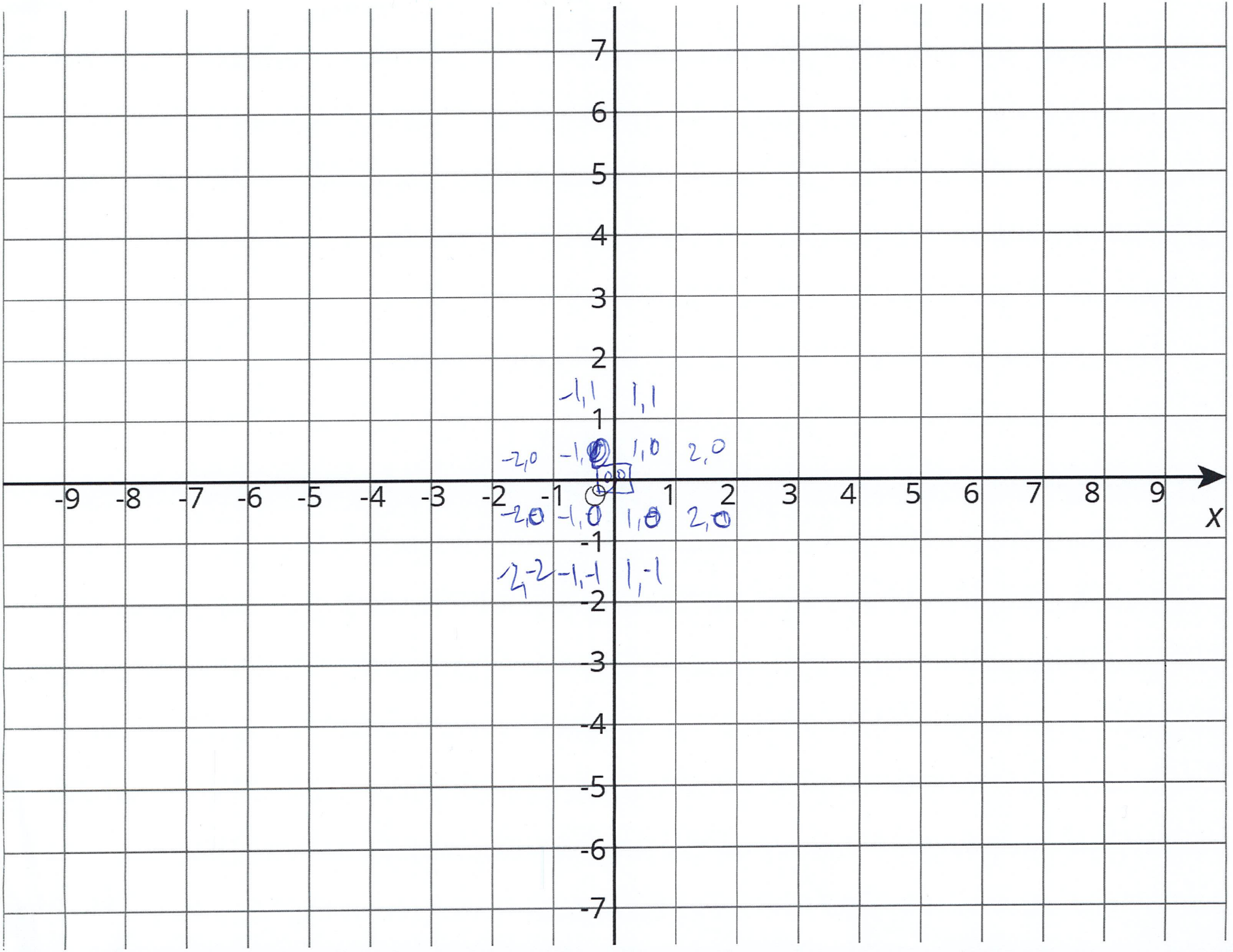
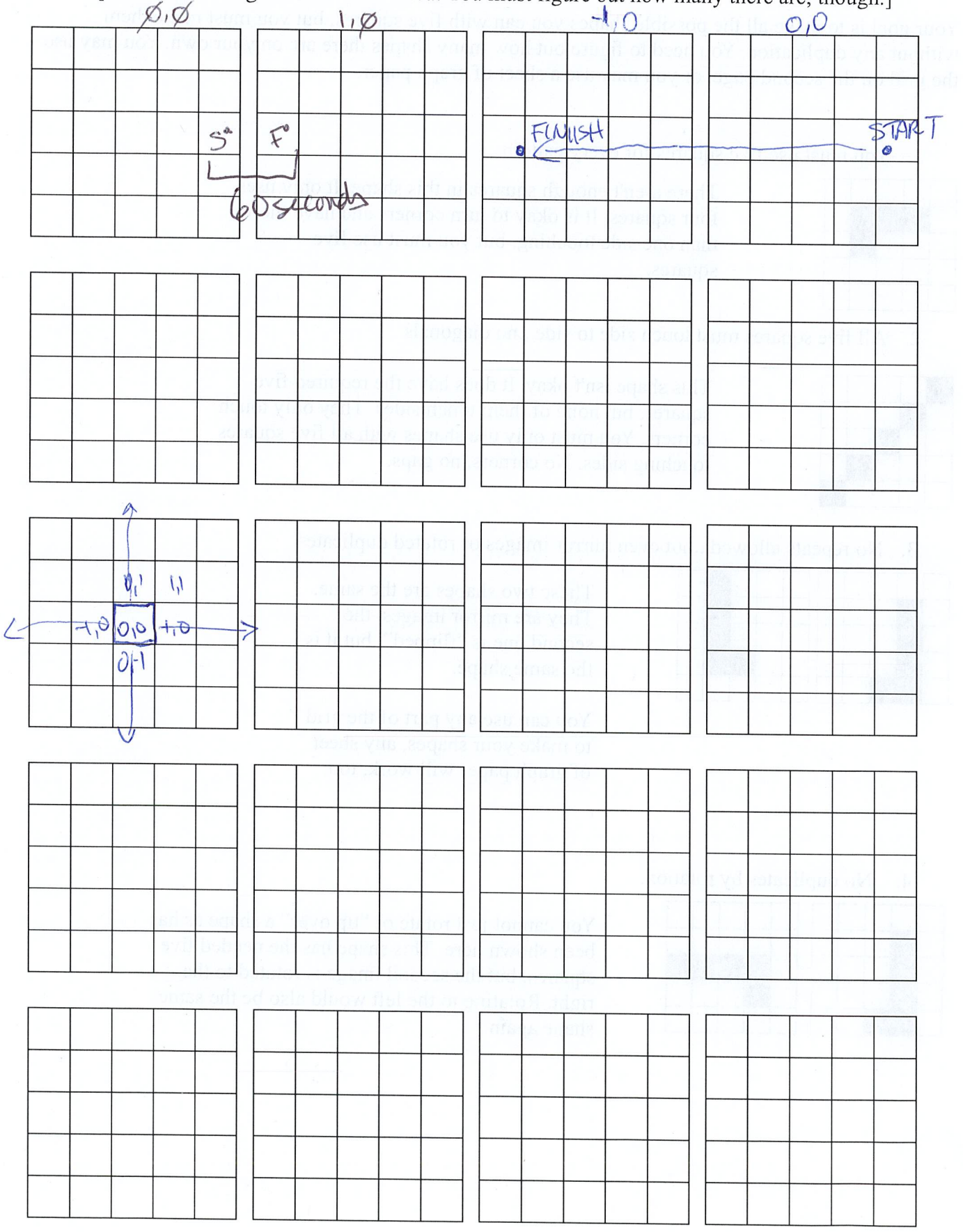

7
6
5
4
3
2
1
-1
-2
-3
-4
-5
-6
-7
-9 -8 -7 -6 -5 -4 -3 -2 -1 1 2 3 4 5 6 7 8 9
X
-1,1
1,1
-2,0
-1,0
1,0
2,0
-2,0
-1,0
1,0
2,0
-2,-2
-1,-1
1,-1

5 Square - a simple puzzler

Student Name _____

Grid Worksheet

[There are more grids than are needed. You must figure out how many there are, though.]



Oct 19/18

A1TZ

A1LZ [A1]

1. Enter zone object name.
2. transition object name

[E1]

A counting system allows for code blocks to execute.

START X, Z

RIGHTX + 1 = X
LEFTX - 1 = X
TOPZ + 1 = Z
TOPZ - 1 = Z

START @ 0,0 , CARTESIEN MAP/GRID

where X, Z is a variable in memory.

Giving control to each block based on the X,Z, allows the destroy & create to work.

A whole section for destroy & create.
Destroy on "Position [LEFT/RIGHT/TOP/BOTTOM][X/Z]"

* A variable to describe a transition is happening.

On change of reposition of X or Z, ~~create~~ use a variable to flag state change!

1. Destroy all objects.
2. Create new objects.

GENERAL IOC FRAMEWORK.

OCT 22/18

DO {

1) TRACK POSITION { VARIBLES USED TO DIRECT MACHINE LOOP INTO CASE [A-Z].

SELECT CASE:

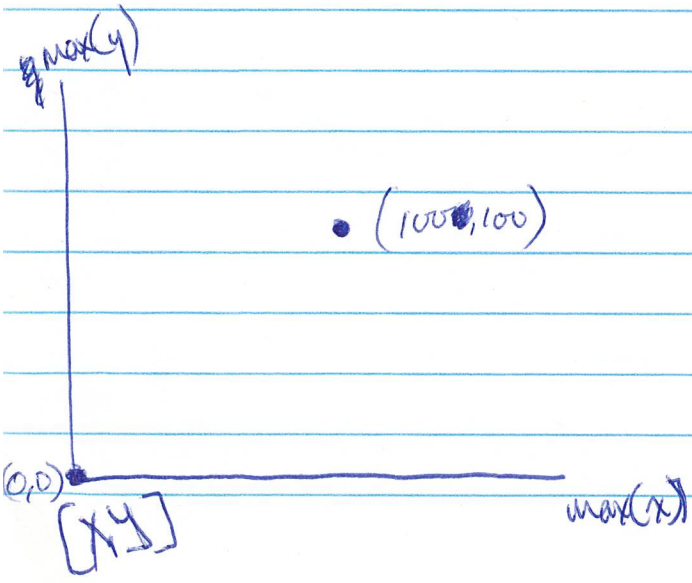
2) GIVE CONTROL TO CODE BLOCK

(A) { USE BLOCK INSIDE UNTIL BREAK.

(B) { LOOP INSIDE UNTIL BREAK.

TBLR

} Loop ;



① define markers.
E1BM
A5RM

② define zones
A1TZ
A1LZ

DESIGN PATTERNS Oct 23/18.

Each sector is created and destroyed base on named & un-named objects.

- A) The named objects are easy to destroy programatically.

* - B) The unnamed objects are not so easy.
  ? + needs working solution here.
  ! solved using destroy all !

Oct 24/18.

Installing, Repairing & Breaking Down of new systems.

Design Patterns which can be used with any new system.

A) Set flag as off for a system

B) Some triggers the install/repair

C) Some triggers the breakdown of system.

+ Similar pattern CREATE/DESTROY when moving to edge of grid.

Oct 26/18

TRACK OBJECT ACROSS MULTI SECTOR; ON COURSE FROM POINT A-B.

WE CAN ~~[illegible]~~ SET:

- Sector location START (x,z) + relative x,y pos.
- Know the object
- Sector location END (x,z) + ~~radii~~ relative x,y pos
- we can direct a object in a direction, set speed
  add-ai name, value1, value2 , type = DIE-THROTTLE
  set exitPoint X, Y, Z
- we can use a timer to figure out distance travelled and when ~~one~~ object should appear.

WHAT WE KNOW:

- an object can be can set an object in a direction & speed.
- we can set a start position sector (x,y) + grid (x,z)
- we can set a destination position sector (x,y) + grid (x,z).
- we can set a timer to determine the object's position. (virtual actualization / estimated sector + grid pos)
- ~~atan2 (Δy, Δx) × 180 ÷ Math.PI (π)) = θ~~

These are all virtual ; background calculations, it can be actualized when the conditions all meet.

Oct 29/18

START (0,0) TIMER: 60 seconds
END (1,0) cross the sector.

- create named object @ x,y,z, angle (@ of ATAN2)
- set timer
- the results are the object appears in the sector based on the timer.

ie: 0-30 sec @ Sector (0,0) ASSERT OUTCOME
31-60 sec @ Sector (1,0)

On create of sector (0,0)
A) evaluate if the object should appear ?

B) its current location, speed & angle.

c)

On destroy of sector, remove named object.

Two calculations:
SECTOR CALC:
1) Position based on throttle value and time
WE KNOW SPEED, START & END SECTOR

GRID CALC: Current position based on time?

S(50,000, 50000) F(0, 50,000)
(x1, y1) (x2, y2)

@ Throttle 1.0, speed of 700m in 10seconds
70 m/s

(FIXED CONSTANT) Throttle = 0.4 , 10 m/s
FORMULA Cargo

d
vt

Hence: @60 seconds the vessel should move 10x60 = 600
600 meters.

- 60 seconds counts down
- start ~~posit~~position 50,000, 50,000
- angle 270°
- destination position 60,000, 50,000
- Speed 10 distance units per second.
- @ 60 seconds, the new position will be 50,600, 50,000.

Thus @ 10 seconds 50,100, 50,000, angle 270°.

$$x(t) = x_1 + vt\cos(\theta)$$
$$y(t) = y_1 + vt(\cos(\theta)) \quad [vt] \text{ @ } \theta, 1$$

Cos is not available, must use first principles to create a cosθ function to determe x(t), x(y).

* for basic functionality, use 90° angles only.

Oct 29/18

Scale Down, before Up. Implement using cosθ : sinθ
using 4 directions only.

Cargo object moves bearing 270°, start (10000, 50000)

Destination (20 000, 50000), speed 0.4 impulse or 10 m/s.

d / tv

$d = 10\,000 - 20\,000 = -10\,000$

$v = 10$ m/s

$t = dv$

$t(x) = dv$

$t(10) = 10000 \times 10$

$d(t) = t$

$d = \frac{10 \text{ seconds}}{10 \text{ m/s}}$

$d = 1$

$x(t) = x_1 + vt\cos\theta$

$y(t) = y_1 + vt\sin\theta$

@ 270° = ∅ slope.