1. What is the role of EmptySprite?
   EmptySprite is a subclass of Sprite, used to serve as a container for empty data or no image. For example, in the ImageSprite class and in the split() function, if the coordinates don't fall within the image, an empty sprite is returned.
2. What is the role of MOVE_INTERVAL and INTERVAL_VARIATION?
   MOVE_INTERVAL and INTERVAL_VARIATION are variables which are assigned fixed values. The program uses both the variables to calculate the interval within which the ghost has to move. The interval is calculated using the following code:

   ```
   public long getInterval() {
     return this.moveInterval + new Random().nextInt(this.intervalVariation);
   }
   ```

   Here, we can observe that the interval will always be the MOVE_INTERVAL plus some value not greater than the INTERVAL_VARIATION to make the ghost look more dynamic.
3. if you wanted to add a fruit, which files would you need to change?
   To add a fruit, first you would need to create a separate class for fruit and separate sub classes for the types of fruit. The in the board.txt we would have to add the position where we would like to place the fruits and denote it with a unique char something like 'F'. Then add a case for the fruit (case 'F':) in the addSquare() method in the MapParse.java class. In the case we need to call the function to create fruits classes and assign it to that grid coordinates. So your updated code should look something like below:

   ```
   protected void addSquare(Square[][] grid, List<Ghost> ghosts,
                       List<Square> startPositions, int x, int y, char c) {
     switch (c) {
       case ' ':
         grid[x][y] = boardCreator.createGround();
         break;
       case '#':
         grid[x][y] = boardCreator.createWall();
         break;
       case '.':
         Square pelletSquare = boardCreator.createGround();
         grid[x][y] = pelletSquare;
         levelCreator.createPellet().occupy(pelletSquare);
         break;
       case 'G':
         Square ghostSquare = makeGhostSquare(ghosts, levelCreator.createGhost());
         grid[x][y] = ghostSquare;
         break;
       case 'P':
         Square playerSquare = boardCreator.createGround();
         grid[x][y] = playerSquare;
   ```

```
            startPositions.add(playerSquare);
            break;
        case ‘F’
            Square fruitSquare =  // get a new fruit square class
            grid[x][y] = fruitSquare;
            break;
        default:
            throw new PacmanConfigurationException("Invalid character at "
                + x + "," + y + ": " + c);
    }
}
```

Also, add a new function in the LevelFactory.java class to create new fruit.