What the two enumerations are for?

Enumerations are, generally speaking, used to create structures similar to class that only has a limited number of given objects crated by it. It serves the purpose of avoiding unwanted objects created by users.
- CellType: Contains 5 types of cells that can exist on the board.
- SnakeMode: Contains 5 moves that the snake can possibly make on the board.
It is helpful to define these two "classes" as enums because the rules of the game make it only possible to make one of 5 moves and only 5 types of cells on the board, so that nobody can accidentally create anymore of these.

SpampedeBrain

what data the objects of this class contain?

-An object of type SpampedeDisplay to which brain passes the data to be represented on the screen(theDisplay)

-An object of type SpampedeData stores the cell types and other variables in SpampedeBrain (theData)

-number of frames displayed so far (cycleNum)

- matching direction names with char keys (REVERSE, UP, DOWN, LEFT, RIGHT, AI_MODE, and PLAY_SPAM_NOISE)

what data the objects of this class know how to do?

-declare the game over (gameOver())

- move the game forward one step changing the positions of Body, head, spam cells (cycle(), updateSnake(), advanceTheSnake(BoardCell nextCell), updateSpam())

- changes direction of snake based on user input (keyPressed())

- determine the path for the nearest Spam cell and direct the snake head to it (getNextCellFromBFS(), getFirstCellInPath)
- reverse snake (reverseSnake())

- play sounds that corresponds to events in the game (playSound_meow (), playSound_spam(), playSound_spamEaten())

SpampedeDisplay

what data the objects of this class contain?

- An object of type SpampedeData whose continents will be displayed (theData)

- the display where the board is drawn (theScreen)

- width of the display in pixels (width)

- height of the display in pixels (height)

- a picture of a can of spam(imageSpam)

what data the objects of this class know how to do?

- Re-draws the board, spam, and snake (updateGraphics())

- draws squares in given locations at with given colors (displayGameOver())

-starts an empty board (clear())

- display messages (displayTitle(), displayGameOver())

CellType

what data the objects of this class contain?

- the character that represents every type (displayChar)

what data the objects of this class know how to do?

- How to return the string representing the CellType

SnakeMode
N/A

BoardCell

-How far is the cell from the left border (row)

-How far is the cell from the top border (column)

-type of the cell(myCellType)

-is this cell added to the search queue? (addedToSearchList)

- which cell did we came from when search reached the cell? (parent)

what data the objects of this class know how to do?

- answer questions about its location (getRow() and getColumn())

- answer questions about the type of the cell(isWall(), isOpen(), isSpam() isBody(), isHead())

-return the of the cell (getCellColor())

- Covert a cell's type(becomeSpam(), becomeOpen(), becomeHead() , becomeBody())

- get info like parent cell (getParent()) or presence in the search list (inSearchListAlready())

- return strings that represents data about the cell (toString(), toStringType(), toStringParent())

- modify parent cell and/or presence in search list (setParent(), clear_RestartSearch(), setAddedToSearchList())

SpampedeData

what data the objects of this class contain?

-The cells in the board(boardCells2D)
- directions of movement of the snake (currentMode)
- cells of type BODY (snakeCells)
- The number of non-wall cells in the initial Board (freeSpots)
- cells of type SPAM (spamCells)
- whether of not the game is over (gameOver)

what data the objects of this class know how to do?

- setup the game interface (addWalls(), fillRemainingCells(), placeSnakeAtStartLocation())
- determine if the snack is in AI mode (inAImode()) and set it to AI mode (setMode_AI())
- determine info about the board (getNumRows(), getNumColumns()) and return a cell in a given position (getCell())
- check if there is spam on the board (noSpam()) add more spam cells (addSpam()) and remove spam cells (removeSpam()
- get cells around a given cell ((BoardCell cell), getNorthNeighbor(BoardCell cell) , getSouthNeighbor(BoardCell cell), getEastNeighbor (BoardCell cell), getWestNeighbor(BoardCell cell), getNeighbors(BoardCell center), getRandomNeighboringCell(BoardCell start)) or neighbors of the snake head (getNorthNeighbor(), getSouthNeighbor(), getEastNeighbor(), getWestNeighbor(), getNextCellInDir())
- setDirection of the snack (setDirectionNorth(), setDirectionSouth(), setDirectionEast(), setDirectionWest())
- set initial direction (setStartDirection())
-get detials about the positions of parts of the snake (getSnakeHead(), getSnakeTail(), getSnakeNeck())
- check if game is over (getGameOver()) and set game over (setGameOver())
- get the color of a cell (getCellColor(int row, int col))
- prepare for new search by clearing old search result (resetCellsForNextSearch())