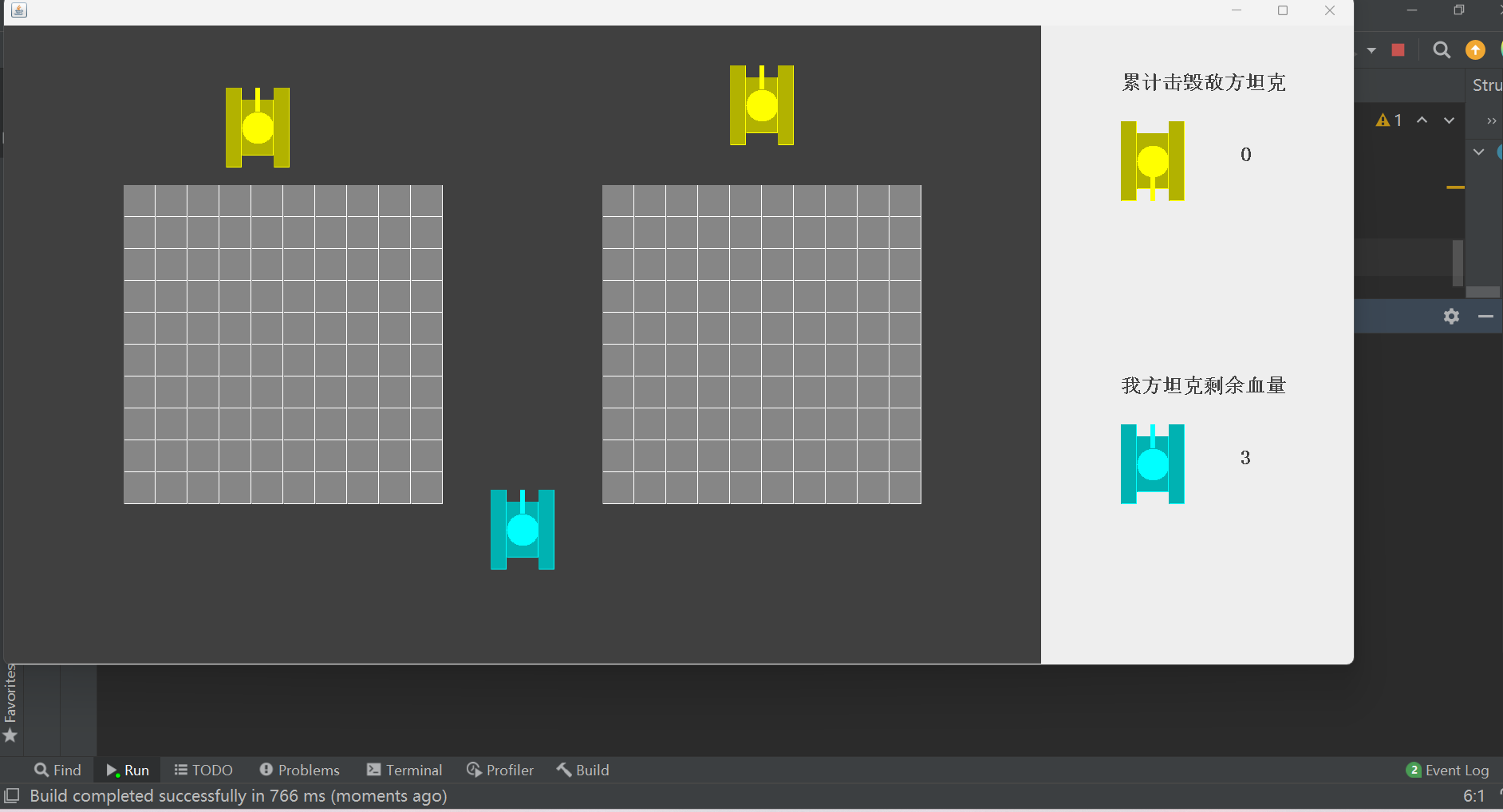
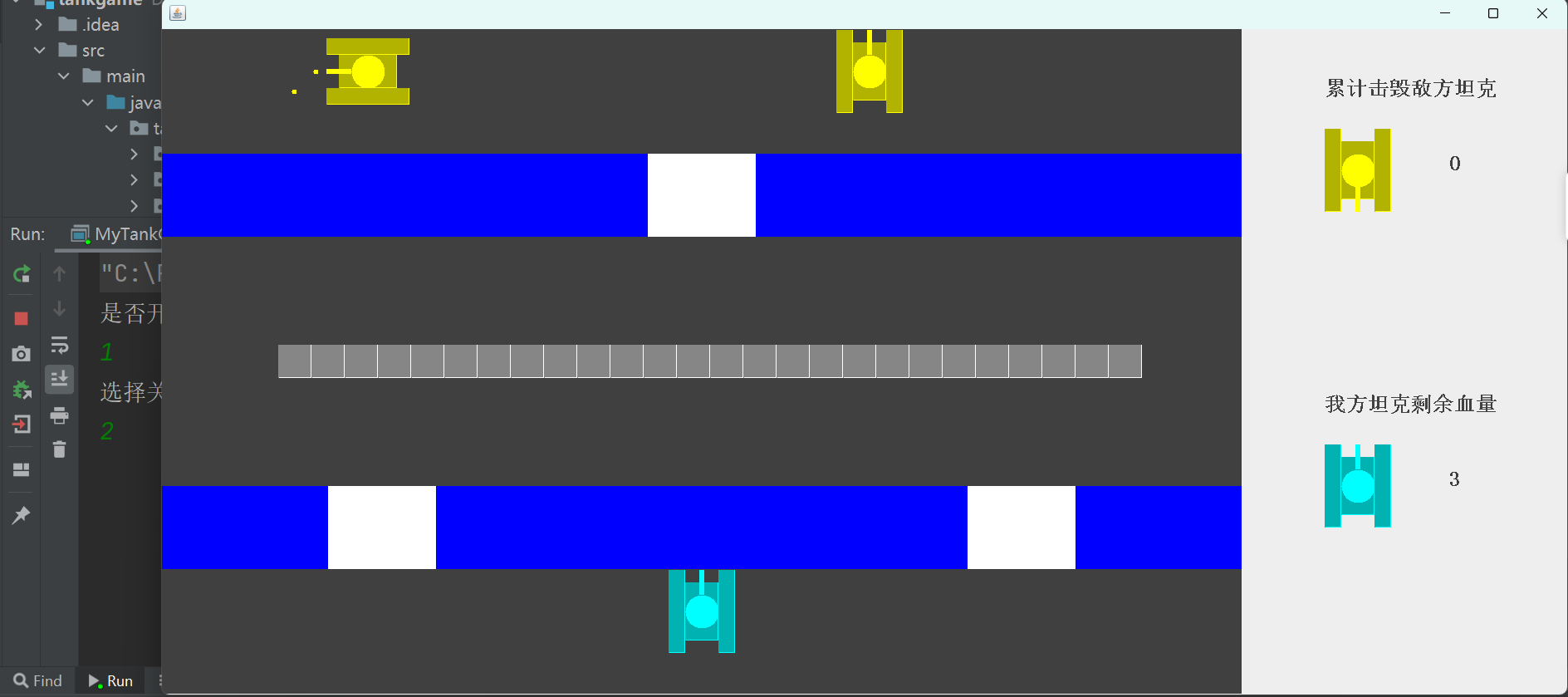
当前版本为0.7，基本功能都已经完成，包括三个关卡、读取玩家输入为坦克进行行动、坦克运动逻辑、子弹运动逻辑、物体碰撞逻辑、结束游戏时的数据保存、游戏启动时的可选择的数据读取。后续可进行的更新有坦克发射子弹总数的控制；自定义关卡；场景物体大小的自定义；存储和读取的数据用数据库进行操作。

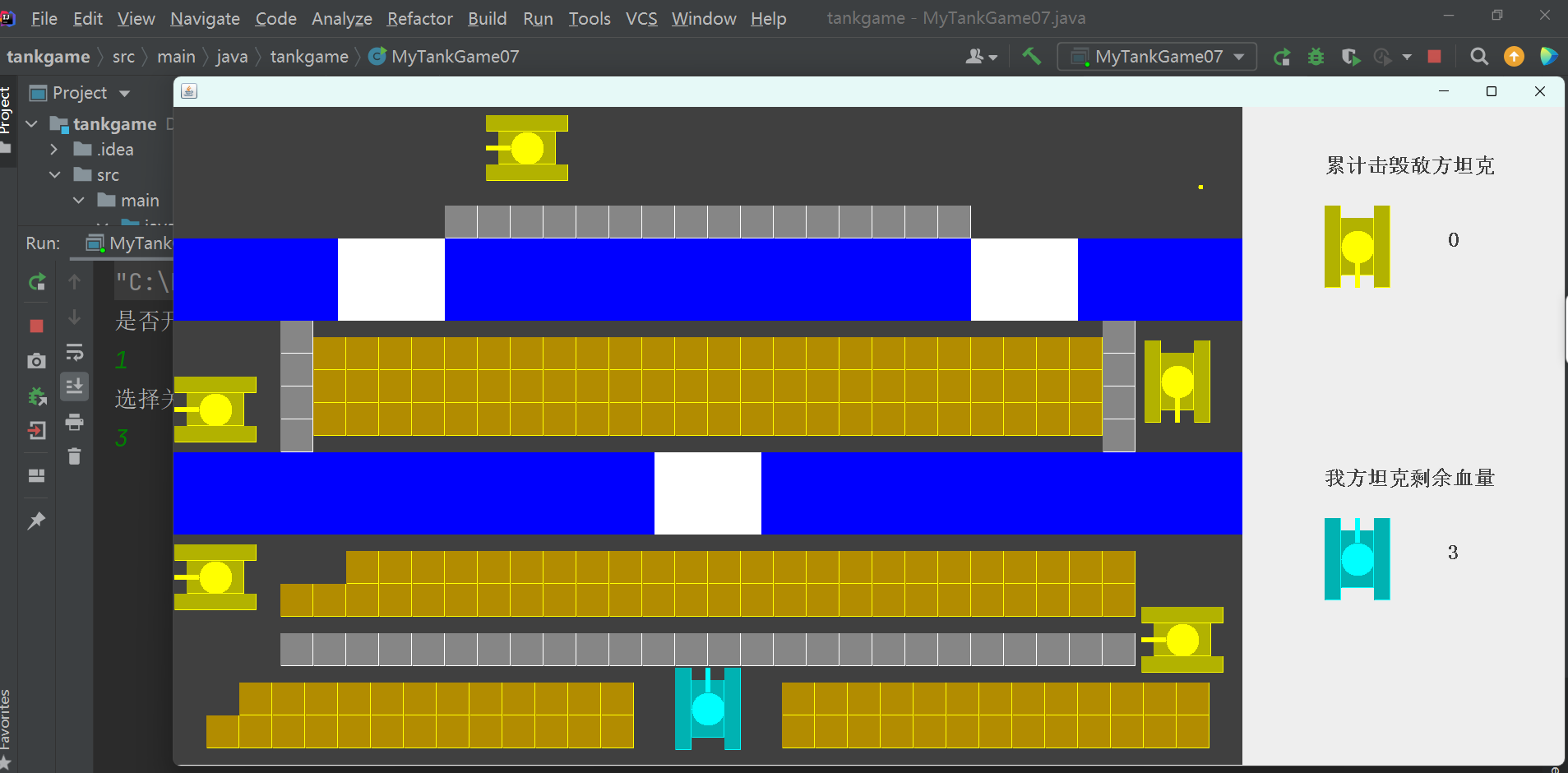
关卡1：



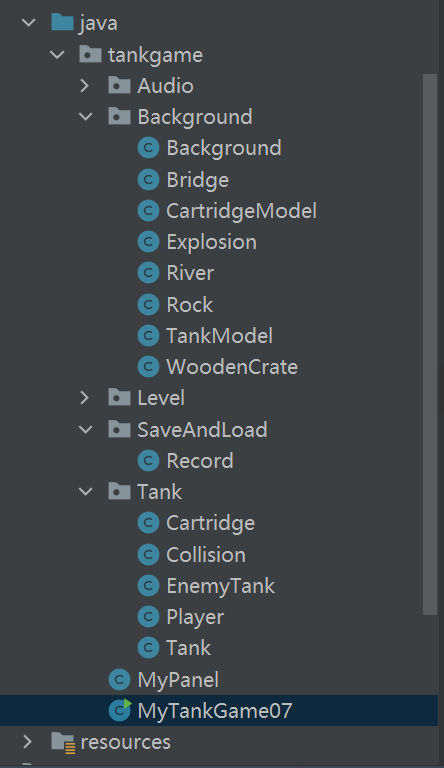
关卡2：



关卡3：



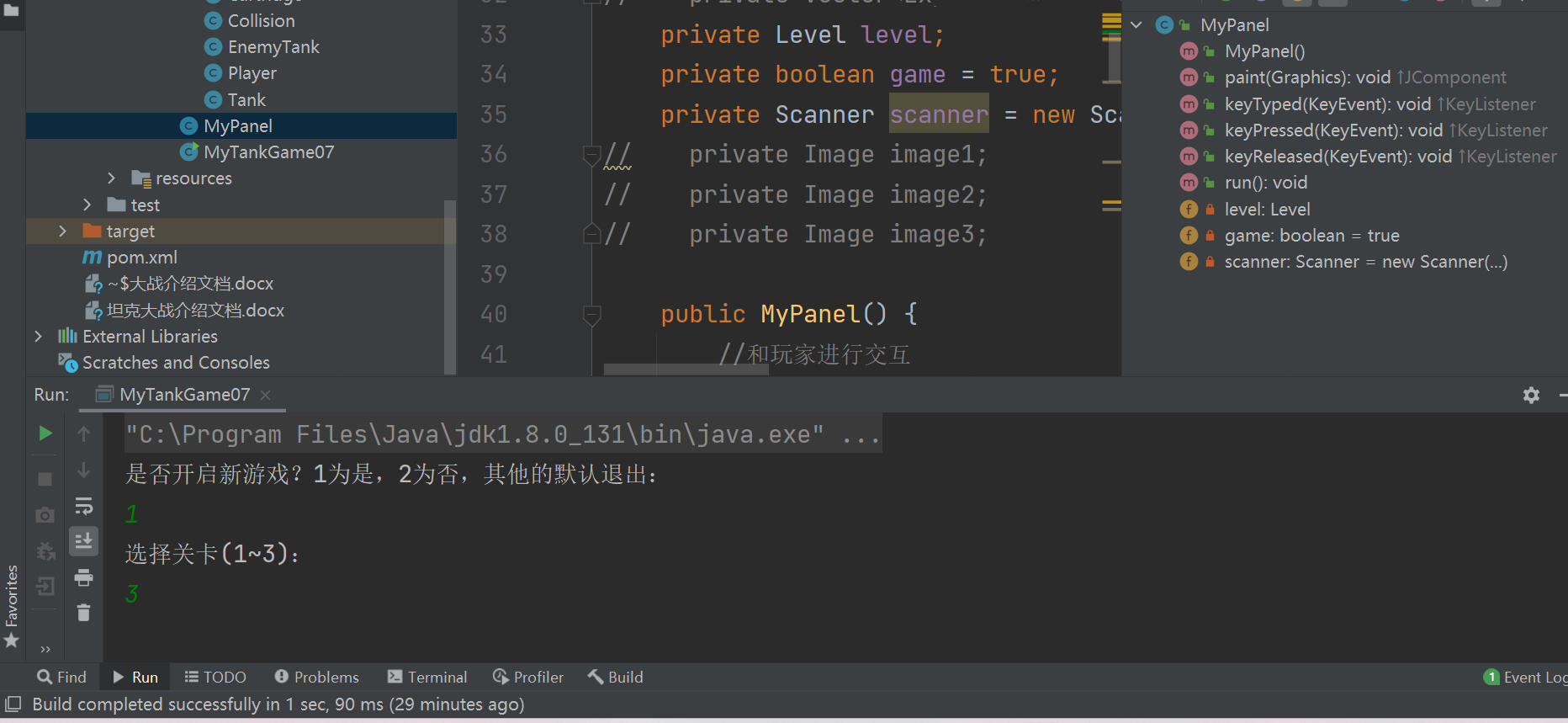
下面进行项目具体功能实现和相关代码解析：



本项目最关键的为Background、Level、SaveAndLoad、Tank四个包下的内容，我会在后面介绍。

首先MyTankGame07.java作为启动类，有着设置窗口大小，启动画布线程，监听键盘输入，在窗口退出时调用Record.java的方法进行数据保存的功能。

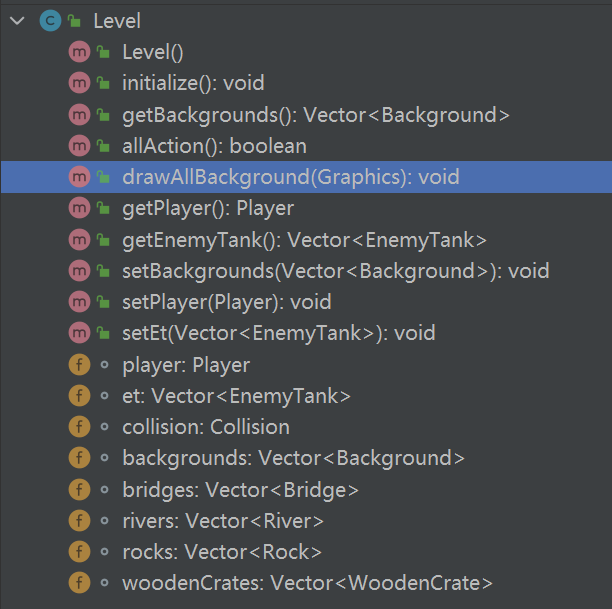
MyPanel.java在早期的版本里，有着初始化关卡，设置各行为逻辑判断，绘图，修改玩家坦克行为等不同的功能，因此在本版本，我将其功能进行了拆解，将其原本没有在使用的功能变成注解进行保留。当前其作为与用户进行最开始的交互，包括新游戏、关卡选择从而进行对应内容的初始化，还有无限重绘、读取键盘输入的内容给玩家坦克设置下一步行动的功能：



Audio包下的AePlayWave.java用于在游戏启动时进行背景音乐的播放，在初始化消息后该线程就会被启动，在音乐播放完成后自动结束，不会在后续重绘中再启动。

Background包的内容都是定义各种物品绘画的大小，并提供绘画方法，其中Background.java为所有该包下其他类的父类，其有一个主要的静态方法drawBackground(Graphics g)用于画初始底色和击杀敌人坦克数和我方坦克生命值，还提供了画桥、河、石头、木箱的空方法给其对应子类去具体实现。TankModel.java和CartridgeModel.java都是提供一个静态方法drawTank(Tank tank, Graphics g)和drawCartridgeModel(Cartridge cartridge, Graphics g)去直接绘制。Explosion.java的绘图使用的是图片，所以设计了一个生命值，每次根据生命值绘图，绘制一次生命值减少一点，从而体现出坦克爆炸的一个过程性。

Level包的内容为各个关卡的初始化，其中Level为所有其他类的父类，提供了一个initialize()的方法给子类重写方法，并在其内部进行场景物品信息的初始。其还有两个关键方法，allAction()方法为所有物体的行动进行合理性推断，drawAllBackground(Graphics g)画出所有的内容，通常使用这个方法后就完成了一次重绘循环。子类对应着各自关卡的初始化，包括地图、玩家和敌人坦克位置。



SaveAndLoad包下只有一个Record.java类，其所有方法和属性都是静态的，实时记录玩家坦克血量和击毁敌方坦克，还有一个Live对象记录给歌曲信息。saveData()方法，用来保存数据，通过读取Live对象内容，对不同的数据进行分开的保存。loadData()方法，在玩家想要继续上局退出时的游戏时被调用，方法会对存储在不同地方的文件数据分开读取，并将其放入Live对象中，在这之前会先调用本类的fileSecurity()对读取位置进行安全性判断。

Tank包下放的内容都是和坦克相关的内容，Tank.java是EnemyTank.java敌人坦克和Player.java玩家坦克的父类，其本身有坐标、速度、当前位置、下一步行动、生命情况、子弹集合的属性，有进行下一步行动的nextAction()方法，与该方法对应的方法有fire()开火、movement()移动、rotation()转向。EnemyTank.java还单独实现Runnable接口，在关卡初始化时启动它的线程，其重写的run()方法内会使用Math.random()获取一个随机数，再对随机数范围判断自动对其下一步行动进行设置。Cartridge.java也实现了Runnable接口，在被坦克发射的时候启动它的线程，其重写的run()方法内会每隔一段时间重复调用自身的bulletTravel()进行移动。Collision.java提供了一个公共certainCollision(Level level)方法判断碰撞结果，只用传入一个关卡对象，其内部自动完成所有坦克和子弹各自碰撞情况和与场景的碰撞情况，具体逻辑可看源代码。

