CmpE 160 Project 2

1)Problem Description

Just like classic Snake game, we need to implement a simulation of snakes which are moving around looking for food, feeding, growing up and reproducing. Snakes must not go randomly but they must go to the food consciously.

2)Problem Solution

ui package-The ui package includes the classes that are necessary for creating the main interface of the game.

Game package-The game package includes the classes that are responsible for constructing the bridge between the backend of the game and the visual part. It provides generic grid-based game system.

Playingpiece package- Includes the Piece, Food, Snake and Nodes classes. Food(RED) and Nodes(Body-YELLOW, Head-BLUE) are inherited from the Piece class which implements drawable single pieces. Snake class uses a LinkedList to link and gather Nodes to build a snake, it defines which action a snake choose, and after the command of NatureSimulator it runs reproduce, move, feed or only stay.

Naturesimulator package- Includes LocalInformation class which provides the information of other pieces on the board to any user, its used to choose actions of snakes and foods to prevent conflicts on game board. Action class implements types of action, it provides communication between naturesimulator class and Snake class. NatureSimulator class is the engine of the game all actions and moves conducted here in the timerTick method.

Reproducing: In the timerTick after a snake decide to reproduce a new snake is created using the second constructor of the Snake class which accepts an old Snake as parameter, this constructor links the old snake's last 4 nodes to each other and switch the tail node to a new head node after the new links set the reproduce method in Snake called from the timerTick and it separates the last 4 node from the original snake . And the pieces list and piecemap updated in the timerTick.

Moving: In the timerTick after a snake decide to move only moveTo method of snake is called. This method changes the coordinates of every node to the previous node starting

from tail to the second node, after second node takes place of the head, head moves to given direction. And the pieces list and piecemap updated in the timerTick.

Feeding: In the timerTick after a snake decide to feed the feed method of Snake called and this method makes the snake move on to the nearby food then in the timerTick again old food removed new food assigned to a random location via getRandomFood method and the pieces list and piecemap updated.

Moving Algorithm: Snake choose a biased random direction to move from the freedirections list using the isGettingCloser method in Snake, it checks whether the direction is towards to food or not. The correct directions have 10k possibility whereas other directions have 1k possibility of choosed randomly. This provides snakes to get out from meaningless loops and even to avoid to get too closer to each other so they don't stuck too quick.

timerTick: First cleans the Drawable list then a for loop traverse the snakeslist and each snake makes its move and updates the piecemap and piecelist for each move. After all moves done the drawable list is updated so they can be drawn to game panel.

How to run the program: When we create a naturesimulator instance in the main with sizes of gridpanel and game speed a default snake and food are automatically created, then we start the application window and timer in main which initializes the program. Sizes of gridpanel and game speed are configurable.