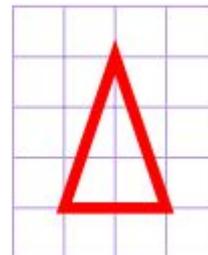
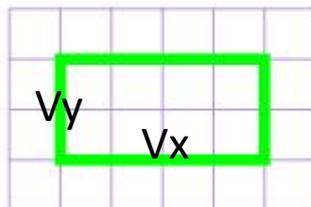


До начала работы

Было на бумаге:



Задание – поместить в квадрат 2x2, и центр картинки должен совпасть с началом координат.

Расчет:

$$\text{aspectFig} = Vx / Vy$$

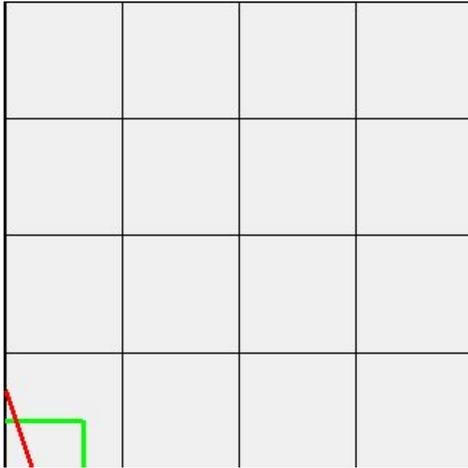
$$\text{aspectForm} = 1 // 2x2$$

$$S = (\text{aspectFig} < \text{aspectForm}) ? 2 / Vy : 2 / Vx$$

$$Vx_m = Vx * S // \text{размер по X и Y внутри 2x2}$$

$$Vy_m = Vy * S$$

В программе – команда model



В файле соответствует команде figure после описания рисунка:

первый рисунок

model 2 1 4 2 # центр в точке (2, 1), размеры 4x2

color 0 255 0 # цвет зеленый

thickness 3 # толщина линии 3

path 5 # путь из четырех ребер (пять вершин)

0.5 0.5 # левый нижний угол

0.5 1.5 # левый верхний угол

2.5 1.5 # правый верхний угол

2.5 0.5 # правый нижний угол

0.5 0.5 # левый нижний угол

преобразования и размещения по описанию

figure

```
else if (cmd == "model") { // начало описания нового рисунка
    float mVcx, mVcy, mVx, mVy; // параметры команды model
    s >> mVcx >> mVcy >> mVx >> mVy; // считываем значения переменных
    float S = mVx / mVy < 1 ? 2.f / mVy : 2.f / mVx;
    // сдвиг точки привязки из начала координат в нужную позицию
    // после которого проводим масштабирование
    initM = scale(S) * translate(-mVcx, -mVcy);
    figure.clear();|
```

Следующий шаг – разместить картинку в прямоугольник заданного размера

Условие - минимальный размер прямоугольника 2x3.

Следовательно, надо пересчитать коэффициент масштабирования из 2x2 в 2x3.

$aspectFig = Vx_m / Vy_m$ // размер в 2x2

$aspectForm = Wx / Wy$; // $Wx = 2, Wy = 3$

$S = (aspectFig < aspectForm) ? Wy / Vy_m : Wx / Vx_m$;

Этот коэффициент должен быть применен к картинке до начала основного рисования.

Расчет в model

$$V_x = 4 \quad V_y = 2$$

$$\text{aspectFig} = V_x / V_y \Rightarrow 2 \quad \text{aspectForm} = 1$$

$$S = (\text{aspectFig} = 2 < \text{aspectForm} = 1) ? 2 / V_y : 2 / V_x$$

$$\Rightarrow S = 2 / V_x = 1/2$$

$$V_{x_m} = V_x * S \Rightarrow V_{x_m} = 2$$

$$V_{y_m} = V_y * S \Rightarrow V_{y_m} = 1$$

Расчет для команд отрисовки в файле (след.шаг)

Вариант, когда $W_x = 2$, $W_y = 3$ (фрейм 40x20)

$\text{aspectFig} = V_x_m / V_y_m \Rightarrow \text{aspectFig} = 2/1 = 2$

$\text{aspectForm} = W_x / W_y;$

$S = (\text{aspectFig} < \text{aspectForm}) ? W_y / V_y_m : W_x / V_x_m \Rightarrow (2 < 2/3) ? 3/1 : 2/2 = 1$

Вариант, когда $W_x = 1$, $W_y = 1.5$ (фрейм 20x10)

$\text{aspectFig} = V_x_m / V_y_m \Rightarrow \text{aspectFig} = 2/1 = 2$

$\text{aspectForm} = W_x / W_y;$

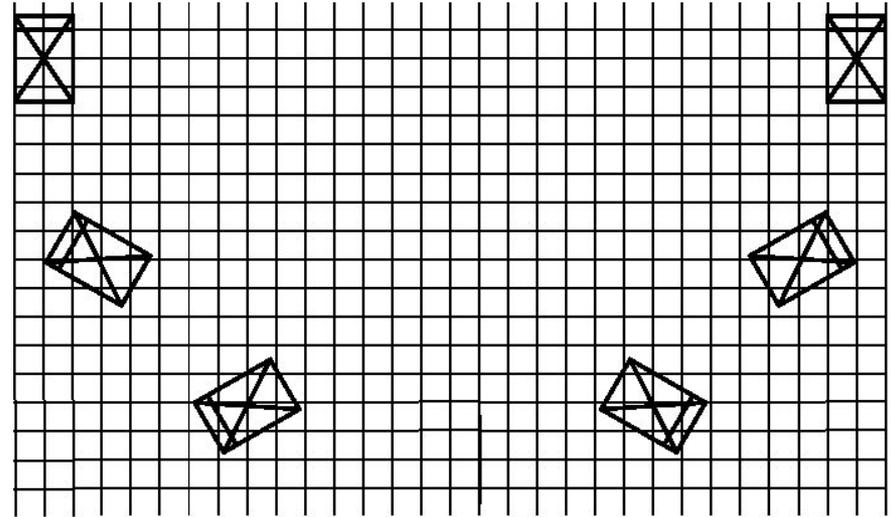
$S = (\text{aspectFig} < \text{aspectForm}) ? W_y / V_y_m : W_x / V_x_m \Rightarrow (2 < 1/1.5) ? 1.5/1 : 1/2 = 0.5$

Повороты, переносы

Задание:

Центры изображений в серии из 6-и изображений расположены на окружности радиуса 14 клеток.

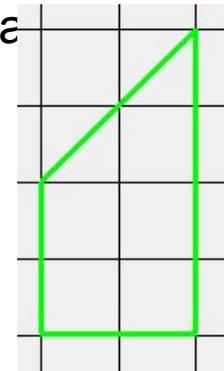
Каждое изображение в серии имеет поворот 30 градусов относительно предыдущего изображения в серии.



Радиус – 14. Сдвиг на 30 градусов по кругу

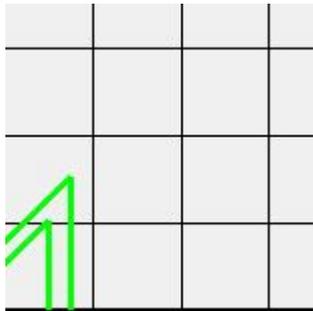
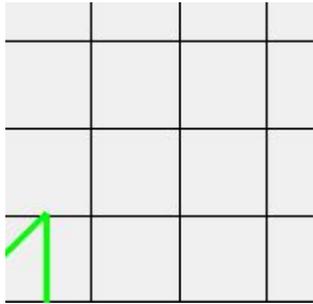
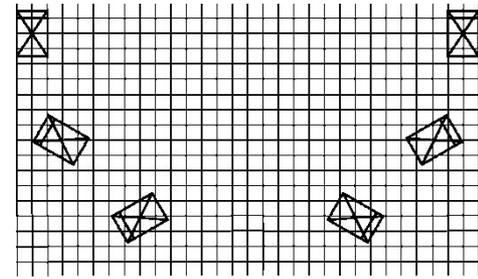
Центр окружности в точке (15, 16)

Разместить не / ниже

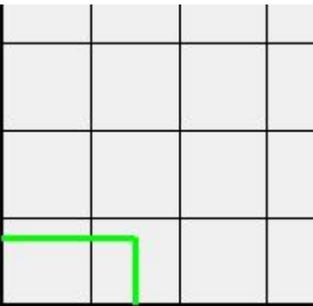


Отрисовка

Стартовое положение –
после ввода, если поставить figure.



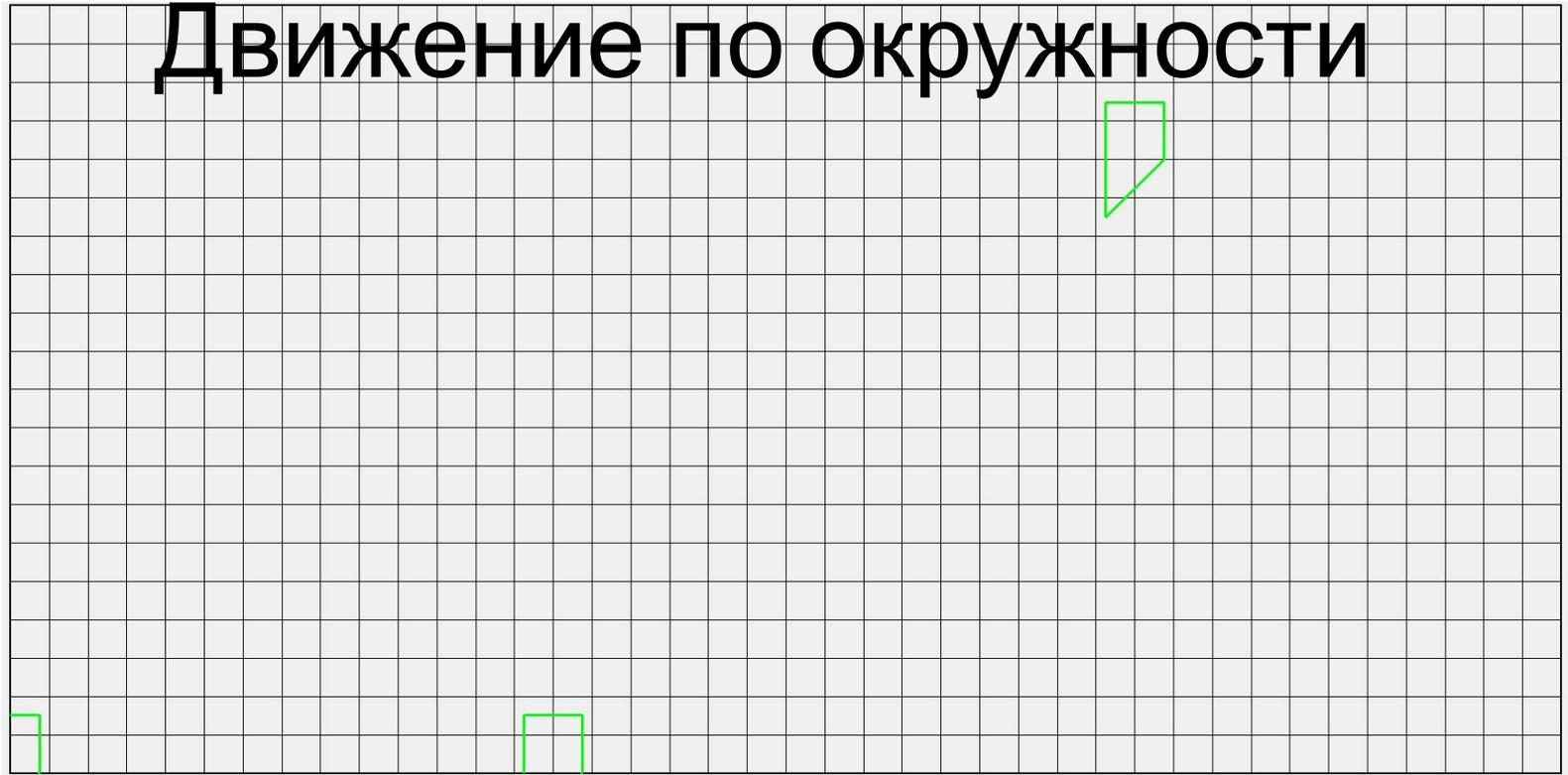
В файле должна быть команда
scale 1.5 - подогнать размер под 2x3



Надо повернуть вверх ногами, чтобы
заняло свое стартовое положение:

Rotate 90

Движение по окружности



Для удобства начнем справа –

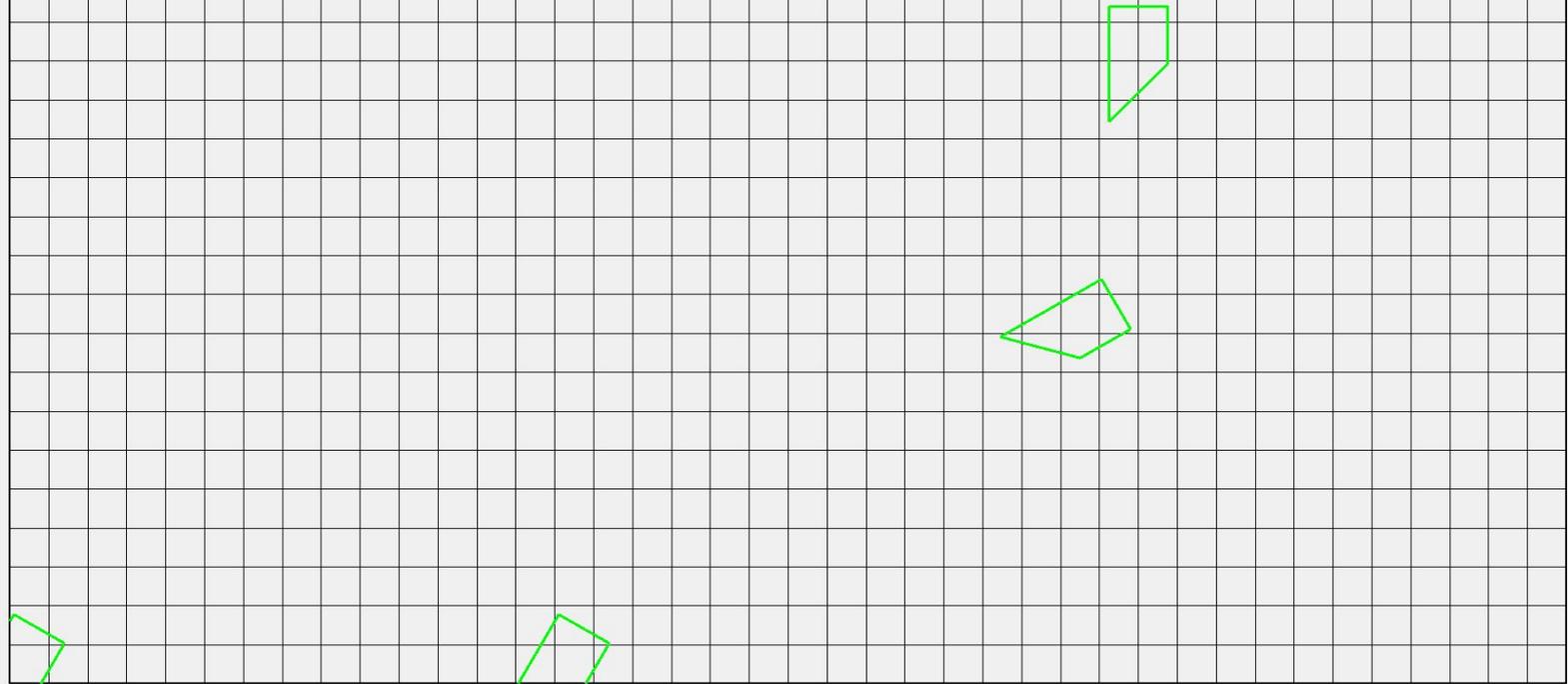
1) Встать на радиус окружности `translate 14 0`

2) Сказать, что ее центр в (15,16)

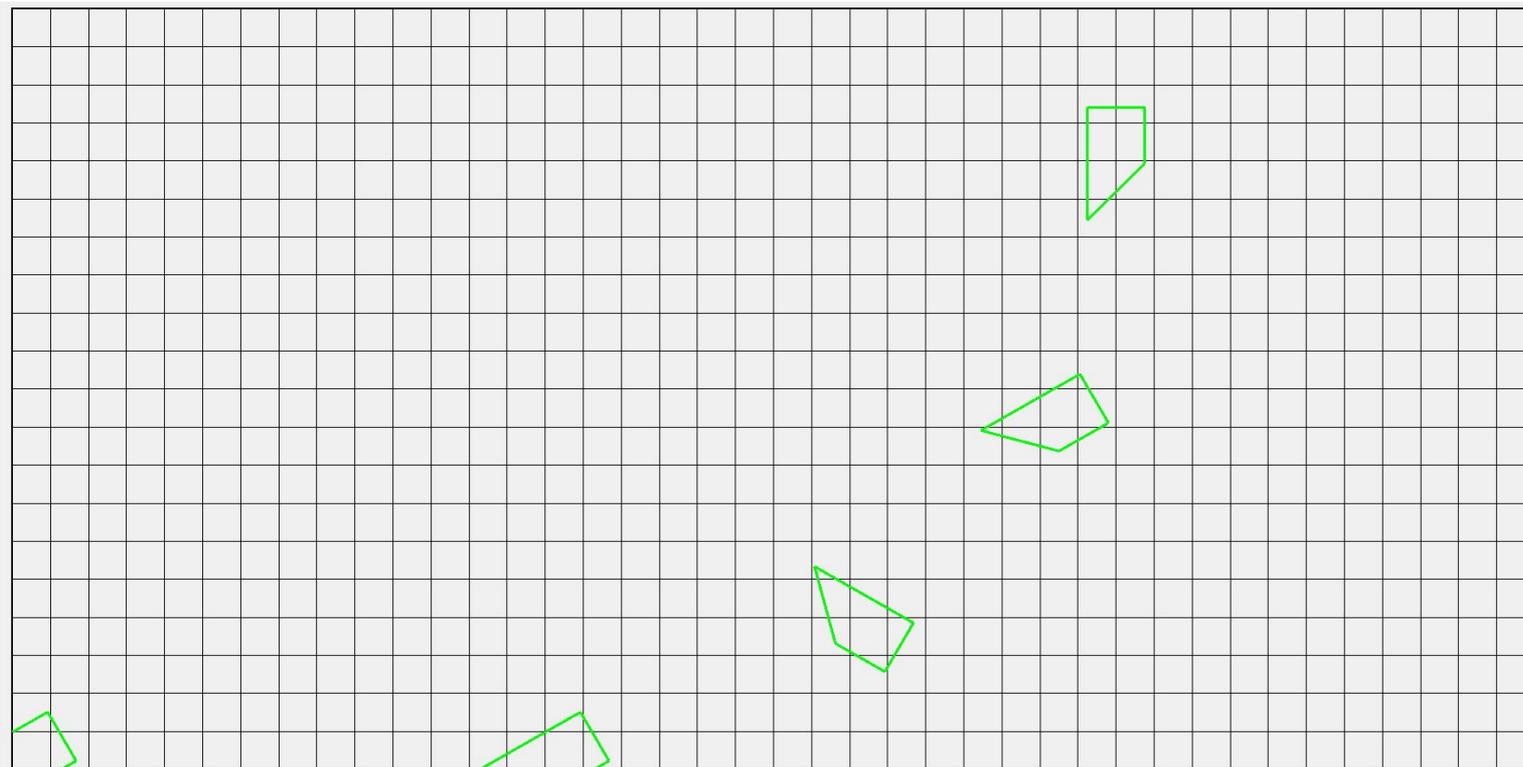
`pushTransform #`

`translate 15 16`

Повороты по окружности



```
#достали из стека стоящую на голове картинку
popTransform
#и повернули ее на 30 градусов вокруг своей оси
rotate -30
pushTransform # запомнили figure
#поставили на окружность и повернули на окружности вокруг центра окружности
translate 14 0 # figure
rotate -30 #figure # показать куда встала - не покажет, так как ушла за экран
translate 15 16 # поставили на место центр окружности
figure
```



#достали из стека повернутую еще на 30

popTransform

#и повернули ее на 30 градусов

rotate -30

pushTransform # запомнили #figure

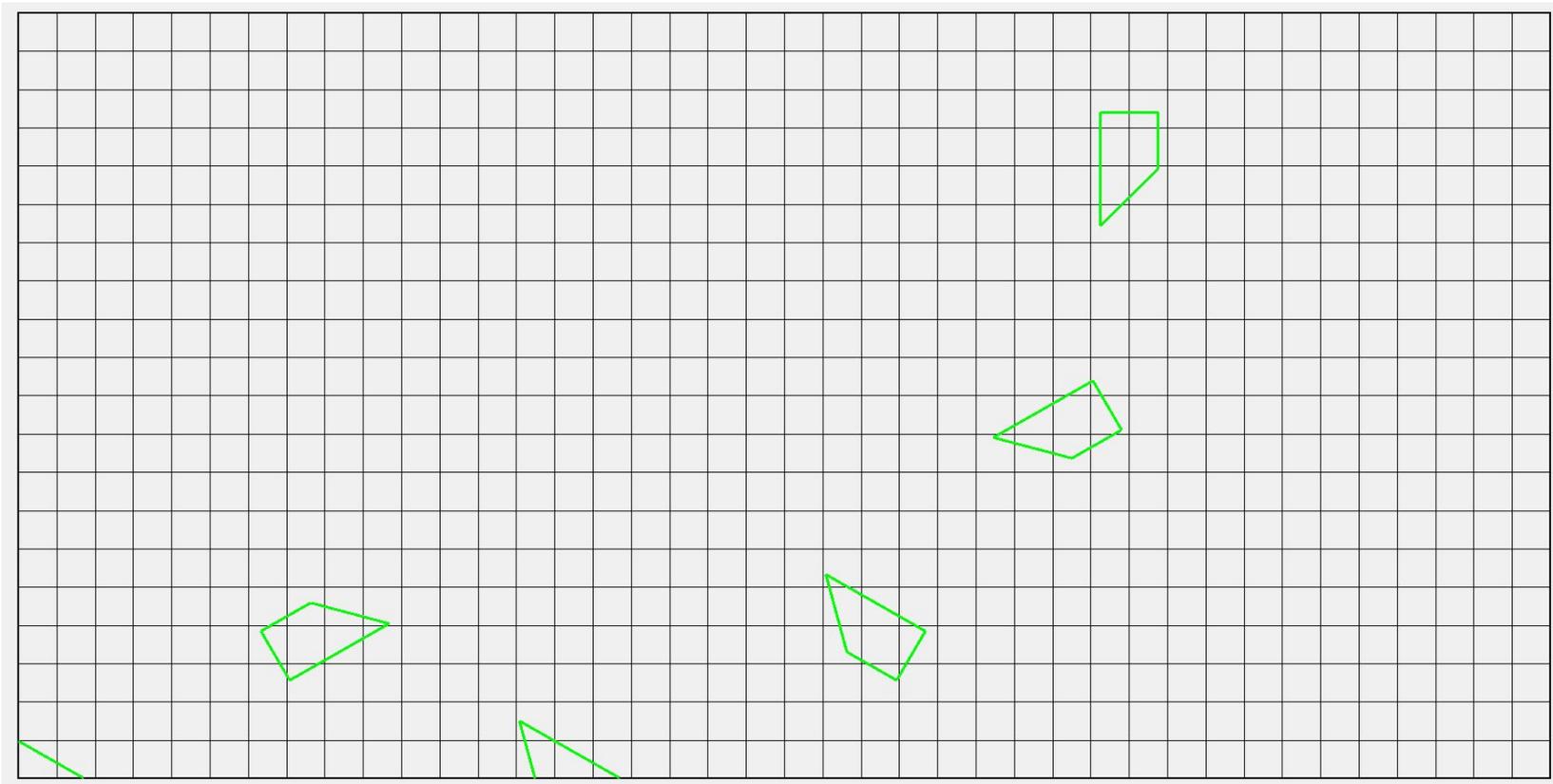
#поставили на окружность и повернули на окружности на -60 градусов (30+30)

translate 14 0 #figure

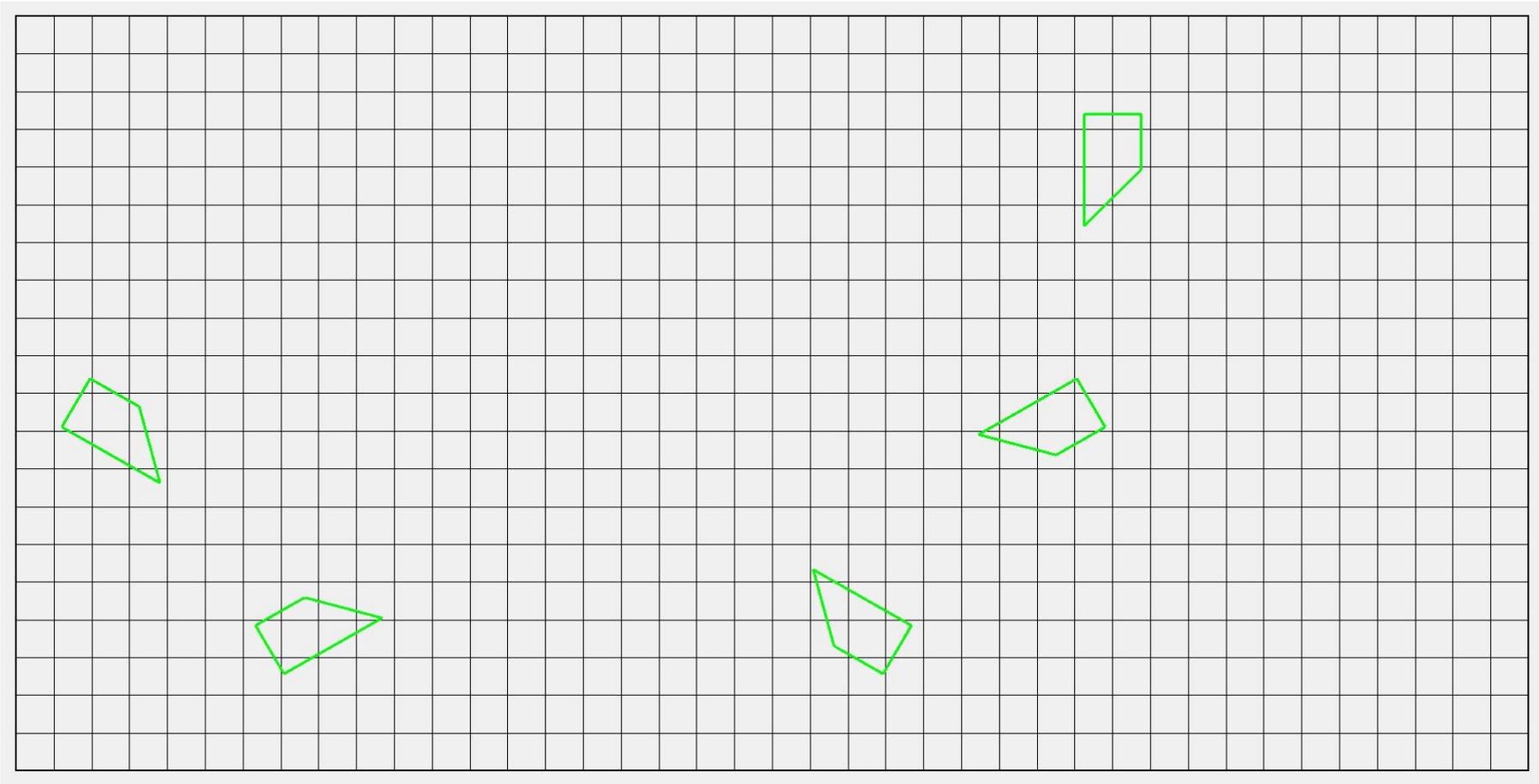
rotate -60 #figure # показать куда встала - не покажет, так как ушла за экран

translate 15 16 # поставили на место центр окружности

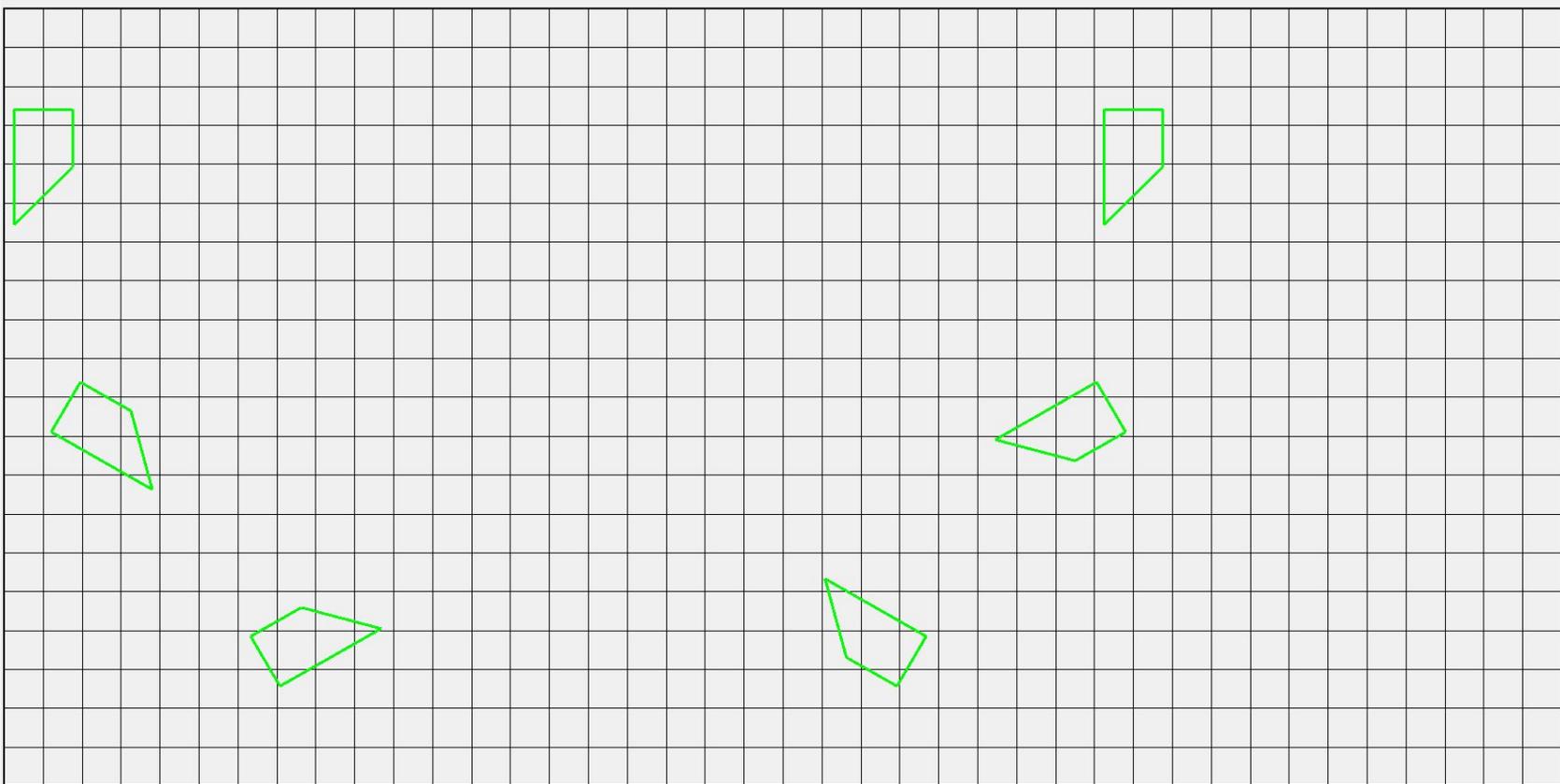
figure



```
#достали из стека повернутую еще на 30  
popTransform  
#и повернули ее на 60 градусов  
rotate -60  
pushTransform # запомнили # figure  
#поставили на окружность и повернули на окружности на -120 градусов (30+30 +60)  
translate 14 0 #figure  
rotate -120 #figure # показать куда встала - не покажет, так как ушла за экран  
translate 15 16 # поставили на место центр окружности  
figure
```



```
#достали из стека повернутую еще на 60
popTransform
#и повернули ее на 30 градусов
rotate -30
pushTransform # запомнили #figure
#поставили на окружность и повернули на окружности на -150 градусов (30+30 +60 +30)
translate 14 0 #figure
rotate -150 #figure # показать куда встала - не покажет, так как ушла за экран
translate 15 16 # поставили на место центр окружности
figure
```



#достали из стека повернутую еще на 30

popTransform

#и повернули ее на 30 градусов

rotate -30

pushTransform # запомнили #figure

#поставили на окружность и повернули на окружности на -180 градусов (30+30 +60 +30)

translate 14 0 #figure

rotate -180 #figure # показать куда встала - не покажет, так как ушла за экран

translate 15 16 # поставили на место центр окружности

figure

Итоговый алгоритм

- 1) Рассчитать самостоятельно переход от истинного размера картинки к 2×2 , а затем от 2×2 к целевому размеру исходного прямоугольника (в нашем случае был 2×3)
- 2) Если есть повороты вокруг оси радиуса R и центра (a, b) , то
 - 2.1) объект перенести на радиус $(a, 0)$ (т.е перенести начало координат)
 - 2.2) выполнить поворот на заданный угол
 - 2.3) поставить окружность на место с центром в (a, b) (перенести начало координат в заданный центр окружности)
- 3) Если объект дополнительно вращается вокруг своей оси, то перед поворотом по окружности, повернуть его вокруг своей оси, пока его центр находится в точке $(0, 0)$