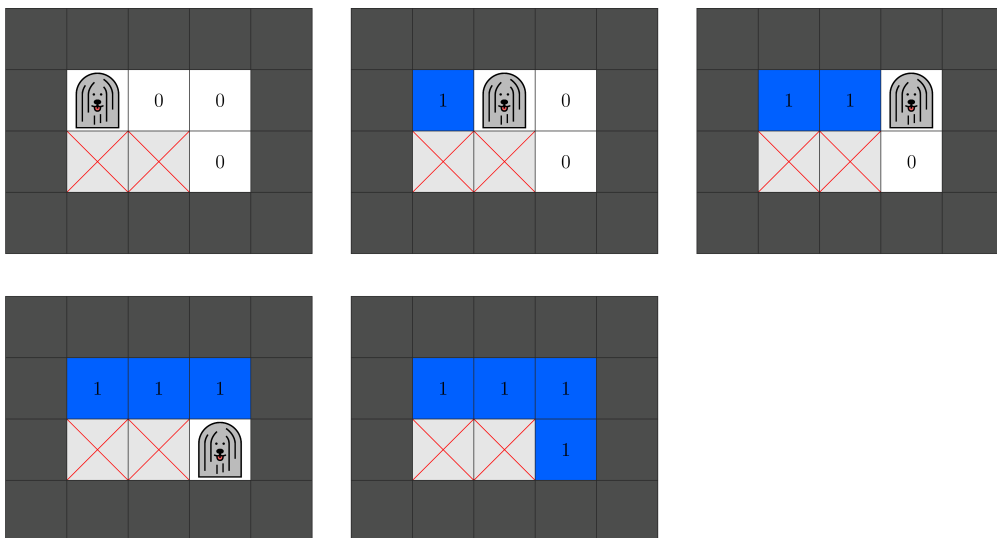
Дуудалт	<i>S</i> төлөвийн массивын команд
set_instruction([0, -2, -1, 0, -2], 1, E)	Өнгийг 1 болгоод зүүн тийш шилжих
set_instruction([0, 1, -1, 0, -2], 1, E)	Өнгийг 1 болгоод зүүн тийш шилжих
set_instruction([0, 1, 0, -2, -2], 1, S)	Өнгийг 1 болгоод урагшаа шилжих
set_instruction([0, -1, -2, -2, 1], 1, T)	Өнгийг 1 болгоод программыг дуусгах

$H = 2$ ба $W = 3$ байх доорх зурагт үзүүлсэн төөрдөг байшингийн тохиолдлыг авч үзье.



Энэ төөрдөг байшингийн хувьд Пулиботын программ дөрвөн алхамтай байна. Пулиботын таних төлөвийн массив болон түүний гүйцэтгэх командууд нь дээр байгаа set_instruction-ын дөрвөн дуудалттай харгалзах ба дарааллаараа таарна. Хамгийн сүүлийн команд нь программыг дуусгана.

Доорх зураг дээр дөрвөн алхам бүрийн өмнөх болон дууссаны дараах төөрдөг байшинг болон түүний эцсийн өнгийг үзүүлэв.



Гэвч 4 команд бүхий уг программ нь өөр зөв төөрдөг байшингууд дээр хамгийн богино замыг олохгүй байж болохыг анхаар. Иймд үүнийг илгээвэл Output isn't correct мэдэгдлийг авна.

Хязгаарлалт

$Z_{MAX} = 19$. Иймд Пулибот 0-ээс 19 хүртэлх өнгийг хэрэглэж чадна.

Пулиботыг тестлэхэд хэрэглэх төөрдөг байшин бүрийн хувьд:

- $2 \leq H, W \leq 15$
- $(0, 0)$ нүднээс $(H - 1, W - 1)$ нүд хүрэх дор хаяж нэг зам байна.

Дэд бодлого

1. (6 оноо) Төөрдөг байшинд саадтай нүд байхгүй.
2. (10 оноо) $H = 2$
3. (18 оноо) Хос хоосон нүд бүрийн хувьд тэдгээрийн хооронд яг нэг зам байна.
4. (20 оноо) $(0, 0)$ нүднээс $(H - 1, W - 1)$ хүрэх хамгийн богино зам бүр нь $H + W - 2$ урттай байна.
5. (46 оноо) Нэмэлт хязгаарлалт байхгүй.

Хэрэв ямар нэг тестийн хувьд set_instruction процедурын дуудалт эсвэл Пулиботын программын биелэлтийн явцад Хэрэгжүүлэлтийн мэдээлэл хэсэг дэх хязгаарлалтуудаас зөрчвөл та уг дэд бодлогоос 0 оноо авах болно.

Дэд бодлого бүр дээр та бараг зөв будалтыг үүсгэснээр хэсэгчилсэн оноо авч болно.

Өөрөөр хэлбэл:

- Хэрэв хоосон нүднүүдийн эцсийн будалт нь Роботын тэмцээний дүрмийг хангаж байвал шийдлийн тест дээрх оноо нь **бүтэн** байна.

- Хэрэв эцсийн будалт нь доорх байдалтай бол шийдлийн тест дээрх оноо нь **хэсэгчилсэн** байна:
 - Дотор нь агуулагдах нүд бүрийн өнгө нь 1 байх $(0, 0)$ -ээс $(H - 1, W - 1)$ хүрэх зам оршин байна.
 - Төөрдөг байшинд үүнээс өөр 1 гэсэн өнгөтэй хоосон нүд байхгүй.
 - Төөрдөг байшин дахь зарим хоосон нүд нь 0 ба 1-ээс өөр өнгөтэй байна.

Хэрэв таны шийдлийн тест дээрх оноо нь бүтэн ч биш хэсэгчилсэн ч биш байвал уг тестэд харгалзах оноо нь 0 байна.

Дэд бодлого 1-4 дээр бүтэн шийдэл 100% авах ба таны хэсэгчилсэн шийдэл дэд бодлогын онооны 50%-г авна.

Дэд бодлого 5 дээр таны авах оноо нь Пулиботын программ дээр хэрэглэсэн өнгөнүүдийн тооноос хамаарна. `set_instruction`-ыг дуудсан бүх дуудалтуудын Z утгуудын хамгийн ихийг Z^* гэж тэмдэглэе. Тест дээр авах оноог доорх хүснэгтийн дагуу тооцоолно:

Нөхцөл	Оноо (бүтэн)	Оноо (хэсэгчилсэн)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

Дэд бодлого бүрийн оноо нь түүний тестүүд дээр авсан оноонуудын минимум нь байна.

Жишээ шалгагч

Жишээ шалгагч нь оролтыг доорх хэлбэрээр уншина:

- мөр 1: $H \ W$
- мөр $2 + r$ ($0 \leq r < H$): $m[r][0] \ m[r][1] \ \dots \ m[r][W - 1]$

Энд, m нь W тооны бүхэл тоог агуулсан H тооны массивуудын массив ба төөрдөг байшингийн захын бус нүднүүдийг илэрхийлнэ. Хэрэв (r, c) нь хоосон нүд бол $m[r][c] = 0$ байх ба харин саадтай нүд бол $m[r][c] = 1$ байна.

Жишээ шалгагч нь эхлээд `program_pulibot()`-г дуудна. Хэрэв жишээ шалгагч протоколын зөрчлийг илрүүлбэл `Protocol Violation: <MSG>` гэж хэвлээд дуусна. Энд `<MSG>` нь доорх алдааны мэдэгдлүүдийн нэг байна:

- Invalid array: $-2 \leq S[i] \leq Z_{MAX}$ нөхцөл ямар нэг i -гийн хувьд хангагдаагүй эсвэл S -ийн урт нь 5 биш байх.
- Invalid color: $0 \leq Z \leq Z_{MAX}$ нөхцлийг хангаагүй.
- Invalid action: A тэмдэгт нь H, W, S, E, N эсвэл T тэмдэгтийн нэг биш.
- Same state array: set_instruction процедурыг ижил S массивтайгаар дор хаяж хоёр удаа дуудсан.

Эсрэг тохиолдолд program_pulibot дууссаны дараа жишээ шалгагч Пулиботын программыг оролтон дээр дүрсэлсэн төөрдөг байшинд ажиллуулна.

Жишээ шалгагч нь хоёр гаралтыг гаргана.

Нэгдүгээрт, жишээ шалгагч нь ажлын хавтсанд Пулиботын үйлдлүүдийн бүртгэлийг robot.bin файл руу бичнэ. Энэ файл нь дараагийн хэсэгт дүрсэлсэн хэрэгслийн оролт болно.

Хоёрдугаарт, хэрэв Пулиботын программ амжилттай дуусаагүй бол жишээ шалгагч доорх алдааны мэдэгдлүүдийн нэгийг хэвлэнэ:

- Unexpected state: Пулиботын таньсан төлөвийн массив нь set_instruction-ыг дуудахад ашиглагдаагүй массив байх.
- Invalid move: Пулиботыг хоосон биш нүд рүү шилжихэд хүргэсэн үйлдлийг гүйцэтгэх.
- Too many steps: Пулибот программаа дуусгалгүйгээр 500 000 алхам гүйцэтгэх.

Эсрэг тохиолдолд $e[r][c]$ нь Пулиботын программ дууссаны дараах (r, c) нүдний төлөв байг. Жишээ шалгагч H тооны мөрийг дараах хэлбэрээр хэвлэнэ:

- мөр $1 + r$ ($0 \leq r < H$): $e[r][0] \ e[r][1] \ \dots \ e[r][W - 1]$

Дэлгэцийн хэрэгсэл

Уг бодлогын хавсралт багц дотор display.py нэртэй файл байгаа. Үүнийг ажиллуулахад энэ Пайтон скрипт нь жишээ шалгагчийн оролтоор тодорхойлогдох төөрдөг байшин дахь Пулиботын үйлдлүүдийг үзүүлнэ. Ингэхийн тулд robot.bin хоёртын файл ажлын хавтсан дотор байх ёстой.

Энэ скриптийг дуудахын тулд доорх командыг биелүүлнэ.

```
python3 display.py
```

Энгийг график интерфейс гарч ирнэ. Энэ нь доорх гол онцлогуудтай:

- Та төөрдөг байшингийн төлөвийг бүхлээр нь харж болно. Пулиботын одоогийн байрлалыг тэгш өнцөгтөөр дүрсэлнэ.
- Та Пулиботын алхмуудаар сумнууд эсвэл харгалзах товчнуудыг ашиглан явж болно. Та мөн тодорхой алхам руу үсэрч болно.

- Пулиботын дараагийн алхмыг доор нь харуулна. Тэнд одоогийн төлөвийн массив болон гүйцэтгэх команд байрлана. Эцсийн алхмын дараа шалгагчийн алдааны мэдэгдлүүдийн нэгийг эсвэл программ амжилттай дууссан тохиолдолд Terminated гэж хэвлэнэ.
- Өнгө илэрхийлэх тоо бүрийн хувьд та арын өнгийг нь эсвэл текстийг нь тодорхойлж болно. Текст нь тухайн өнгөтэй нүд бүр дээр бичигдэх богино тэмдэг мөр байна. Та дараах аргуудаар арын өнгө эсвэл текстийг оноож болно:
 - Colors товчийг дарахад гарч ирэх цонхон дээр тохируулах.
 - colors.txt файлыг засварлах.
- robot.bin-г дахин ачаалахын тулд Reload товчийг ашиглана. robot.bin-г өөрчлөн ашиглах нь илүү хэрэгтэй байна.