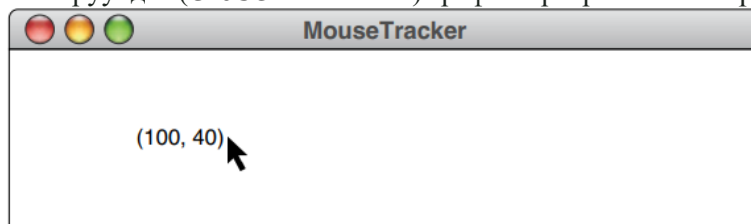


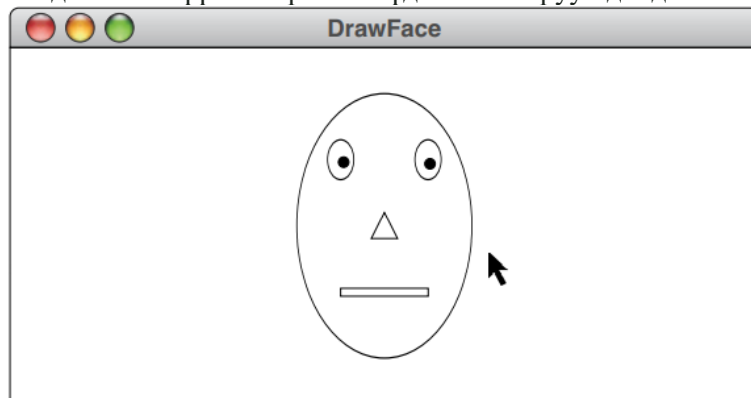
## Даалгавар 12

Eric Roberts-ийн “The Art and Science of Java” номны Бүлэг 10-ийн програмчлалын дасгалууд болно.

- Бүлэг 9-ийн 3-р дасгалаар гүйцэтгэсэн **RandomColorLabel** программыг дараах хэлбэрээр өөрчилнө үү.
  - Аль нэг **GLabel** дээр хулганаар дарах (press) үед тухайн GLabel-ийг дээр нь бичигдсэн текстэд харгалзах өнгөөр будна. Хулганаар дарахаа болих үед (release) өөрийн өнгөнөөс бусад санамсаргүй өнгөөр будна.
- Дэлгэц дээгүүр хулганыг хөдөлгөх (move) болон чирэх (drag) үед хулганын байгаа байрлалд хулганын одоо байгаа координатыг харуулдаг (**GLabel** ашиглана) график программ бичнэ үү. Жишээлбэл:

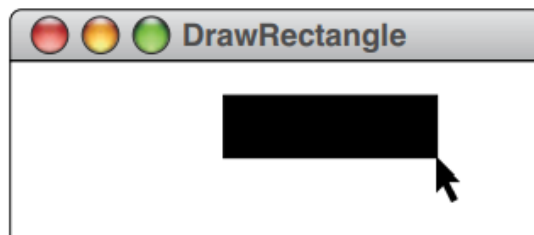
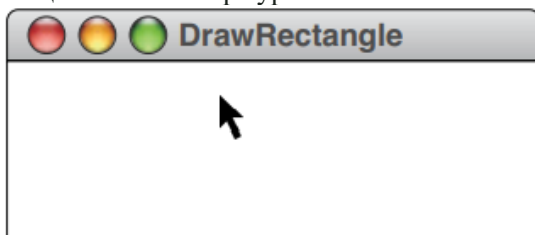


- Бүлэг 9.4 дээр хэрэгжүүлсэн **DrawFace** программыг нүдний хүүхэн хараатай болгож дахин бичнэ үү. Харах чиглэл нь хулганын дэлгэц дээр байгаа байрлал руу хардаг байна. Жишээлбэл, хулгана доош, нүүрний баруун талд байвал хүүхэн хараа нь нүднийхээ баруун доод хэсэгт байна.

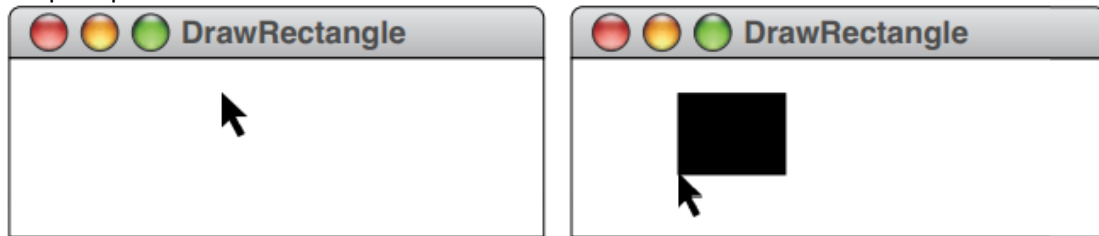


Хулганыг хөдөлгөх үед хүүхэн хараа нь дагаж хөдөлдөг байна. Хулгана заавал нүүрний гадна байх албагүй тул нүд бүрийн хувьд хүүхэн хараа харж буй чиглэлийг тус тусад нь тооцох хэрэгтэй. Жишээлбэл, хулгана хоёр нүдний хооронд байрлавал хүүхэн хараа нь харилцан эсрэг чиглэлд буюу солир харах хэрэгтэй.

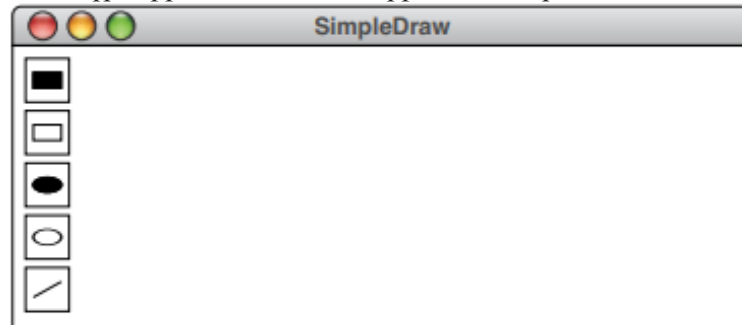
- DrawLine** программыг өөрчлөн канвас дээр өөр дүрснүүдийг зурдаг болгож болно. Ихэнх зурах программд хулганын дарж эхэлсэн байрлал болон дарахаа больсон байрлалд эсрэг орой бүхий тэгш өнцөгт үүсгэх үйлдэл байдаг. Жишээлбэл, хулганыг дараад дарж эхэлсэн байрлалаас баруун доош хөдөлгөхөд дараах зурагт харуулсан шиг тэгш өнцөгт үүснэ. Хулганын чирэх үед тэгш өнцөгт дагаад томорч, жижгэрдэг байх бөгөөд хулганыг дарахаа болих үед тэгш өнцөгт дахин өөрчлөгдөхгүй. Үүний дараа дахин өөр тэгш өнцөгтийг шинээр зурах боломжтой болно.



Хэдий энэ дасгалын код нь богино боловч дараах зүйлсийг анхаарах хэрэгтэй. Өмнөх жишээнд тэгш өнцөгтийг зүүн дээд өнцгөөс эхлээд баруун доод буланг чирч өөрчлөн зурсан. Тэгвэл хулганыг заавал баруун доош чиглэлд бус бусад чиглэлд чирэхэд мөн адил харгалзах тэгш өнцөгтийг зурдаг байх хэрэгтэй. Жишээ болгон зүүн доош чирэх үед хэрхэн зурагдахыг харууллаа.



5. **DrawLine, DrawRectangle** программаас санаа авч дараах зурагт үзүүлсэн таван дүрснээс сонгож зурдаг программ бичнэ үү. Сонгох дүрснүүд нь дэлгэцийн зүүн талд байрлана.



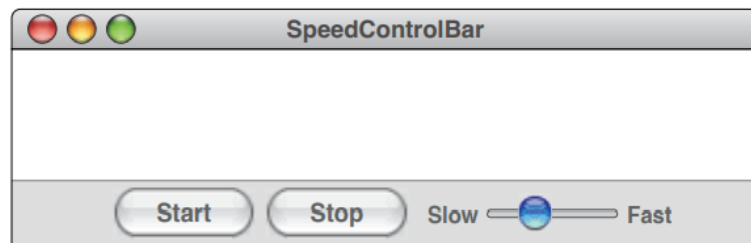
Аль нэг дүрс дээр дарахад (click) тэр дүрс сонгогдох бөгөөд сонгох дүрснүүдээс баруун талын хоосон байрлалд зурах үйлдлийг хийхэд (drag) сонгогдсон дүрс зурагддаг байна. Аль нэг зурагдсан дүрс дээр дараад чирэх үйлдлийг хийхэд тухайн дүрс хулганыг даган байрлалаа өөрчилдөг байна. Тухайн дүрсийг чирэхгүйгээр дээр нь дарах (click) үед давхарлан өрөх дарааллын хамгийн наана буюу нүүрэнд сонгогдоно.

6. Зураг 10-9-д үзүүлсэн **DrawStarMap** программ нь **JComboBox** ашиглан зурах одны өнгийг сонгодог болсон. Тус программ нь зурах өнгөө сонгохдоо **JComboBox**-ийн одоогийн сонгогдсон текстийг дараалсан олон тооны **if** ашиглан шалгаж харгалзах **Color** объектыг буцаадаг. Энэ нь ахин шинээр өнгө нэмэх үед **initColorChooser** болон **getCurrentColor** гэсэн хоёр өөр газар зэрэг өөрчлөлт хийх шаардлагатай байгаа нь сул тал юм. Эдгээр газрын аль нэгт нь л өөрчлөлтийг буруу хийвэл программ зөв ажиллах боломжгүй болно.

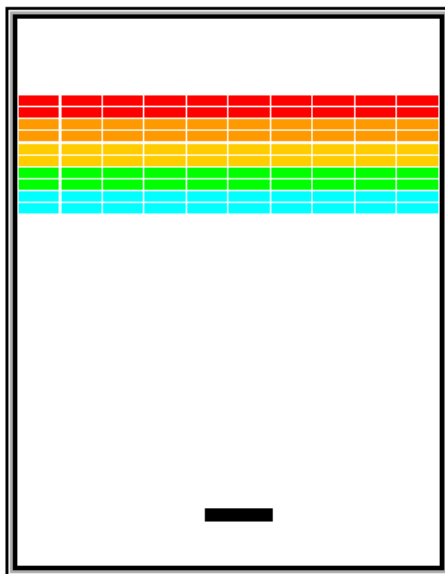
Тус сул талыг засах илүү үр дүнтэй арга байдаг. Хэрэв **JComboBox**-д Жавагийн өнгөнүүдийг **if** заавар ашиглахгүй шууд төрөл хөрвүүлэлт хийх зорилгоор шууд оноовол дээрх асуудал шийдэгдэх боломжтой. **JComboBox** нь олон элементүүдээс тогтох бөгөөд өөрийн сонголтод гарах текстүүдийг бичихдээ элементийнхээ **toString** аргыг ашиглан текст болгож бичдэг бөгөөд текст нь богино, уншууртай байх хэрэгтэй байдаг. **JComboBox**-д Жавагийн өнгөнүүдийг шууд оноож өгвөл **RED** гэсэн текстийн оронд дараах текст харагдах болно.

`java.awt.Color[r=255,g=0,b=0]` Дээрх текстийг зарим хүмүүс улаан өнгө гэж мэдэх боловч зарим нь мэдэж чадахгүй. Тэгвэл тус асуудлыг шийдэхийн тулд **java.awt**-ийн **Color** классаас удамшсан (**extend**) **LabeledColor** гэсэн классыг тодорхойл. Тус классын **toString** арга нь өнгөний нэрийг буцаадаг байна. Тус сайжруулсан классыг ашиглахдаа: `colorChooser.addItem(new LabeledColor(Color.RED, "Red"))`; гэж ашиглана. **DrawStarMap** классыг **LabeledColor** классыг ашигладаг байхаар өөрчилнө үү.

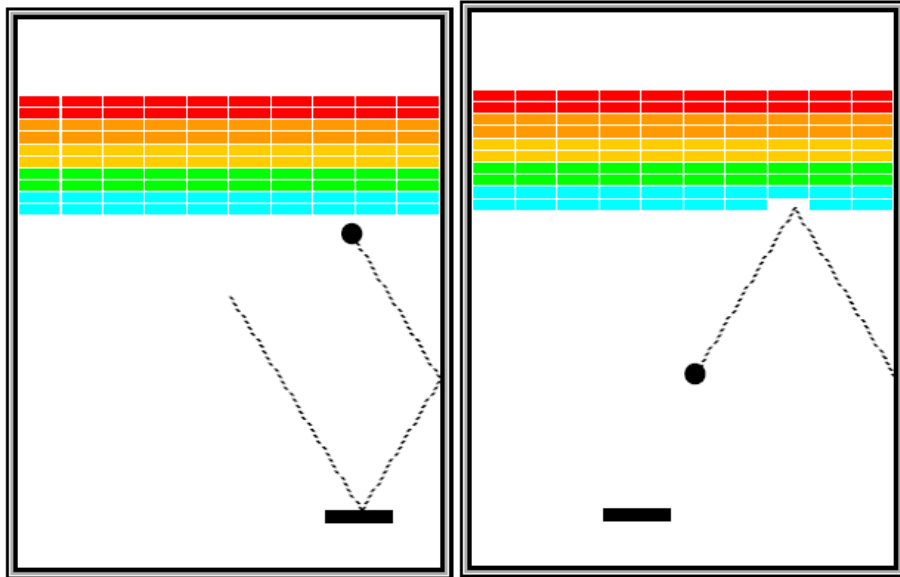
7. **DrawStarMap** программыг дараах байдлаар сайжруулна уу.
- Хэрэглэгч одыг чирч хөдөлгөх боломжтой байна. Хэрэв хулганаар дарах үйлдэл хоосон газар хийгдвэл өмнөх шиг шинэ од зурна. Харин ямар нэг од дээр дарах үед Зураг10-4 дэх **DragObject** программ шиг тухайн сонгогдсон одыг чирэх боломжтой болно.
  - **KeyListener** ашиглан дөрвөн сумны тусламжтайгаар сонгогдсон байгаа одыг тухайн сумны чиглэлд нэг нэг цэгээр зөөдөг байна. Мөн **DELETE** товчийг дарахад сонгогдсон од устдаг байна.
  - **TextField** ашиглан (удирдлагын хэсэгт нэмж байршуулна) одны нэрийг оруулах боломжтой болгоорой. Хэрэглэгч **TextField** дээр нэр бичээд **ENTER** товч дарахад үед тухайн нэр сонгогдсон одны баруун талд гардаг байна. Тус текст нь тухайн агшинд сонгогдсон өнгөтэй байна. Тухайн нэр нь тухайн одтой хамт хөдөлдөг байна. (Өөрөөр хэлбэл од нь нэртэй байх бөгөөд тус нэрийг өөрчлөх боломжтой байна. Хэрэв нэр зааж өгөөгүй бол "" буюу хоосон нэртэй байна.)
8. График программын **init** арга дотор дараах зурагт харуулсан шиг хурдыг хянах хяналтын талбарыг бэлдэж өгнө үү.



9. Бүлэг 4-т гүйцэтгэсэн **BouncingBall** программд хурдны хяналтын талбарыг нэмж харгалзах өөрчлөлтийг гүйцэтгээрэй. Програмын `run` аргад бөмбөг хөдөлгөх код хэвээр байх бөгөөд хэрэглэгч `start` товчлууур дээр дарах үед бөмбөг хөдөлж эхлэх бөгөөд `stop` товч дарах үед зогсдог байна. Үүнийг хэрэгжүүлэхийн тулд нэг **boolean** хувьсагч ашиглах бөгөөд түүний утга **true** эсвэл **false** байгаагаас хамаарч бөмбөгөө хөдөлгөх хэрэгтэй. Бөмбөгийг хөдөлж буй хөдөлгөөний хэрэглэгчид харуулахын тулд **pause()** аргыг ашигласан хэвээр байхыг мартаж болохгүй. **Slider**-ийг ашиглан бөмбөгний хөдлөх хурдыг хурдлуулах эсвэл удаашруулан тохируулдаг байна.
10. Apple компанийг үүсгэн байгуулагч Steve Jobs –санаачилсан **Breakout** тоглоомыг хэрэгжүүлнэ үү. Тус тоглоом нь эхлэхдээ дараах зурагт харуулсанчлан дэлгэцийг бэлтгэсэн байна. Дэлгэцийн дээд хэсэг дэх төрөл бүрийн өнгө бүхий тэгш өнцөгтүүд нь тоосго бөгөөд дэлгэцийн доод хэсэг дэх бараан тэгш өнцөгт нь тавцан юм. Тавцан нь босоо тэнхлэгийн дагуу хөдлөхгүй бөгөөд хулганын хөдөлгөөнийг даган баруун зүүн тийш холхино.



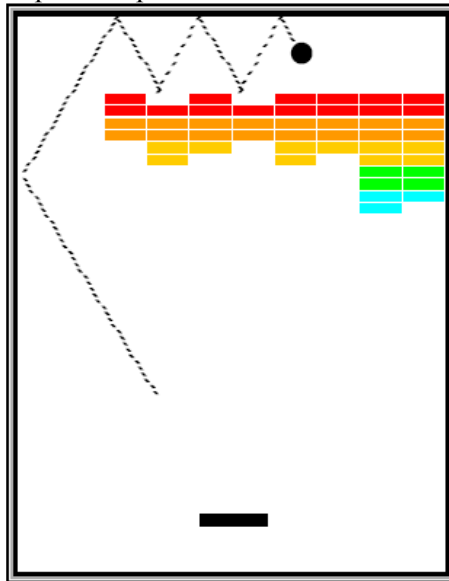
Тоглогч нь 3-н боломжтой(бөмбөгийг 3 удаа алдах эрхтэй). Шинэ боломж эхлэх бүрд бөмбөг нь дэлгэцийн төв хэсгээс санамсаргүй өнцгөөр доош унана. Бөмбөг нь тавцан болон хананд ойх бөгөөд туссан өнцөг болон ойсон өнцөг нь хоорондоо тэнцүү байна. Дараах зурагт бөмбөг хана болон тавцанд ойн явж байгаа замыг тасархай шугамаар дүрслэн харууллаа. Бөмбөг тоосгыг онох үед тоосго устан бөмбөг буцан ойх болно. Тоглогч тавцанг удирдан бөмбөгийг тосон авч ойлгох ёстой.



Тоглоом дуусах дараах хоёр нөхцөл тохиолдоно.

1. Бөмбөгийг тосон авч амжилгүй шаланд хүрвэл нэг боломжоо алдах бөгөөд бөмбөг дэлгэцийн голоос санамсаргүй өнцгөөр доош унаж дараагийн боломж эхэлнэ. Хэрвээ бүх боломж дууссан бол тоглоом дуусна.
2. Хэрвээ бүх тоосго дуусвал тоглогч хожиж тоглоом дуусна.

Аль нэг багана бүрэн устсан тохиолдолд тус баганаар дамжуулан тоосгонуудын дээр гарах боломжтой бөгөөд тоосго болон дээд хана хоёрын хооронд бөмбөг олон дахин ойх боломжтой.



Тус тоглоомд тоглогчийн ур чадварыг тооцох шаардлагагүй бөгөөд зөвхөн бөмбөг газарт унах, хана болон тоосго, тавцанд ойход хангалттай.

Программыг ажиллуулах run арга нь дотор оо хоёр аргыг дуудаж ажиллуулдаг байна. Эхний арга нь тоглоомд бэлтгэх үйлдлийг гүйцэтгэх бол дараачийн арга нь тоглох үйлдлийг гүйцэтгэнэ.

Тоосгоны тоо, хэмжээ, хоорондох зай болон эхний мөр нь цохноос хэр зайд байрлахыг тодорхойлохдоо нэрлэсэн тогтмол ашиглаарай. Зөвхөн тоосгонуудыг цонхны гол байрлуулах х координатын утгыг тооцоолоход л хангалттай. Тоосгонууд нь хоёр хоёр мөрөөрөө хоорондоо адилхан өнгөтэй байх ба тэдгээр нь солонгын өнгөний дарааллаар будагдана. (**RED, ORANGE, YELLOW, GREEN, CYAN.**)

Тавцангийн хэмжээ болон байрлах у координат (дэлгэцийн доод хэсэгт) нь нэрлэсэн тогтмол байна. Программ эхлэхэд тавцан нь х тэнхлэгийн хувьд дэлгэцийн голд байрлана.

Тавцанг удирдахдаа хулганын хөдөлж буй (move) байрлалын х координатыг ашиглана. Тавцанг удирдахдаа тавцан нь цонхонд бүтэн харагдаж байхаар тооцоолоорой.

**GraphicsProgram**-д тодорхойлсон байрлалд байгаа элементийг авахдаа **getElementAt(x, y)** гэж авдаг бөгөөд энэ арга нь x, y байрлалд байгаа объектыг буцаадаг. Хэрвээ ямар нэгэн объект байхгүй бол **null** утга буцаана. Гэхдээ бөмбөг нь зөвхөн нэг цэгээс бүтдэггүй олон цэгүүдтэй байдаг учир бөмбөгний дараах дөрвөн цэг тус бүр дээр шалгах хэрэгтэй.



1. **getElementAt** аргыг ашиглан тодорхой координат дээр **GObject** байгаа эсэхийг шалгана.
2. Хэрвээ **null** биш утга буцаж ирвэл тухайн объект бөмбөгтэй мөргөлдсөн гэсэн үг.
3. Хэрвээ **null** утга ирвэл дараагийн өнцгийн цэгийг шалгах замаар бүх дөрвөн цэгийг шалгана.
4. Хэрвээ бүх өнцөг нь мөргөлдөөгүй бол бөмбөг ямар нэгэн зүйлтэй мөргөлдөөгүй гэсэн үг.

Мөргөлдсөн объектыг хайж олоход дараах аргыг тодорхойлон ашиглах нь тохиромжтой.

```
private GObject getCollidingObject()
```

тус арга нь бөмбөгний мөргөлдсөн объектыг буцаах бөгөөд хэрвээ мөргөлдөөгүй бол **null** утга буцаах бөгөөд дараах хэлбэрээр дуудаж ашиглавал зохимжтой.

```
GObject collider = getCollidingObject();
```

Хэрвээ мөргөлдсөн объект нь тавцан биш бол аль нэгэн тоосготой мөргөлдсөн гэсэн үг юм. Үүнийг дараах аргаар шалгана.

```
if (collider == paddle) . . .
```

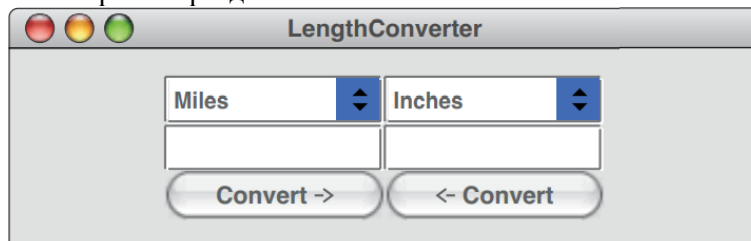
Хэрвээ мөргөлдсөн объект нь тоосго бол түүнийг цонхноос **remove(collider)** гэж хасна.

11. Дараах уртын хэмжээснүүдийн хувьд нэг хэмжээсийг нөгөө хэмжээс рүү хөрвүүлдэг хөрвүүлэгч программ бичнэ үү.

12 inches = 1 foot  
6 feet = 1 fathom  
3 feet = 1 yard

16.5 feet = 1 rod  
5280 feet = 1 mile  
220 yards = 1 furlong

Тус программ нь дараах хэлбэртэй харагдана.



Аль төрлөөс аль төрөл рүү хөрвүүлэхийг тодорхойлохдоо дараалсан **if** ашиглах эсвэл 6-р дасгалд хэрэгжүүлсэн шийдлээс санаа авч болно.

12. Бүлэг 10-т хэрэгжүүлсэн **Calculator** программыг бодит тоон утга оруулах үед ажилладаг болгоно уу. Асуудлыг шийдэхдээ хаана цэгтэй тоо орох боломжтой болон хаана тус тоон дээр тооцоол хийхээ сайтар тунгаан шаардлагатай цөөн хэдэн газар л өөрчлөлтийг хийнэ.