5/21/2018

OBJECT ORIENTED PROGRAMMING

Project Report

PROJECT REPORT

# PROPOSAL:

## Name:

A car racing game titled “Need for Sleep”.

## Domain/Scope:

A game that helps the user to develop a sense of speed management and avoiding other cars in a tactful manner in order to win the game.

## Functional Requirements:

* J-Swing graphics library
* AWT Graphics library
* Javax Graphics and Sound library

## Novelty of Idea:

The user acts as a car driver who is supposed to move forward avoiding all the other cars that comes in his way and accelerating and braking in such a way that that he reaches the finish line before a huge traffic jam occurs.

# CONCEPTS OF OOP MAPED:

No program can map the concepts of OOP more effectively than a game. Games extensively apply the concepts of OOP. Our game also maps almost all the concepts of OOP the snippets of which are given in this report. We have used:

* Composition
* Inheritance
* Polymorphism
* Interface Implementation
* File Handling
* Exception Handling

# METHODOLGY:

## Acknowledgements:

We have developed our game by taking help from different forums, tutorials found on the internet for example the collision detection formula and speed-o-meter. ([*http://zetcode.com/tutorials/javagamestutorial/*](http://zetcode.com/tutorials/javagamestutorial/) *)*

## Creation of Graphics:

Our game is GUI based. The map which is displayed in our game has been developed using a crop function. We took a bigger image that contained all the things required in our game world and created sub images with the help of that crop function. The sub images are then played in a desired pattern at the desired coordinates. The images of cars, power-ups and objects that can damage the car are taken from google and then loaded. The speed-o-meter is made using help from the internet and uses sin and cosine functions of Math.

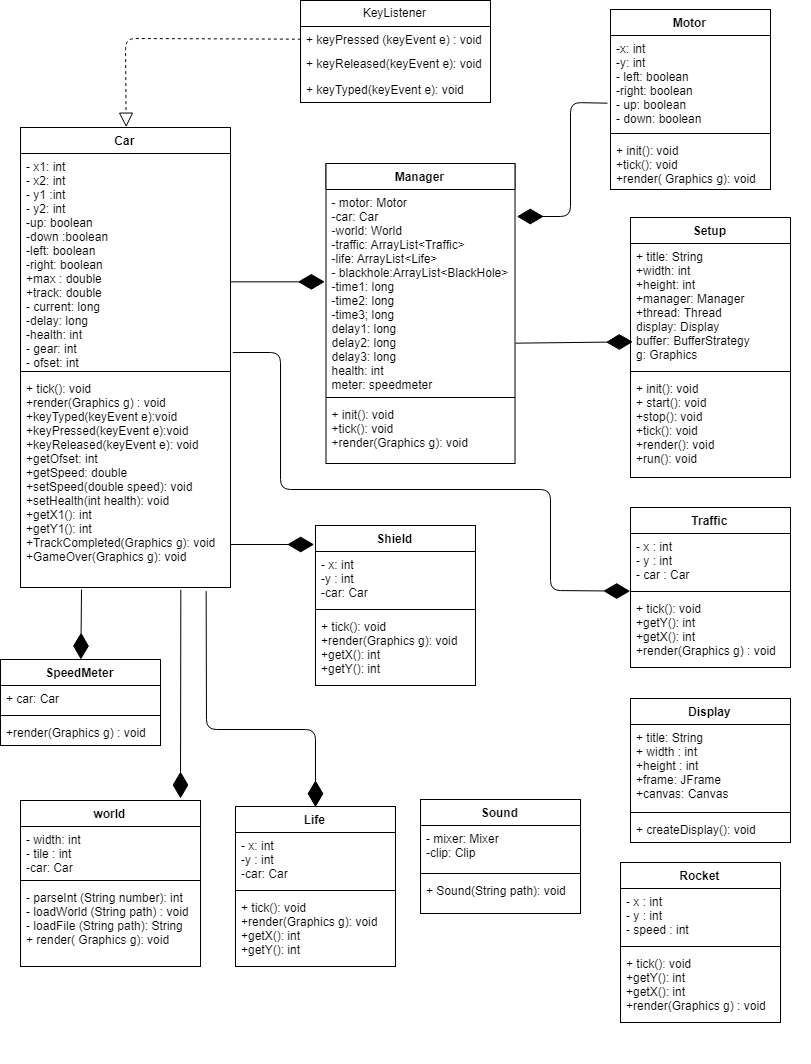
## Other Functions:

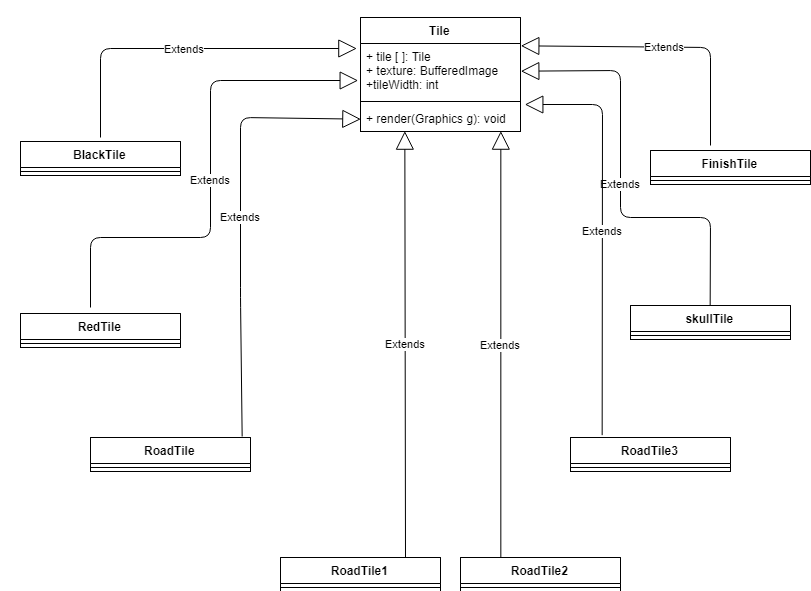
Our game extensively comprises of functions. There is a function for almost everything. Examples include keyPressed and keyReleased methods to handle event when a key is pressed on the keyboard, which help in accelerating, turning and slowing the car down, init methods to setup objects. There’s also a render method which helps in rendering images in the program.

## Most Widely Used OOP Concept:

The most widely used OOP concept in our game is composition followed by exception handling, inheritance and then implementation of interface.

# UML DIAGRAM:





# SOFTWARE/LIBRRARY USED:

## Software Used:

Our code has entirely been written on NetBeans.

## Libraries Used:

Since our game is GUI based, we have used the following GUI libraries:

1. AWT Graphics library
2. JSwing graphics library.

CODE SNIPPETS:

Here are a few instances of the code where the concepts of OOP have been practically implemented.

1. **Implementation of Interface:**

To handle the events generated by the keyboard inputs, the interface KeyListener has been implemented by the class Car. It enables the program to handle the events in favor of the game.

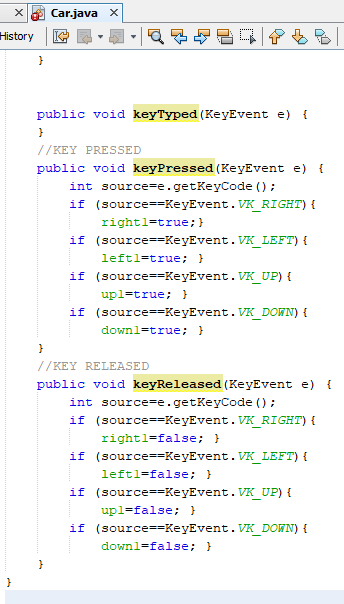


Figure Methods of keyListener implemented.

1. **Composition:**

Composition provides a “has-a” relation between classes which is extremely useful when coding for a game. Many instances of this concept have been mapped such as:

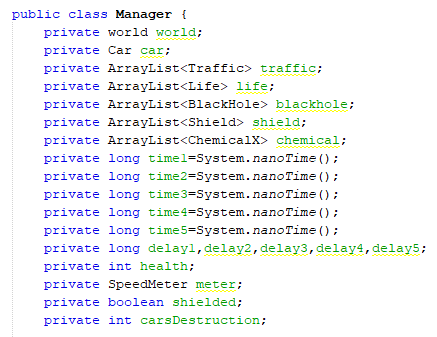


Figure Variables of World, Car and SpeedMeter class included as data members in manager class.

1. **Inheritance:**

When loading images, inheritance concept was extensively used enabling the use of parent methods by the child classes.

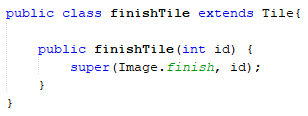


Figure Class finishTile extending parent class Tile.

1. **Exception Handling:**

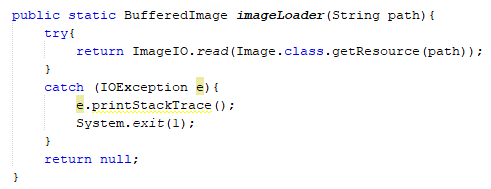
To avoid the crashing of code due to exceptions, exception handling has been used. It ensures the smooth running of game, even if unexpected events occur.

Figure Exception handling used in Image class.

1. **File Handling:**

In order to load graphics, background images and audios, many files have been read by using the methods of file handling such as:



Figure Loading of graphics using file Handling.