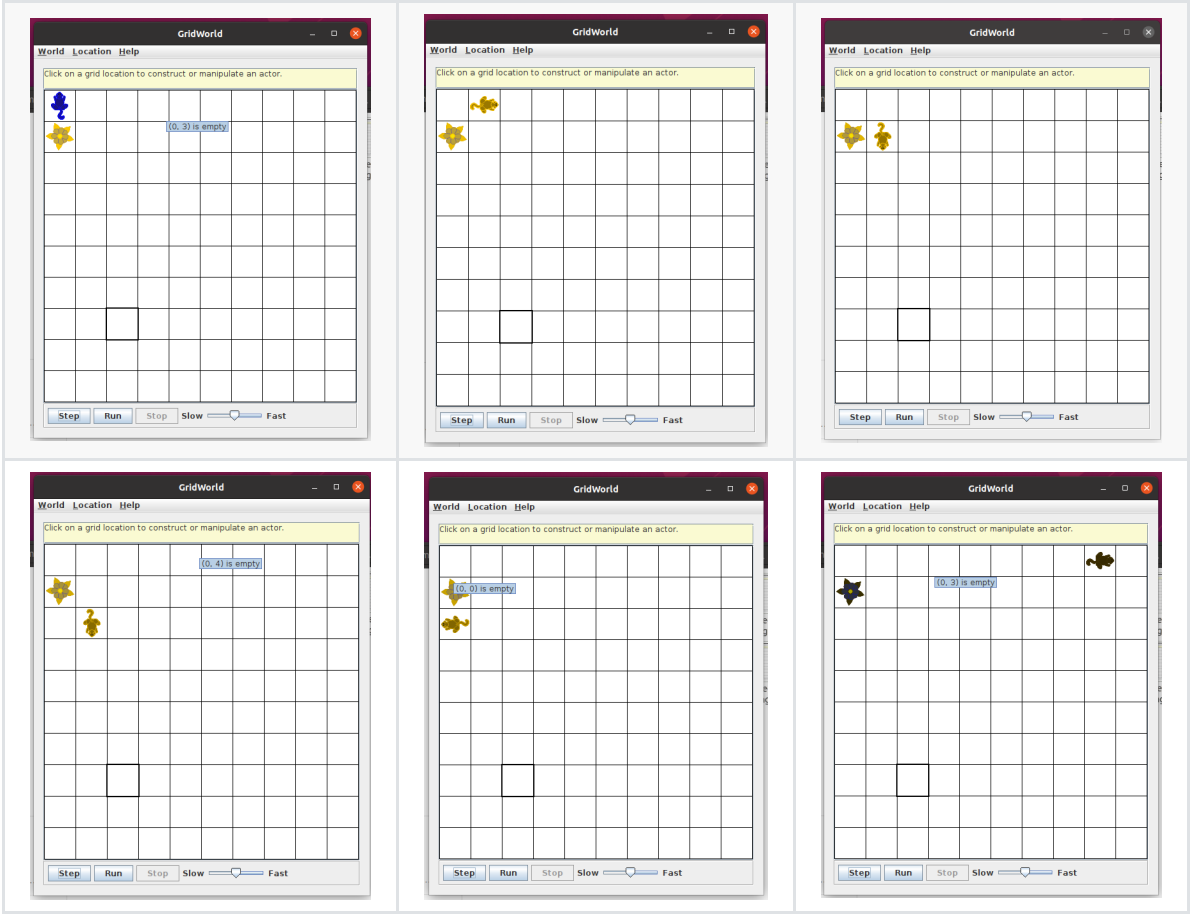
Stage 2 - Part 4 Test Report

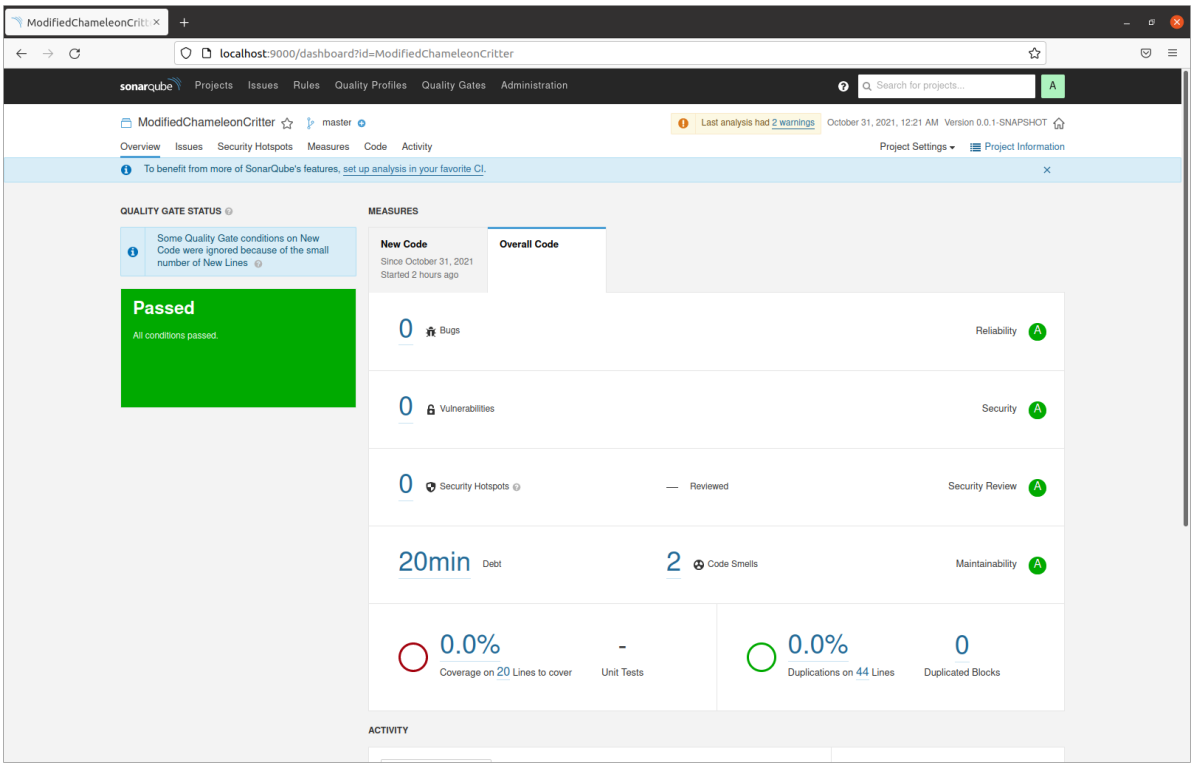
19335286 郑有为

ModifiedChameleonCritter

- **类的说明**：ModifiedChameleonCritter类修改ChameleonCritter中的`processActors`方法，如果要处理的Actor列表为空，则ChameleonCritter的颜色将变暗。
- **实现说明**：重写了`processActors`方法，将Critter颜色变暗的方法参考Flower类的编写，使用一个`DARKENING_FACTOR`变暗因子，每次将颜色调暗即是减小RGB三个通道的值（即乘以（1 - `DARKENING_FACTOR`））。
- **运行结果**：以下是运行ModifiedChameleonCritterRunner的结果
  - 图1：首先创建了一个黄色的花和蓝色的变色龙。
  - 图2：变色龙右移，并改变它的颜色为花的颜色（黄色）。
  - 图3 - 5：变色龙下、右移，并改变它的颜色为花的颜色，没有变暗。
  - 图6：变色龙走了若干步后，周围没有Actor，因而它的颜色慢慢变暗。

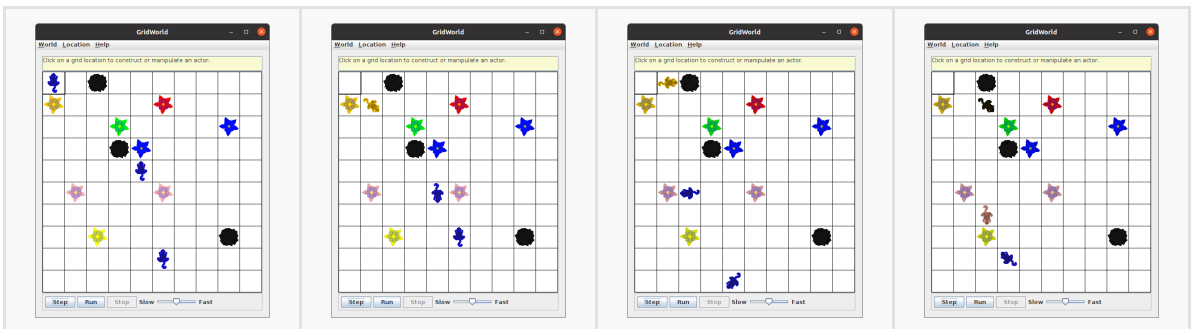


- **Sonar** 代码检查结果：Passed

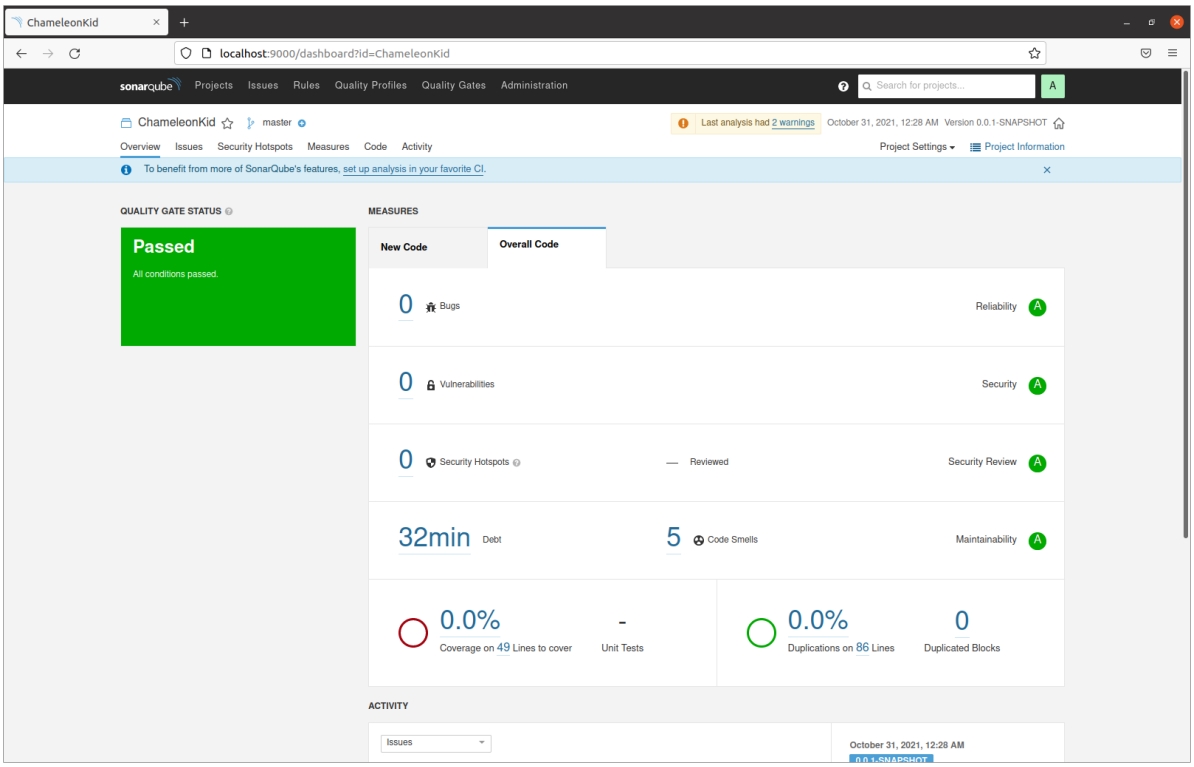


ChameleonKid

- **类的说明**：ChameleonKid类，继承与我们上一个改进版的ChameleonCritter类。小变色龙会把自己的颜色变成紧跟在前面或后面的一个 Actor 的颜色。如果这两个地方都没有Actor ，那么小变色龙的颜色就会变暗。
- **实现说明**：
  - 重写了`getActors`方法，让每次小变色龙只取正前方和正后方位置上的 Actor，它的实现类似于 CrabCritter 类的`getActors`方法，给定一个方向数组然后遍历。
  - 同时重写`getLocationsInDirections`，目的是为了让`getActors`获得指定方向的位置，实现上和 CrabCritter 类的`getLocationsInDirections`是基本一致的。
- **运行结果**：以下是运行ChameleonKidRunner的结果
  - 图1：图片中创建了三个小变色龙，分别位于(0,0), (4,4), (8,5)，其余的是不同颜色的花和石头作为障碍物。
  - 图2：第一次移动，左上小变色龙变成了旁边花的黄色（因为花在它的正后方）；中间的变色龙变为蓝色，他正前方是蓝花（虽然它本身是蓝色的）；下面一个因前后无 Actor 而颜色变暗。
  - 图3：第二次移动，三只小变色龙因前后无 Actor 而变暗；
  - 图4：第三次移动，左上变色龙变成了与正前方黑色石头一样的颜色，中间小变色龙变成了正前方花的分红色，左下变色龙一人因前后无 Actor 颜色变暗。

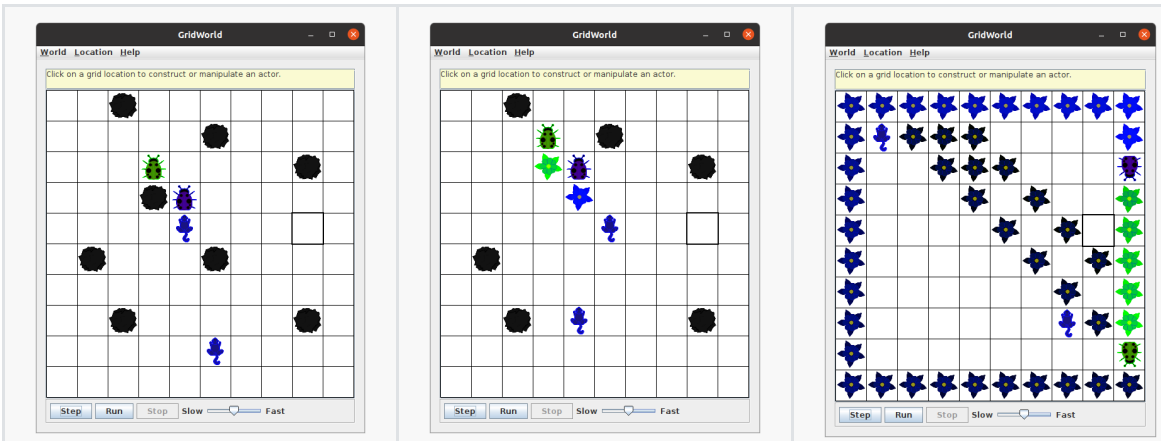


- **Sonar** 代码检查结果：Passed

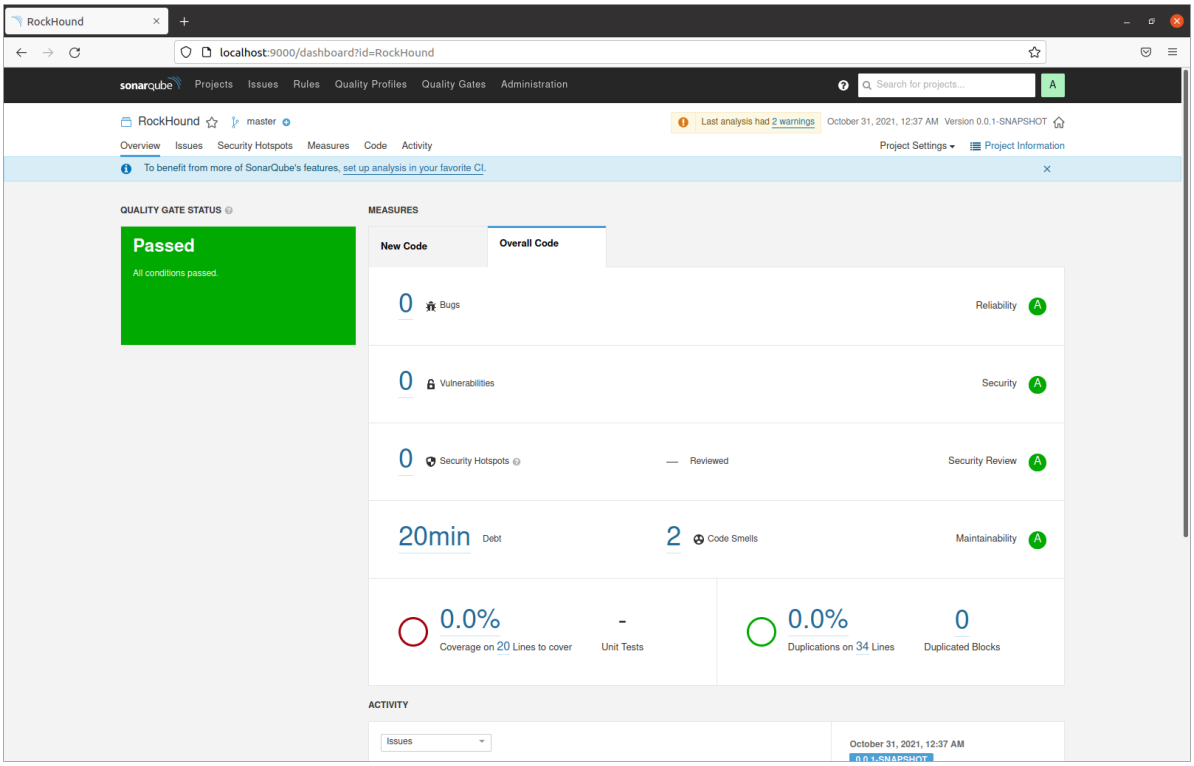


RockHound

- **类的说明**：RockHound类，继承自 Critter，以与 Critter 相同的方式处理 Actor。它从网格中移除列表中的所有岩石，移动方式为默认 Critter 移动方式。
- **实现说明**：重写了`processActors`方法，只移除相邻位置的石头，而不移除其他包括花在内的对象。
- **运行结果**：运行 RockHoundRunner
  - 图1：创建了两个 RockHound 、两个 Bug 和若干个石头
  - 图2：可以看到上边的一个 RockHound 吃掉了 石头(5, 5)
  - 图3：经过若干步，RockHound们吃掉了所有的石头，而花和虫都没有被吃掉。

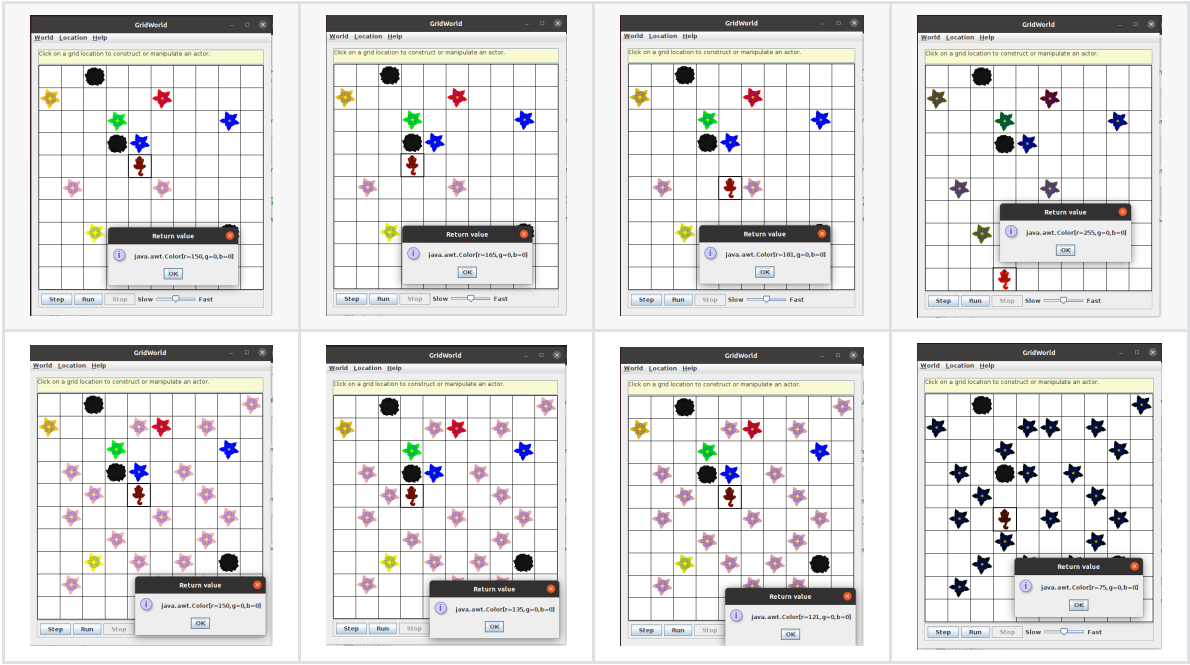


- **Sonar** 代码检查结果：Passed

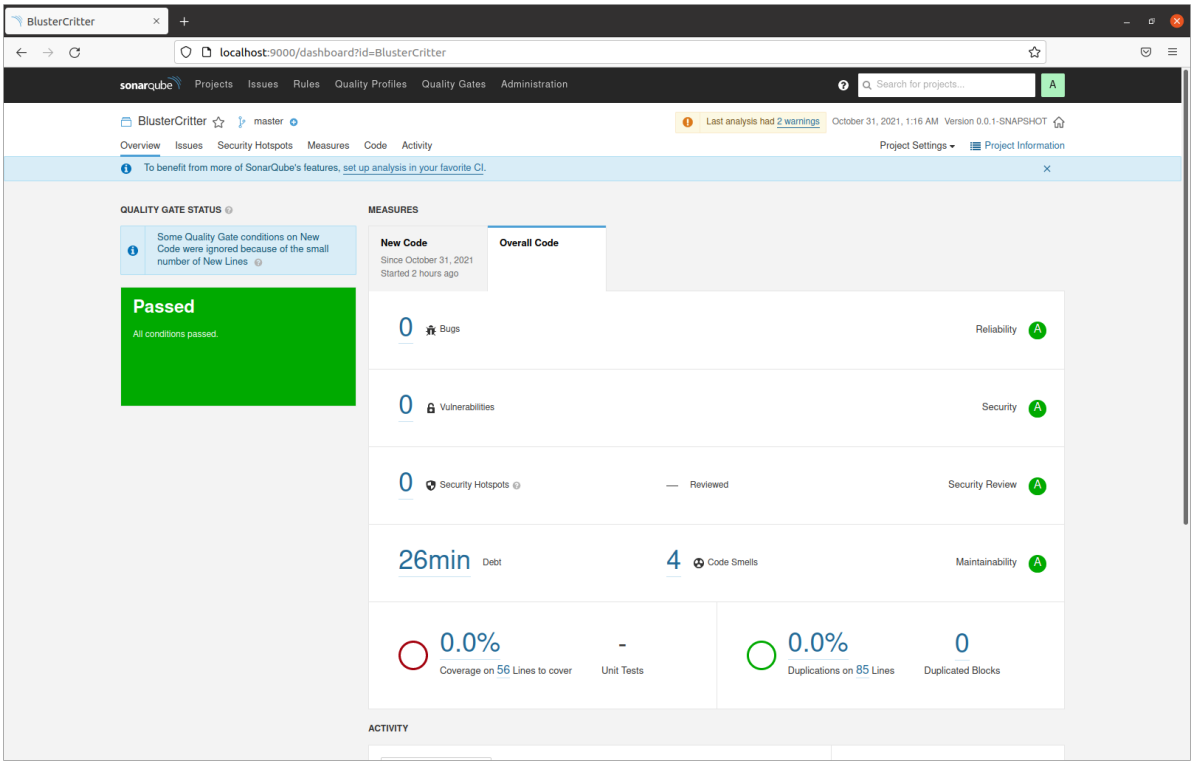


BlusterCritter

- **类的说明**：创建一个扩展了Critter的类BlusterCritter。BlusterCritter查看当前位置**两步内的所有邻居**。（对于不在边缘附近的BlusterCritter，这包括24个位置）。它计算这些地方的生物数量。如果有少于 c 的生物，BlusterCritter的颜色会变亮（颜色值增加）。如果有 c 或更多的生物，BlusterCritter的颜色变暗（颜色值降低）。这里，c 是表示 Critter 勇气的值。它应该在构造函数中设置。
- **实现说明**：
  - 提供带参数 c 的构造函数，缺省是为10。
  - 重写了`getActors`方法，因为默认的发法只会取相邻一步位置的邻居，因此我们徐奥根据当前位置的行列来对位置进行遍历，来获取两个字范围内出它自己之外的所有 Actor，注意要处理越界，即调用 Grid 的`isValid`来判断位置是否合法。
  - 重写`processActors`方法，我们可以通过调用`getActor`来获取周围 Actor 的列表，然后调用列表的`size`方法来获取周围 Actor 的个数，与 c 值比较，来对颜色进行修改，如何修改与 ChameleonCritter 的重写类似，但要注意RGB值每项不能超过255，否则会报错。（我们默认颜色的变化因子为0.1）
- **运行结果**：运行 BlusterCritterRunner（图1到图4），图5到图8是用鼠标重新布局后做测试，实验中设置 c 为 5
  - 在图1到图4中， BlusterCritter 的周围的 Actor 一直没有超过5个，因此它的颜色不断变亮，我们点击 getColor 得到颜色变化：
    - 前三步BlusterCritter的颜色：(150,0,0)，(165,0,0)，(181,0,0)，执行若干步后的颜色：(255,0,0)此时最亮。
  - 在图5到图8中， BlusterCritter 的周围的 Actor 大部分时间都大于等于5个，因此它的颜色不断变暗，我们点击 getColor 得到颜色变化：
    - 前三步BlusterCritter的颜色：(150,0,0)，(135,0,0)，(121,0,0)，执行若干步后的颜色：(75,0,0)。



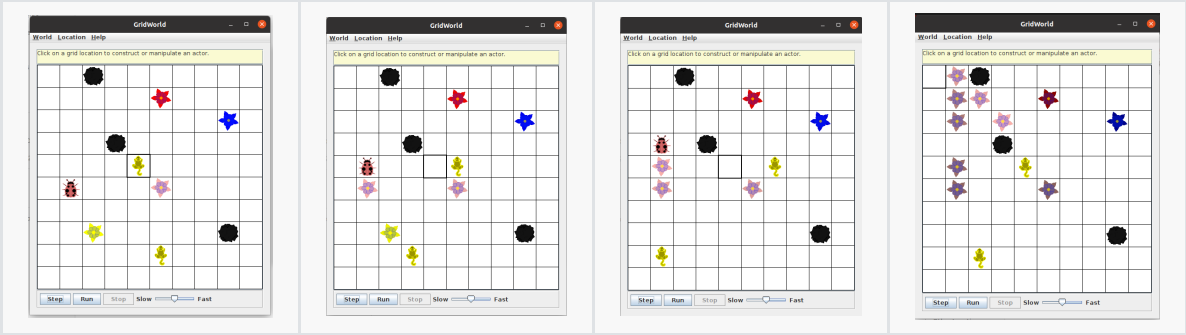
- **Sonar** 代码检查结果：Passed



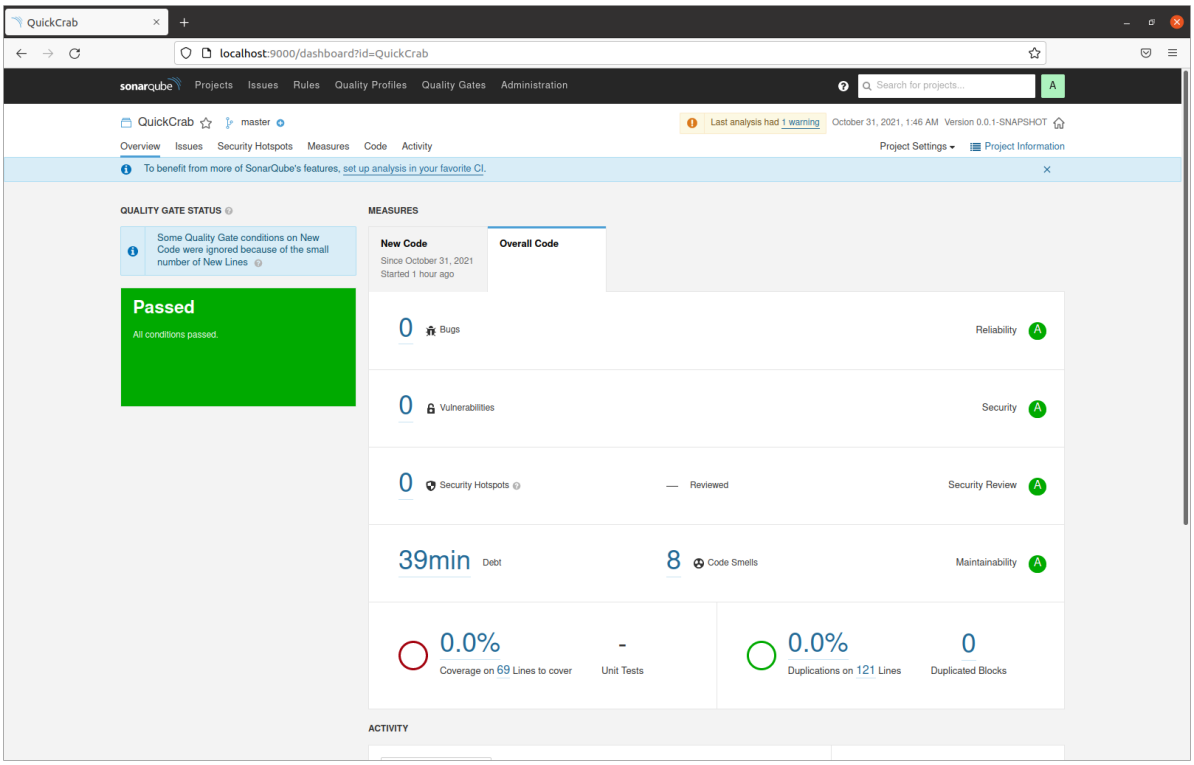
QuickCrab

- **类的说明**：继承于CrabCritter类的QuickCrab。它处理 Actor 的方式与 CrabCritter 相同。移动方式比较特殊：如果这个位置和中间的位置都是空的，那么QuickCrab就会移动到这两个随机选择的位置中的一个，这两个位置是它左边或右边的两个空格（就是可以一次沿着一个方向移动两格子）。否则，QuickCrab 就会像 CrabCritter 一样移动。
- **实现说明**：重写了`getMoveLocations`方法，顾名思义，就是修改螃蟹下一步能够移动到的位置，首先旁先只能左移右移，故移动位置最多只有四个，左侧右侧分别用一嵌套的条件判断是否合法即可。
- **运行结果**：运行 QuickCrabRunner

- 图1：首先创建了两个 Crab（它这里没有显示螃蟹图案的原因是我忘记把 Crab.gif 拖当前文件夹里了），和若干个 Bug 和 Flower 作为螃蟹的食物。
- 图2：上面一只螃蟹右移了一格，下面那只左移了两格，因为这些位置为空（(5, 4), (9, 4), (9, 3)），所以都是合法的。
- 图3：上面一只螃蟹再一次右移了一格，下面那只再一次左移了两格，并吃掉了黄花 (7,2)。
- 图4：移动了若干次后，上面那只螃蟹正好吃掉了小虫。



- **Sonar** 代码检查结果：Passed

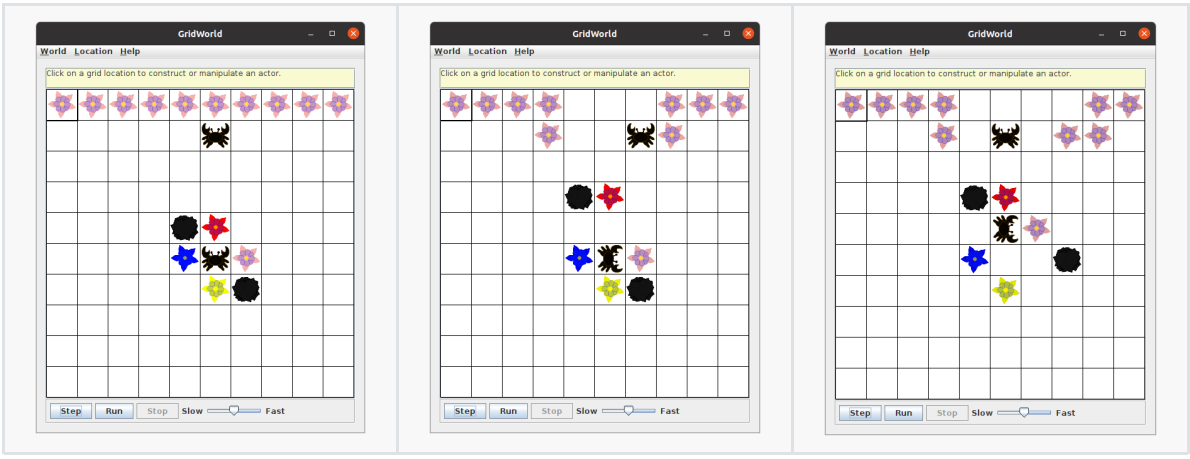


KingCrab

- **类的说明**：扩展了 CrabCritter 的类 KingCrab，它以与 CrabCritter 相同的方式处理 Actor。KingCrab 会使它处理的每个参与者将一个位置移离 KingCrab。如果 Actor 不能移动，KingCrab 就会将其从网格中移除。当 KingCrab 完成对演员的处理后，它会像 CrabCritter 一样移动。
- **实现说明**：
  - 重写`processActors`方法，获取相邻位置的所有 Actor 并逐个驱使它们，当它们无路可走时就移除它们。如何为 Actor 选择离开的路线，使用`isSaveLocaion`方法来判断
  - 提供`isSaveLocation`方法：我们通过计算 Actor 每一个空的相邻位置到 KingCrab 的距离，如大于等于2即可移动到该位置，具体的计算方法就是通过勾股定理。
- **运行结果**：运行 KingCrabRunner
  - 图1：在网格上边部分创建了一排花和一个KingCrab，下边船舰一个被石头和花包围的螃蟹

- 图2：首先看上面一排花，中间一朵花因没有位置被螃蟹吃掉，左右两朵被挤到两侧，螃蟹处理完后向右移动了一格；再看下面一螃蟹，螃蟹把正前方的左前方的石头和花推了上去，然后因为左右没有位置而改变移动方向。
- 图3：首先看上面一只螃蟹，螃蟹右前方的花朵因为旁边没有距离螃蟹位置大于1的位置，故被吃掉，右边的花因不在螃蟹的 getActor 范围而不受影响；再看下面一螃蟹，螃蟹把正前方的左前方的石头和花推了出去，然后因为向上移动了一格子。

> 注：这里的花移动位置有问题，代码已改正为距离判断大于等于2（1.9）。



- **Sonar** 代码检查结果：Passed