Zoë Gordin (zeg2103) - Language, Manager
Eleanor Murguia (egm2142) - Tester
Nadia Saleh (ns3059) - Systems

ZEN: A Language for Geometric Games

Introduction

We will implement a language that allows users to write games where the graphics are basic geometric shapes. Our language provides built-in types and functions that allow users to draw shapes that will make up classic games such as Snake and Tetris.

ZEN is an object-oriented language with Java-like syntax for creating new objects.

Features

Basic Features

- Primitive types: `int`, `float`, `bool`, `string`,
- Data structures: `List`, `Dictionary`, `Coord`
- Operators:
  - Arithmetic: `+, - , *, /, %`
  - Assignment: `=`
  - Logical: `<, >, ==, !`
  - Boolean: `and, or`
- Control Flow: `if, else, while, for`
- Comments: `#inline comment`
- Keywords:
  - `new` - create new object
  - `class` - indicates a class

Built in Features

- Types:
  - `Circle`
  - `Ngon` (i.e triangle, rectangle, etc.)
  - `Board` (i.e. space where game will be played)
- Functions
  - `random(int max)` - generates a random number between 0 and max
  - `wait(int milliseconds)` - similar to sleep(), takes an integer value and pauses the process for that number of milliseconds
  - `getKey()` – returns an integer that represents what key was pressed by the user

- `print(string output)` - prints output to the terminal, if the item being printed is not a string it is cast to a string

Data Structure Methods

List

```
List testList = []
testList.length # get length
testList.get(x) # get element at x
testList.remove(x) # remove element at x
testList.add(element) # add element to list
```

Dictionary

```
Dictionary testDict = {key: value}
testDict[key] = value # assign value to a key
testDict[key] # return value
testDict.remove(key) # removes entry from dictionary
```

Coord

```
Coord testCord = (x,y)
testCord.getX()# get x coordinate value
testCord.getY()# get y coordinate value
```

Circle

```
Circle testCircle = new Circle(x, y, radius)
testCircle.getX() # get x coordinate value
testCircle.getY() # get y coordinate value
testCircle.setX(x)# set x coordinate value
testCircle.setY(y)# set y coordinate value
```

Ngon

```
Ngon testNgon = new Ngon(x, y, width, height)
testNgon.getX() # get x coordinate value
testNgon.getY() # get y coordinate value
testNgon.setX(x)# set x coordinate value
testNgon.setY(y)# set y coordinate value
```

Board

```
Board testBoard = new Board()
testBoard.place(element) # place element on board
testBoard.remove(element) # remove element from board
testBoard.get(x,y) # get element at point on board
```

Sample Program

Below is an example of program that could be written in ZEN. It is a simple program to implement an endless version the game Snake: a game in which the user moves a line which grows every time it encounters a square and tries to avoid hitting the line while moving around the screen. In our version, there is no way to lose the game, the snake just continues to grow with each food it eats.

```
class SnakePiece {
      Coord location;
      String direction;

      SnakePiece(Coord loc, String dir) {
            location = loc;
            direction = dir;
      }

      advance() {
            if (direction == “up”) {
                  location.setY(y+1);
            } else if (direction == “down”) {
                  location.setY(y-1);
            } else if (direction == “left”) {
                  location.setX(x-1);
            } else if (direction == “right”) {
                  location.setX(x+1);
            }
      }

      setDirection(String s) {
            direction = s;
      }

      getDirection() {
            return direction;
      }

      getLocation() {
            return location;
```

```
    }

    toShape() {
        return new Ngon(1, 1, x, y);
    }
}

class Food {
    Coord location;

    Food(Coord loc) {
        Location = loc;
    }

    toShape() {
        return new Circle(1, x, y);
    }
}


class PlayGame {
    List snakeSquares;
    Dictionary turns;
    String currentDirection;
    Food currentFood;

    PlayGame() {
        turns = {};
        currentFood = NewFood();
        currentDirection = “right”;
        snakeSquares = [];
        for(int i = 0; i<3; i++){
            if(i == 0) {
                snakeSquares.add(new SnakePiece(new
            Coord(random(boardWidth), random(boardHeight)),
            currentDirection);
            } else {
                snakeSquares.add(new SnakePiece(new
            Coord(snakeSquares.get(0).getLocation().getX()-i,
            snakeSquares.get(0).getLocation().getY()),
            currentDirection);
                }
        }
```

```
}

newFood() {
     currentFood = new Food(new Coord(random(boardWidth),
random(boardHeight)));

     board.place(currentFood);
     return currentFood;
}

eatFood() {
     board.remove(currentFood);
     snakeSquares.add(new SnakePiece(
          snakeSquares.get(snakeSquares.length() -
     1).getLocation(),
          currentDirection
     );
     newFood();
}


tick() {
     while (true) {
          onKeyPress(getKey());
          for (int i = 0; i < snakeSquares.length(); i++) {
               SnakePiece sp = snakeSquares[i];
               Coord loc = sp.getLocation();
               if (turns[loc]) {
                    sp.setDirection(turns[loc]);
                    if (i == snakeSquares.length() - 1) {
                         turns.remove(loc);
                    }
               }
               sp.advance();
               if (loc == currentFood.getLocation()) {
                    eatFood();
                    newFood();
               }
          }
     wait(500);
     }
}
```

```
    onKeypress(int key) {
        currentSpot = snakeSquares.get(0).getLocation();
        if (key == 24) {
            turns.add(currentSpot, “up”);
        } else if (key == 25) {
            turns.add(currentSpot, “down”);
        } else if (key == 27) {
            turns.add(currentSpot, “left”);
        } else if (key == 26) {
            turns.add(currentSpot, “right”);
        }
    }
}

class Tester {
    main() {
        PlayGame game = new PlayGame();
        game.tick();
    }
}
```