Gomoku Game: Final project for the honor section
Final project

Will replace the final project
Worths ?? points
Will be hard and
fun (hopefully)
Project Specification: don’t know, but negotiable
What’s gomoku?

Extended version of tic-tac-toe
Reference of Gomoku

* Wiki:
http://en.wikipedia.org/wiki/Gomoku
* Online game:
http://www.citycat.ru/iq/gomoku.html
* Facebook app:
www.facebook.com/apps/application.php?id=121800942304
Demo

Demo!
Library Needed:
pygame
Download
http://pygame.org/download.shtml
import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Gomoku")
screen = pygame.display.get_surface()
pygame.draw.rect(screen, (255, 0, 0), [100,100,100,100])
pygame.display.update()
sample code

import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))

screen size is 400 x 300
sample code

import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Gomoku")
title of the window: Gomoku (make sure you add "")
import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Gomoku")
screen = pygame.display.get_surface()

assign memory to "screen", so we can draw something on it.
sample code

```python
import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Gomoku")
screen = pygame.display.get_surface()
pygame.draw.rect(screen, (255, 0, 0), [100,100,100,100])
draw a rectangle of size 100 x 100, starting from (100, 100) color:
```
import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Gomoku")
screen = pygame.display.get_surface()
pygame.draw.rect(screen, (255, 0, 0), [100, 100, 100, 100])
pygame.display.update()

show the result
import pygame
pygame.init()
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("your title")
screen = pygame.display.get_surface()
do sth...
pygame.display.update()

eyou can download this from www.cse.msu.edu/~chengy
one more example: draw a line

```python
1 import pygame
2 pygame.init()
3 screen = pygame.display.set_mode( (600, 600) )
4 pygame.display.set_caption("Gomoku")
5 screen = pygame.display.get_surface()
6 pygame.draw.line(screen, (255, 0, 0), (0, 0), (600, 600), 1)
7 pygame.display.update()
```
resources

project page: www.cse.msu.edu/~chengy
ask a question: http://stackoverflow.com/search?q=pygame
data representation

What data structure to use to represent the board? dictionary? list? or...?
how to represent cursor?
how to represent black/white pieces?
how to ...
You need to use some code to represent entities on your screen.
data representation: board

board: 2-dimensional list
what is it?
a list of lists
lst = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
lst[0][0] = 1
lst[2][1] = 8
data representation: board

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Goal for this week

given a list "board", draw board(from last time), with black/white pieces.
use functions
e.g.
def draw_board():
draw some lines...
draw pieces
def place_piece(row, col, color):
change board[row][col]
draw_board()
Events

pygame detects keyboard/mouse events
events stored in a list (quick demo)
every event is an object: type, key
how do we check whether 'q' is pressed?
event.type == KEYDOWN
event.key == K_q
check manual for keycode.
check course website for sample code
program flow

use functions!!!
while 1:
  1. detect input, up/down/left/right to move cursor, space to place a piece, q to quit game
  2. update board (2 dimensional list)
  3. test winning (next time)
  4. draw board
  5. draw pieces
test winning: last step

still use function
input: position of last piece: e.g. cursor_x, curso_y
output: True or False
find the number of adjacent pieces in 4 ways.
if the number $\geq$ 5 return True:
else False
  - vertically
  - horizontally
  - diagonally NW-SE and NE-SW
everything together

while 1:
1. detect input, up/down/left/right to move cursor, space to place a piece, q to quit game
2. update board(2 dimensional list)
3. test winning
4. draw board
5. draw pieces
White should be placed at A, B or C automatically by your code.
Simple AI

Attack

White should be placed at A, B automatically by your code.