



实验2 对象与类

1 实验目的

- (1) 熟悉对象和类的基本表示方法；
- (2) 熟悉对象识别以及特征分析的方法。

2 实验环境

开发环境：JDK 8.0（或更高版本）

开发工具：Eclipse

设计工具：StarUML（或PlantUML等其他工具）

3 实验内容

3.1 对象识别与分析

问题描述：回力车是一种玩具车，后推小车蓄力后释放，小车会往前行驶直到停止。尝试对游戏场景进行分析，识别相应游戏中涉及的对象及对象特征，并实现一个程序来模拟游戏场景。



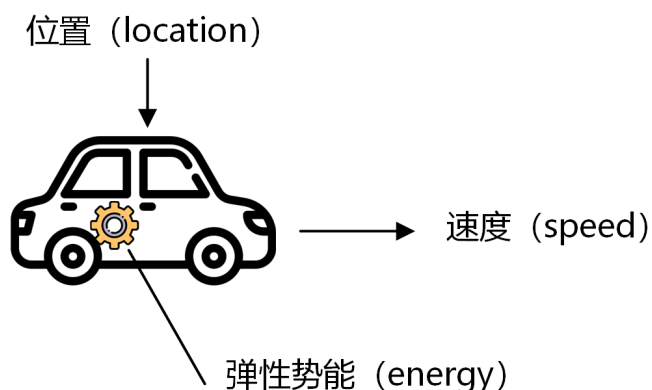
- (1) 识别场景中可能存在的对象。小车（SmallCar）作为主角很容易被识别，控制小车的角色可以设定为玩家（Player），如果需要表示小车的运行状态需要以道路（Road）为参

考。

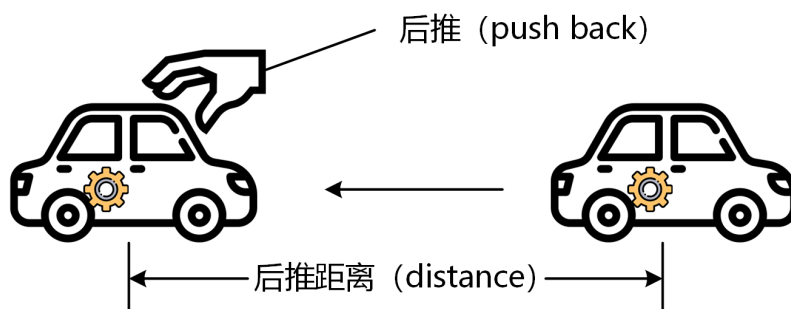


(2) 对象特征及关系分析。

- 小车运行时涉及弹性势能（`energy`）、速度（`speed`）和位置（`location`）三个基本特征。

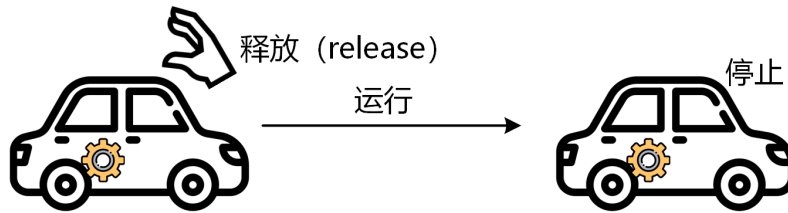


- 玩家可以选择一个小车进行操控。后推（`push`）小车时，小车处于制动（`braked`）状态，根据后推距离积累（`accumulate`）能量，假设后退的距离与蓄积能量之间的转换比例为常量 `DIST_TO_ENERGY`，小车能积累的最大能量为常量 `MAX_ENERGY`。

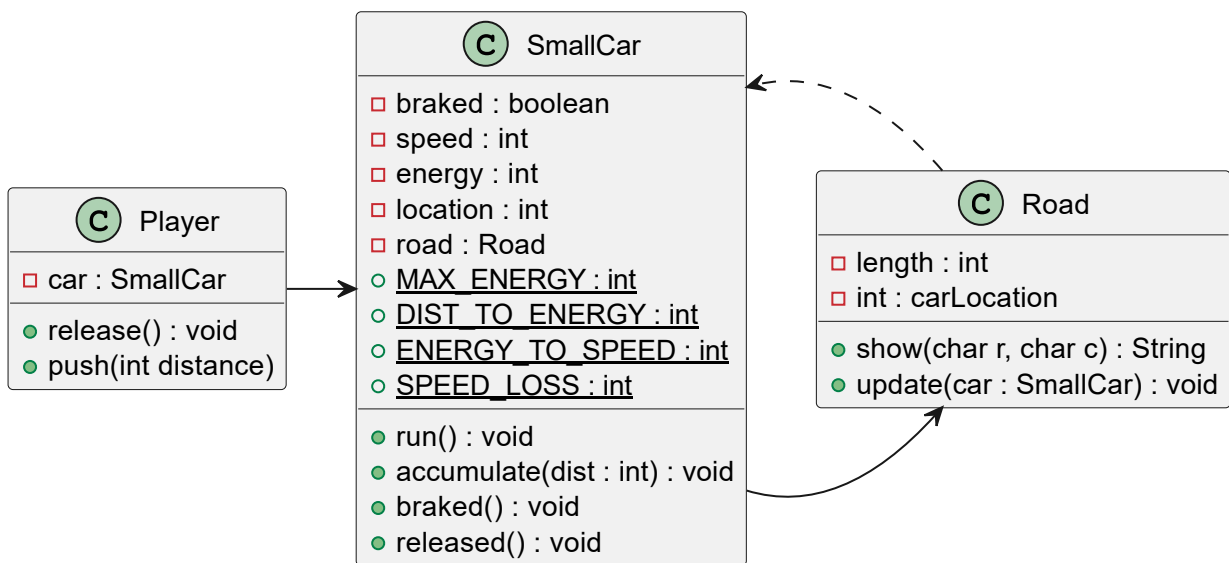


- 玩家释放（`release`）小车后，小车解除制动状态，势能释放产生加速度，小车开始运行（`run`），假设势能转换为速度的比例为常量（`ENERGY_TO_SPEED`）；行驶中的小车也会受到阻力影响而减速，假设阻力导致的速度损失也为常量

(SPEED_LOSS) 。



- 假设道路指定一个空间范围，即长度 (length) ， 小车超出范围后无法显示；
小车在指定道路运行时，实时向道路发送消息更新 (update) 位置
(carLocation) 信息；客户端程序中可以指定符号显示 (show) 小车和道路状态。



(3) 将分析得到的类图转换为实际代码框架，并完善。

```

1  public class SmallCar {
2      private boolean braked;
3      private int energy;
4      private int speed;
5      private int location;
6
7      private Road road;
8
9      public final static int MAX_ENERGY = 100; // 最大蓄能
10     public final static int RADIST_TO_ENERGYTE = 2; // 距离转换为能量比率
11     public final static int ENERGY_TO_SPEED = 3; // 单位能量转换为速度比率
12     public final static int SPEED_LOSS = 2; // 阻力造成的速度损失
13
14     // 小车单次运行
15     public void run() {
16     }
17     // 蓄力
18     public void accumulate(int dist) {}
19     public void braked() {}
20     public void released() {}
21 }
22 public class Road {
23     private int length;
24     private int carLocation;
25     public Road(int len) {
26         carLocation = -1;
27         length = len;
28     }
29     public String show(char r, char c) {
30     }
31     public void update(SmallCar car) {
32     }
33 }
34 public class Player {
35     private SmallCar car;
36     public Player(SmallCar car ) {
37         this.car = car;
38     }
39     public void push(int distance) {}
40     public void release() {}
41 }

```

(4) 依次实现具体的功能代码。

a. 小车单次运行涉及位置和速度变化，状态变化后还需要通知道路更新状态。

```

public void run() {
    if(isRunning()) {
        setLocation(getLocation() + speed); // 移动

        // 能量释放得到加速度
        if(energy != 0) {
            speed += ENERGY_TO_SPEED;
            energy--;
        }
        speed -= SPEED_LOSS; // 阻力, 速度衰减
        if(speed < 0) speed = 0;

        road.update(this); // 道路信息更新
    }
}
// 判断当前状态是否为运行
public boolean isRunning() {
    return !braked && (speed != 0 || energy != 0);
}

```

b. 玩家后推小车, 小车积蓄能量。

```

public void accumulate(int dist) {
    energy += RADIST_TO_ENERGYTE * dist;
    if(energy > MAX_ENERGY) energy = MAX_ENERGY; // 达到能量最大值
}

```

c. 小车制动与制动解除

```

public void braked() {
    this.braked = true;
    this.speed = 0; // 制动后速度为零
}
public void released() {
    this.braked = false;
}

```

d. 玩家的操作

```

public void push(int distance) {
    car.braked();
    car.accumulate(distance);
}
public void release() {
    car.released();
}

```

e. 道路状态更新

```
public void update(SmallCar car) {
    if(car.getLocation() < length) {
        carLocation = car.getLocation();
    }else {
        carLocation = length - 1;
    }
}
```

f. 道路显示

```
public String show(char r, char c) {
    StringBuilder view = new StringBuilder();
    char[] temp = new char[length];
    Arrays.fill(temp, '\b');
    view.append(temp);
    Arrays.fill(temp, r);
    temp[carLocation] = c;
    view.append(temp);
    return view.toString();
}
```

(5) 编写一个客户端程序来模拟小车操作过程。

```
public class Test {
    public static void main(String[] args) {

        SmallCar car = new SmallCar();
        Player player = new Player(car);
        Road road = new Road(100);
        car.setRoad(road);

        // 游戏过程
        player.push(2);
        player.release();
        while(car.isRunning()) {
            car.run(); // 单次的游戏逻辑
            System.out.print(road.show('.', '>')); // 屏幕刷新
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

观察运行结果：

.....>.....

(6) 思考与训练

- 理解和熟悉类的声明、成员的声明和使用等基本知识；
- 尝试按自己的理解进行对象分析、建模和实现；
- 理解给定的代码，并完善注释。

4 实验要求

4.1 实验评价

- 1、必须完成实验3.1的所有步骤，实验中的程序能正常运行。
- 2、遇到问题时能及时与指导老师沟通并解决问题。

4.2 实验报告

本次实验以验证为主，不需要提交实验报告。

5 实验教学录屏

仅供预习和复习参考，实验内容和要求以正式课堂为准。

<https://www.bilibili.com/video/BV1qV411c7nZ/>