

动 20 步后调用过程 Row 即可。因此，重复调用过程 Row 就能绘制任意行数，如图 4-22 所示的过程 Checkers。

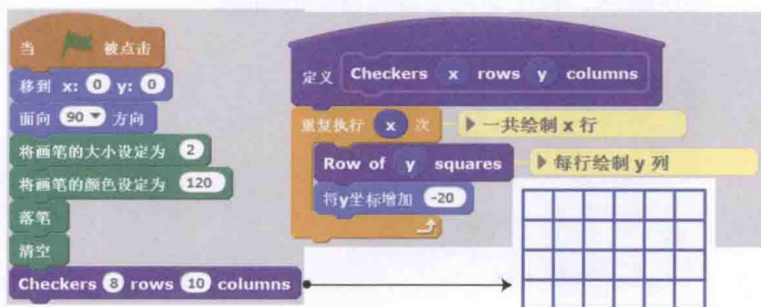


图 4-22：过程 Checkers 及其绘制结果

过程 Checkers 有两个参数：行数和列数。当某行绘制完毕后，角色向下移动 20 步，为绘制下一行做好准备。

本节的例子说明了程序划分为更小、更易管理的片段的重要性。对复杂的程序来说，我们可以复用已经编写好并通过测试的过程，甚至在其基础上继续构建过程，而不需要过分担心底层是如何实现的。这样便能将自己的精力放在最需要解决的问题上。

试一试 4-2

如果把初始化脚本中的面向 90° 方向改为面向 45° 方向^{译注 3}，运行后的结果和你预期的结果一致吗？如何修改才能绘制斜着的棋盘呢？尝试完成此处修改并运行脚本，测试你的答案。

分析问题的思维方式

相信你已经察觉到将程序划分为小部分并逐一解决是一种非常重要的思维方式。现在我们就来讨论一下如何分析问题。不同问题之间虽然都存在差异，也没有千篇一律的解决方法，但这正是解决难题的魅力所在！

译注3：原文有误，应为45°。