

2024-2025 学年小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册教学设计合集

目录

一、走进图形化编程的世界

- 1.1 第 1 课 Scraino 初体验
- 1.2 第 2 课 奔跑吧，小猫
- 1.3 第 3 课 猫捉老鼠
- 1.4 第 4 课 幸运大转盘
- 1.5 第 5 课 乌鸦喝水
- 1.6 第 6 课 成语大赛
- 1.7 第 7 课 一起去寻宝
- 1.8 第 8 课 倒计时器
- 1.9 第 9 课 制作电子点菜单
- 1.10 第 10 课 打地鼠
- 1.11 第 11 课 神奇的画笔
- 1.12 第 12 课 小熊的花园
- 1.13 第 13 课 声控的秘密
- 1.14 第 14 课 有趣的体感游戏

走进图形化编程的世界第 1 课 Scraino 初体验

课题：		
科目：	班级：	课时：计划 <u>3</u> 课时
教师：	单位：	

2024-2025 学年小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册教学设计合集

目录

一、走进图形化编程的世界

- 1.1 第 1 课 Scraino 初体验
- 1.2 第 2 课 奔跑吧，小猫
- 1.3 第 3 课 猫捉老鼠
- 1.4 第 4 课 幸运大转盘
- 1.5 第 5 课 乌鸦喝水
- 1.6 第 6 课 成语大赛
- 1.7 第 7 课 一起去寻宝
- 1.8 第 8 课 倒计时器
- 1.9 第 9 课 制作电子点菜单
- 1.10 第 10 课 打地鼠
- 1.11 第 11 课 神奇的画笔
- 1.12 第 12 课 小熊的花园
- 1.13 第 13 课 声控的秘密
- 1.14 第 14 课 有趣的体感游戏

走进图形化编程的世界第 1 课 Scraino 初体验

课题：		
科目：	班级：	课时：计划 <u>3</u> 课时
教师：	单位：	

2024-2025 学年小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册教学设计合集

目录

一、走进图形化编程的世界

- 1.1 第1课 Scraino 初体验
- 1.2 第2课 奔跑吧，小猫
- 1.3 第3课 猫捉老鼠
- 1.4 第4课 幸运大转盘
- 1.5 第5课 乌鸦喝水
- 1.6 第6课 成语大赛
- 1.7 第7课 一起去寻宝
- 1.8 第8课 倒计时器
- 1.9 第9课 制作电子点菜单
- 1.10 第10课 打地鼠
- 1.11 第11课 神奇的画笔
- 1.12 第12课 小熊的花园
- 1.13 第13课 声控的秘密
- 1.14 第14课 有趣的体感游戏

走进图形化编程的世界第1课 Scraino 初体验

课题：		
科目：	班级：	课时：计划 <u>3</u> 课时
教师：	单位：	

一、教材分析
<p>本节课选自小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册，课程内容为“走进图形化编程的世界第 1 课 Scraino 初体验”。本节课旨在让学生初步了解 Scratch 编程软件，培养学生的编程兴趣和动手能力。教材内容与实际操作紧密结合，通过生动有趣的案例，让学生在轻松愉快的氛围中学习编程知识。</p>
二、核心素养目标
<ol style="list-style-type: none"> 1. 信息意识：培养学生对信息技术和编程的敏感性，认识到编程在生活中的应用价值。 2. 计算思维：通过编程实践，发展学生的逻辑思维和问题解决能力。 3. 数字化学习与创新：鼓励学生运用 Scratch 进行创新创作，提高数字化学习素养。 4. 信息责任：教育学生尊重知识产权，合理使用网络资源，培养良好的网络道德。
三、教学难点与重点
<ol style="list-style-type: none"> 1. 教学重点， <ol style="list-style-type: none"> ① 熟悉 Scratch 编程软件的基本界面和操作。 ② 掌握 Scratch 中基本图形和角色的使用方法，包括移动、旋转、绘制等操作。 ③ 理解编程逻辑，学会使用积木块编写简单的程序。 2. 教学难点， <ol style="list-style-type: none"> ① 理解编程逻辑的抽象概念，将现实问题转化为编程问题。 ② 创意编程，根据学生的兴趣和需求，设计富有创意的编程项目。 ③ 解决编程过程中遇到的问题，培养学生的调试能力和问题解决能力。
四、教学方法与手段
<p>教学方法：</p> <ol style="list-style-type: none"> 1. 讲授法：通过教师的讲解，介绍 Scratch 编程软件的基本功能和操作步骤，帮助学生建立初步的认知。 2. 实验法：引导学生动手实践，通过实际操作 Scratch 软件，学习编程知识和技能。 3. 情境教学法：创设实际编程情境，让学生在解决具体问题的过程中学习编程，提高解决问题的能力。 <p>教学手段：</p> <ol style="list-style-type: none"> 1. 多媒体教学：利用 PPT 展示 Scratch 软件界面和编程步骤，直观演示编程过程。 2. 互动教学软件：使用 Scratch 教学软件，让学生在虚拟环境中进行编程练习，提高学习兴趣。 3. 互联网资源：利用网络资源，如在线教程、社区论坛等，为学生提供丰富的学习资源和交流平台。
五、教学过程

1. 导入 (约 5 分钟)

- 激发兴趣：通过展示一些有趣的 Scratch 动画作品，提问学生“你们知道这些动画是如何制作的吗？”来激发学生的好奇心和兴趣。

- 回顾旧知：引导学生回顾之前学习的信息技术基础知识，如计算机的基本操作和互联网的使用。

2. 新课呈现 (约 15 分钟)

- 讲解新知：介绍 Scratch 编程软件的基本界面，包括角色、舞台、积木块等元素，讲解它们的基本功能和用途。

- 举例说明：通过简单的 Scratch 动画示例，展示如何通过积木块编写程序来控制角色移动、旋转和绘制图形。

- 互动探究：让学生分组讨论，提出一些简单的编程问题，如“如何让角色跳起来？”或“如何让角色跟随鼠标移动？”，鼓励学生尝试解决。

3. 实践操作 (约 25 分钟)

- 学生活动：每个学生都获得一台装有 Scratch 软件的计算机，开始尝试自己创作简单的动画。

- 教师指导：教师在教室中巡回指导，帮助学生在遇到困难时提供帮助，同时观察学生的学习进度。

4. 巩固练习 (约 15 分钟)

- 学生活动：学生根据教师提供的模板或自由发挥，创作一个简单的 Scratch 动画，如绘制一个会动的笑脸或绘制一个简单的游戏。

- 教师指导：教师鼓励学生展示自己的作品，并给予反馈和评价，同时指出作品中可以改进的地方。

5. 总结与反思 (约 5 分钟)

- 教师总结：回顾本节课的学习内容，强调 Scratch 编程的趣味性和实用性。

- 学生反思：让学生谈谈自己在学习过程中的收获和感受，以及对于编程学习的认识。

6. 布置作业 (约 2 分钟)

- 学生作业：布置一个简单的 Scratch 编程作业，要求学生在课后创作一个具有特定功能的动画，并分享给其他同学。

7. 课堂小结 (约 2 分钟)

- 教师小结：对整个教学过程进行简要回顾，强调编程学习的重要性，并鼓励学生在家里继续学习和探索。

教学过程中，教师应灵活调整时间分配，根据学生的实际学习情况适当增减时间。

六、拓展与延伸

1. 提供与本节课内容相关的拓展阅读材料

- 《Scratch 编程入门教程》：适合初学者的入门书籍，详细介绍了 Scratch 的基本操作和编程技巧。

- 《Scratch 创意编程实例集》：收录了多个创意编程实例，帮助学生拓展思维，提高编程水平。

-

《儿童编程启蒙》：针对儿童读者的编程启蒙书籍，以故事和游戏的方式介绍编程知识，激发孩子对编程的兴趣。

2. 鼓励学生进行课后自主学习和探究

- 学生可以尝试使用 Scratch 创作一些简单的动画，如动画故事、游戏等，以巩固所学知识。
- 引导学生探索 Scratch 的高级功能，如使用变量、列表等，提高编程技能。
- 鼓励学生参加线上编程社区，与其他编程爱好者交流心得，共同进步。
- 指导学生通过编程解决实际问题，如制作家庭作业管理器、天气预报工具等，提高编程实用性。
- 鼓励学生参加 Scratch 相关的比赛和活动，展示自己的编程成果，激发学习动力。
- 引导学生关注编程在各个领域的应用，如教育、娱乐、艺术等，拓宽视野，激发创新思维。

七、板书设计

① Scraino 编程软件介绍

- 软件名称：Scratch
- 操作界面：舞台、角色、积木块
- 基本功能：图形化编程、角色控制、事件响应

② Scratch 编程基础

- 积木块类型：运动、外观、控制、声音、事件、数据、操作
- 基本操作：拖放积木块、连接程序、测试运行

③ 编程逻辑与技巧

- 条件判断：if-else 结构
- 循环结构：重复执行程序
- 变量与列表：存储和操作数据

④ 编程实例分析

- 简单动画：角色移动、旋转、绘制图形
- 游戏设计：控制角色、设置规则、判断胜负

⑤ 编程实践要点

- 创意构思：确定编程主题和目标
- 程序调试：解决程序中的错误
- 作品展示：分享自己的编程成果

八、课堂小结，当堂检测

课堂小结：

在本节课中，我们学习了 Scratch 编程软件的基本操作和编程知识。首先，我们了解了 Scratch 的界面布局，包括舞台、角色和积木块等元素。通过实际操作，学生们掌握了如何使用 Scratch 中的运动、外观、控制、声音等积木块来控制角色的动作和外观变化。

接着，我们学习了编程逻辑的基础知识，包括条件判断和循环结构。通过具体的例子，学生们理解了如何使用 if-else 结构和重复执行程序来编写更加复杂和有趣的程序。

在编程实践环节，学生们分组合作，尝试创作了自己的 Scratch 动画和游戏。在这个过程中，学生们不仅巩固了所学知识，还锻炼了自己的创意思维和问题解决能力。

当堂检测：

为了检测学生对本节课内容的掌握程度，我们将进行以下检测：

1. 基础知识问答：

- 请简要介绍 Scratch 编程软件的基本功能和特点。
- 列举 Scratch 中常用的积木块类型。
- 解释条件判断和循环结构在编程中的作用。

2. 编程实践操作：

- 请使用 Scratch 编程软件，编写一个简单的动画，让角色在舞台上移动和旋转。
- 请使用 Scratch 编程软件，设计一个小游戏，让角色根据玩家的操作进行跳跃。

3. 创意展示：

- 请展示你在本节课中创作的 Scratch 作品，并简要介绍你的创意思路和实现方法。

检测过程中，教师将巡视教室，观察学生的操作情况，并给予必要的指导和帮助。

检测结束后，教师将针对学生的表现进行评价，并对本节课的知识点进行总结。

通过本节课的学习，学生们不仅掌握了 Scratch 编程软件的基本操作和编程技巧，还提高了自己的逻辑思维和问题解决能力。希望学生们能够将所学知识应用到实际生活中，创作出更多有趣的编程作品。同时，教师也将继续关注学生的学习进度，提供必要的支持和指导。

走进图形化编程的世界第 2 课 奔跑吧，小猫

课题：		
科目：	班级：	课时：计划 3 课时
教师：	单位：	
一、设计思路		

本课以“奔跑吧，小猫”为主题，结合小学信息技术（信息科技）鲁教版五年级下册教材内容，旨在引导学生通过图形化编程的方式，学习控制角色动作，提高学生的编程兴趣和动手实践能力。课程设计注重理论与实践相结合，通过实际操作，让学生在解决问题的过程中，掌握编程的基本概念和操作方法。教学活动包括角色创建、动作编程、逻辑判断等环节，旨在培养学生的创新思维和问题解决能力。

二、核心素养目标

1. 信息意识：培养学生对信息技术的兴趣和好奇心，认识到编程在生活中的应用价值，提高学生获取、处理和利用信息的能力。
2. 计算思维：通过编程实践，锻炼学生的逻辑思维和问题解决能力，学会将实际问题转化为编程问题，并运用编程逻辑解决。
3. 数字化学习与创新：培养学生利用图形化编程工具进行创新设计的能力，激发学生的创造力和想象力。
4. 信息伦理：引导学生遵守网络规则，尊重知识产权，树立正确的网络安全意识。
5. 信息安全：让学生了解编程过程中可能遇到的安全问题，掌握基本的安全防护措施。

三、重点难点及解决办法

重点：

1. 角色动作编程：重点在于让学生掌握如何通过编程指令控制角色的动作，如前进、后退、转弯等。
2. 逻辑判断：重点在于让学生理解并应用条件语句，实现角色行为的分支。

难点：

1. 编程逻辑理解：部分学生对编程逻辑的理解可能存在困难，难以将逻辑思维转化为编程代码。
2. 问题解决：在编程过程中，学生可能会遇到各种问题，需要具备良好的问题解决能力。

解决办法：

1. 通过实际操作演示，逐步讲解编程逻辑，帮助学生理解。
2. 设计阶梯式任务，从简单到复杂，逐步提高学生的编程难度，让学生在实践中逐步掌握。
3. 鼓励学生互相讨论，分享解决方法，培养学生的团队合作和问题解决能力。

四、教学方法与手段

教学方法：

1. 讲授法：结合实际案例，讲解编程基本概念和指令，帮助学生建立初步的认识。
2. 演示法：通过教师现场演示编程过程，让学生直观理解编程步骤。
3. 实践法：鼓励学生动手操作，亲自编写程序，体验编程的乐趣和成就感。

教学手段：

1. 多媒体课件：利用 PPT 展示编程界面和指令，提高教学内容的可视性。
2. 编程软件：运用图形化编程软件，让学生在软件环境中进行编程实践。
3. 教学视频：播放编程教程视频，为学生提供额外的学习资源，巩固课堂所学知识。

五、教学实施过程

1. 导入新课

- 利用故事引入，激发学生对编程的兴趣。
- 展示编程作品，引导学生思考编程在现实生活中的应用。

2. 编程实践

- 学生分组，每组选择一个角色进行编程。
- 教师指导学生如何使用图形化编程工具。
- 学生通过编程，实现角色的基本动作。

3. 逻辑判断

- 教师讲解逻辑判断的概念和作用。
- 学生在编程过程中应用逻辑判断，解决角色行为的问题。

4. 问题解决

- 学生在编程过程中遇到问题，互相讨论，共同解决。
- 教师给予个别指导，帮助学生突破难点。

5. 总结与反思

- 学生展示自己的编程作品，分享学习心得。
- 教师点评，总结课程重点，布置课后作业。

六、知识点梳理

1. 图形化编程基础

- 图形化编程的概念和特点
- 图形化编程软件的基本界面和功能

2. 角色创建与编辑

- 角色创建的基本步骤
- 角色属性的设置与调整

3. 角色动作编程

- 基本动作指令：前进、后退、转弯等
- 动作组合与顺序执行
- 动作循环与暂停

4. 逻辑判断与条件语句

- 条件判断的概念和作用
- 使用条件语句实现角色的智能行为
- 多重条件判断与嵌套逻辑

5. 角色行为控制

- 角色碰撞检测
- 角色得分与生命值管理
- 角色死亡与复活处理

6. 编程项目实践

- 项目规划与设计
- 项目开发与调试
- 项目展示与评价

7. 编程规范与最佳实践

-

代码编写规范

- 代码注释与文档编写
 - 版本控制与代码管理
8. 编程安全与伦理
- 编程过程中的安全风险
 - 编程伦理与法律法规
 - 网络安全与数据保护
9. 编程工具与资源
- 常用图形化编程软件介绍
 - 编程学习网站与社区资源
 - 编程辅助工具与插件
10. 编程思维与创新能力
- 编程思维的概念与培养
 - 创新能力的激发与培养
 - 编程项目中的创新实践

七、板书设计

- ① 编程基础概念
- 图形化编程简介
 - 编程软件界面与功能
- ② 角色创建与编辑
- 角色创建步骤
 - 角色属性设置
- ③ 角色动作编程
- 基本动作指令
 - 动作组合与顺序
- ④ 逻辑判断与条件语句
- 条件判断概念
 - 条件语句应用
- ⑤ 行为控制
- 碰撞检测
 - 得分与生命值管理
- ⑥ 编程实践
- 项目规划与设计
 - 项目开发 with 调试
- ⑦ 编程规范与最佳实践
- 代码编写规范
 - 代码注释与文档
- ⑧ 编程安全与伦理
- 编程安全风险
 - 编程伦理与法律法规
- ⑨ 编程工具与资源



常用编程软件

- 编程学习资源

⑩ 编程思维与创新能力

- 编程思维培养

- 创新能力激发

八、反思改进措施

反思改进措施（一）教学特色创新

1. 项目式学习：在课程中引入实际项目，让学生通过解决具体问题来学习编程，提高学生的实践能力和解决问题的能力。

2. 合作学习：鼓励学生分组合作，共同完成编程任务，培养团队协作精神和沟通能力。

反思改进措施（二）存在主要问题

1. 学生基础差异：部分学生对编程基础知识的掌握程度不一，教学过程中需要更加关注不同学生的学习需求。

2. 实践机会不足：由于时间限制，学生实际操作的机会有限，需要寻找更多机会让学生动手实践。

3. 评价方式单一：目前主要依靠作业和课堂表现来评价学生的学习成果，可以考虑引入更多元化的评价方式。

反思改进措施（三）改进措施

1. 针对学生基础差异，我会设计分层教学方案，针对不同层次的学生提供相应的学习资源和支持。

2. 为了增加学生的实践机会，我会尝试将部分教学内容与课外实践活动相结合，比如组织编程比赛或项目开发活动。

3. 在评价方面，我将尝试引入形成性评价和过程性评价，不仅关注学生的最终成果，也关注他们在学习过程中的努力和进步。同时，可以引入学生自评、互评等评价方式，让学生更加积极参与评价过程。

• 典型例题讲解

例题 1：

小猫需要从起点跑到终点，路程为 100 个单位。请编写程序控制小猫以每秒 2 个单位的速度前进，计算小猫到达终点所需的时间。

答案：小猫到达终点所需的时间 = 路程 / 速度 = $100 / 2 = 50$ 秒。

例题 2：

小猫在起点，需要先向右转 90 度，然后向前走 30 个单位到达 A 点，接着向左转 90 度，再向前走 20 个单位到达 B 点。请编写程序控制小猫完成这个路径。

答案：

1. 右转 90 度

2. 向前走 30 个单位

3. 左转 90 度

4.

向前走 20 个单位

例题 3：

小猫在起点，需要先向前走 10 个单位到达 A 点，然后向左转 90 度，再向前走 15 个单位到达 B 点。如果小猫现在需要返回起点，请编写程序控制小猫完成这个动作。

答案：

1. 向后走 10 个单位
2. 右转 90 度
3. 向后走 15 个单位
4. 左转 90 度

例题 4：

小猫在起点，需要先向前走 20 个单位到达 A 点，然后向右转 90 度，再向前走 20 个单位到达 B 点。如果小猫需要从 B 点回到 A 点，并且改变方向，请编写程序控制小猫完成这个动作。

答案：

1. 向前走 20 个单位
2. 右转 90 度
3. 向前走 20 个单位
4. 左转 180 度

例题 5：

小猫在起点，需要先向前走 10 个单位到达 A 点，然后向左转 90 度，再向前走 10 个单位到达 B 点。小猫现在需要重复这个路径 5 次，请编写程序控制小猫完成。

答案：

1. 向前走 10 个单位
2. 左转 90 度
3. 向前走 10 个单位
4. 左转 90 度
5. 重复步骤 1-4 共 5 次

• 课堂

1. 课堂提问

- 通过提问，检查学生对编程概念的理解程度。
- 设计开放性问题，鼓励学生发散思维，提出自己的想法。
- 根据学生的回答，及时调整教学进度和深度。

2. 观察学生参与度

- 观察学生在课堂活动中的参与情况，如小组讨论、编程实践等。
- 评估学生是否能够积极参与，是否能够与同学合作解决问题。
- 根据观察结果，提供针对性的指导和帮助。

3. 实时反馈

- 在学生编程过程中，提供实时反馈，帮助学生纠正错误。
- 鼓励学生提问，及时解答学生的疑问。
- 通过示范和解释，帮助学生理解复杂的概念。

4. 课堂测试

- 定期进行课堂测试，评估学生对编程知识的掌握情况。

- 测试可以包括编程指令的记忆、简单程序的编写等。
-

根据测试结果，调整教学策略，确保学生能够跟上教学进度。

5. 学生自评与互评

- 引导学生进行自评，反思自己的学习过程和成果。
- 组织学生进行互评，学会欣赏他人的优点，同时也发现自身的不足。
- 通过自评和互评，提高学生的自我管理和评价能力。

6. 作业评价

- 对学生的编程作业进行详细批改，包括程序的准确性、代码的整洁性、逻辑的合理性等。
- 提供具体的反馈意见，指出学生的优点和需要改进的地方。
- 鼓励学生在作业中展示自己的创意和解决问题的能力。

7. 评价工具与技术

- 利用在线评价系统，方便快捷地记录和追踪学生的学习进度。
- 结合多媒体技术，如视频、音频等，提供更加丰富的评价内容。
- 利用编程软件内置的评价功能，如错误提示、代码分析等，辅助教学评价。

走进图形化编程的世界第 3 课 猫捉老鼠

课题：		
科目：	班级：	课时：计划 3 课时
教师：	单位：	
一、设计思路		
<p>本节课以“猫捉老鼠”为主题，结合鲁教版五年级下册信息技术课程内容，旨在让学生通过图形化编程的方式，学习编程基础，培养编程思维。课程设计注重理论与实践相结合，通过互动式教学，引导学生自主探究，激发学习兴趣，提高学生信息素养。教学过程分为四个环节：导入新课、自主探究、合作学习、总结反思，确保学生全面掌握知识，提升技能。</p>		
二、核心素养目标		
<ol style="list-style-type: none">1. 计算机科学素养：培养学生理解编程逻辑，掌握基本编程概念，提高问题解决能力。2. 信息意识：引导学生认识到信息技术在生活中的应用，培养信息获取、处理和交流的能力。3. 创新实践：鼓励学生通过编程实践，发挥创造力，解决实际问题，培养创新精神。4. 数字公民素养：教育学生遵守网络道德规范，增强网络安全意识，成为负责任的数字公民。		

三、学习者分析

1. 学生已经掌握了哪些相关知识：学生在进入五年级下册之前，已经学习了基本的计算机操作和简单的信息处理技能。他们对计算机的基本功能和一些常用的软件有所了解，如文字处理、简单图片编辑等。此外，学生可能已经接触过简单的逻辑思维训练，如谜语、游戏等，这有助于他们在编程学习中理解逻辑关系。
2. 学生的学习兴趣、能力和学习风格：五年级学生的好奇心强，对新鲜事物充满兴趣，特别是与游戏和动画相关的内容。他们的逻辑思维能力逐渐增强，能够理解简单的编程逻辑。学习风格上，学生表现出较强的动手操作能力和合作学习意愿，但也存在个体差异，部分学生可能更倾向于独立思考和操作。
3. 学生可能遇到的困难和挑战：学生在学习图形化编程时，可能会遇到编程逻辑理解困难、编程操作不熟练等问题。此外，对于一些抽象的概念，如变量、循环等，学生可能难以在短时间内掌握。因此，教学中需要注重实际操作和实例讲解，通过逐步引导帮助学生克服困难。

四、教学资源

- 软硬件资源：计算机教室、图形化编程软件（如 Scratch、Alice 等）、投影仪、音响设备
- 课程平台：学校内部教学平台，用于发布课程资料和作业
- 信息化资源：图形化编程教程视频、相关案例和编程素材库
- 教学手段：PPT 演示文稿、实物教具（如编程小车）、课堂讨论、小组合作学习

五、教学实施过程

1. 课前自主探索

教师活动：

发布预习任务：通过在线平台或班级微信群，发布预习资料（如 PPT、视频、文档等），明确预习目标和要求，例如“了解 Scratch 软件的基本界面和功能”。

设计预习问题：围绕“猫捉老鼠”编程案例，设计问题如“如何设置角色的移动和交互”。

监控预习进度：通过平台功能或学生反馈，监控学生的预习进度，确保预习效果。

学生活动：

自主阅读预习资料：学生按照预习要求，阅读 Scratch 软件的使用说明，了解其基本操作。

思考预习问题：学生尝试自己设计简单的角色移动，记录自己的操作步骤和遇到的问题。

提交预习成果：学生将操作步骤和遇到的问题以笔记或思维导图的形式提交给教师。

教学方法/手段/资源：

自主学习法：学生通过自主阅读和操作，培养自主学习能力。

信息技术手段：利用在线平台，实现预习资源的共享和监控。

2.

课中强化技能

教师活动：

导入新课：通过展示“猫捉老鼠”动画片段，引出编程主题，激发学生兴趣。

讲解知识点：详细讲解 Scratch 中的角色创建、编程块的使用，以及循环、条件判断等编程概念。

组织课堂活动：设计小组合作，让学生共同完成“猫捉老鼠”游戏的基本框架。

解答疑问：针对学生在编程过程中遇到的具体问题，如“如何让猫跟随老鼠移动”，进行及时解答。

学生活动：

听讲并思考：学生认真听讲，思考如何将理论知识应用于实践。

参与课堂活动：学生积极参与小组讨论，共同解决问题，实现编程目标。

提问与讨论：学生在实践中遇到问题时，主动提问并与其他同学讨论解决方案。

教学方法/手段/资源：

讲授法：教师通过讲解，帮助学生理解编程概念。

实践活动法：通过小组合作，让学生在实践中学习编程技能。

合作学习法：通过小组讨论，培养学生的团队合作意识和沟通能力。

3. 课后拓展应用

教师活动：

布置作业：要求学生根据所学知识，独立完成“猫捉老鼠”游戏的完整版本。

提供拓展资源：推荐相关的编程网站和书籍，鼓励学生进一步学习。

反馈作业情况：教师及时批改作业，针对学生的进步和不足给予反馈。

学生活动：

完成作业：学生按照作业要求，完成游戏编程，并测试其功能。

拓展学习：学生利用推荐的资源，学习更高级的编程技巧。

反思总结：学生对自己的编程过程进行反思，总结经验教训。

教学方法/手段/资源：

自主学习法：学生通过自主完成作业，巩固学习效果。

反思总结法：学生通过反思，提升自我学习能力。

六、教学资源拓展

1. 拓展资源：

- 编程书籍：《Scratch 编程入门》、《Scratch 创意编程实例集》等，适合学生自学和拓展。
- 在线编程社区：如 Scratch 官方社区、Code.org 等，提供编程资源和交流平台。
- 图形化编程软件：介绍其他图形化编程软件，如 PICOe、Blockly 等，帮助学生了解编程的多样性。
- 编程游戏：推荐一些编程相关的游戏，如 Code Monkey、Khan Academy 的编程课程等，让学生在游戏中的学习编程。
- 教育视频：查找与编程相关的教育视频，如 TED 演讲、Khan Academy 的视频教程等，丰富学生的知识面。

2. 拓展建议：

-

鼓励学生参加编程比赛和活动，如 Scratch 全国编程大赛、编程马拉松等，提升编程技能。

- 引导学生关注编程领域的最新动态，如人工智能、物联网等，拓宽视野。
- 建议学生阅读编程相关的科普书籍，了解编程的发展历程和未来趋势。
- 鼓励学生跨学科学习，将编程与其他学科相结合，如数学、物理、艺术等，发挥编程的创造力。
- 鼓励学生参与开源项目，了解开源文化，培养团队合作精神。
- 建议学生利用在线编程平台，如 CodePen、JSFiddle 等，进行在线编程练习和展示。
- 引导学生关注编程伦理，培养良好的编程习惯和道德观念。
- 鼓励学生参加编程社团或兴趣小组，与其他编程爱好者交流学习经验。

七、内容逻辑关系

① 本文重点知识点：

- 编程基础概念：变量、循环、条件判断、事件等。
- 图形化编程软件操作：Scratch 界面介绍、编程块使用、角色和舞台设置。
- 猫捉老鼠游戏设计：角色移动、碰撞检测、游戏逻辑实现。

② 本文重点词：

- 编程块：用于实现编程逻辑的图形化组件。
- 事件：触发编程块执行的特定条件。
- 脚本：由编程块组成的程序片段。

③ 本文重点句：

- “在 Scratch 中，我们可以通过编程块来控制角色的行为。”
- “循环可以让角色重复执行某个动作，直到满足特定条件。”
- “条件判断可以让角色根据不同的条件执行不同的动作。”

八、课堂小结，当堂检测

课堂小结：

本节课，我们通过“猫捉老鼠”的案例，学习了图形化编程的基本概念和操作方法。以下是本节课的主要内容：

1. 认识了 Scratch 编程软件的基本界面和功能，了解了编程块的使用。
2. 掌握了变量的概念，学会了如何创建和使用变量。
3. 学习了循环和条件判断的编程块，掌握了它们在程序中的应用。
4. 通过小组合作，成功设计了“猫捉老鼠”游戏的基本框架。

为了巩固所学知识，以下是对本节课的当堂检测：

一、选择题

1. 以下哪个不是 Scratch 中的编程块？（ ）
A. 移动编程块 B. 造型编程块 C. 调音编程块 D. 事件编程块
2. 在 Scratch 中，用于控制角色移动的编程块是？（ ）
A. 移动编程块 B. 造型编程块 C. 调音编程块 D. 事件编程块
3. 以下哪个编程块可以实现重复执行某个动作？（ ）

A. 移动编程块 B. 造型编程块 C. 重复编程块 D.

事件编程块

二、简答题

1. 简述变量在 Scratch 编程中的作用。
2. 列举两种循环编程块，并说明它们在程序中的用途。
3. 如何在 Scratch 中实现猫捉老鼠游戏中的“猫追老鼠”效果？

三、编程题

请使用 Scratch 编程软件，设计一个简单的“猫捉老鼠”游戏，要求如下：

1. 游戏中有一个猫和一个老鼠角色。
2. 猫可以移动和转向。
3. 老鼠可以移动和转向。
4. 当猫碰到老鼠时，游戏结束。

• 反思改进措施

反思改进措施（一）教学特色创新

1. 实践导向：本节课以“猫捉老鼠”游戏为案例，让学生在实践中学习编程，激发了学生的学习兴趣，提高了他们的动手能力。
2. 合作学习：通过小组合作的方式，学生能够互相帮助，共同解决问题，培养了他们的团队合作精神和沟通能力。

反思改进措施（二）存在主要问题

1. 教学深度不足：在讲解编程概念时，可能过于简化，导致学生对一些复杂概念的理解不够深入。
2. 学生个体差异：由于学生的学习基础和兴趣不同，部分学生在编程实践中可能会遇到较大的困难，需要更多的个别指导。
3. 评价方式单一：主要依靠作业和课堂表现来评价学生的学习成果，缺乏多元化的评价方式。

反思改进措施（三）改进措施

1. 深化教学内容：在讲解编程概念时，可以适当增加难度，引入更多的编程技巧和优化方法，帮助学生更深入地理解编程逻辑。
2. 个性化辅导：针对学生的学习差异，可以提供个性化的辅导计划，对于基础较弱的学生，提供更多的练习和指导，确保他们能够跟上课程进度。
3. 多元化评价：采用多元化的评价方式，如课堂表现、小组合作、项目展示等，全面评估学生的学习成果，同时鼓励学生自我评价和反思。
4. 加强家校沟通：与学生家长保持良好的沟通，了解学生在家的学习情况，共同关注学生的成长和发展。
5. 丰富教学资源：利用网络资源和校内资源，为学生提供更多编程实践的机会，如编程比赛、项目合作等，激发学生的学习热情。
6. 鼓励创新思维：在教学过程中，鼓励学生发挥创意，设计自己的游戏和程序，培养学生的创新能力和解决问题的能力。

• 典型例题讲解

1. 例题：

在 Scratch 中，小猫想要向右移动 100 步，请编写相应的脚本。

答案：

- 当绿旗被点击时，执行以下动作：
-

移动 100 步

2. 例题：

编写一个脚本，使得小老鼠在舞台中随机移动，并且每次移动的距离在 20 到 50 步之间。

答案：

- 当绿旗被点击时，执行以下动作：
- 将小老鼠的 X 坐标设置为随机数（0 到舞台宽度之间）
- 将小老鼠的 Y 坐标设置为随机数（0 到舞台高度之间）
- 移动一个随机数（20 到 50 步）

3. 例题：

编写一个脚本，使得小猫在遇到小老鼠时，能够发出“喵喵”的声音，并停止移动。

答案：

- 当小猫与老鼠发生碰撞时，执行以下动作：
- 发出“喵喵”的声音
- 停止所有动作

4. 例题：

编写一个脚本，使得小老鼠在遇到障碍物时，能够改变方向，继续前进。

答案：

- 当小老鼠与某个障碍物发生碰撞时，执行以下动作：
- 转向 90 度（向左转）
- 移动 5 步
- 转回原来的方向

5. 例题：

编写一个脚本，使得小猫在捕捉到小老鼠后，能够重新开始游戏，即小猫和小老鼠回到初始位置。

答案：

- 当小老鼠被小猫捕捉时，执行以下动作：
- 将小猫的 X 坐标和 Y 坐标重置为初始位置
- 将小老鼠的 X 坐标和 Y 坐标重置为初始位置
- 重置小猫的移动和转向状态
- 重置小老鼠的移动和转向状态

走进图形化编程的世界第 4 课 幸运大转盘

课题：

科目：

班级：

课时：计划 3 课时

教师：

单位：

一、课程基本信息
<p>1. 课程名称：小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册走进图形化编程的世界第 4 课 幸运大转盘</p> <p>2. 教学年级和班级：五年级</p> <p>3. 授课时间：2022 年 10 月 14 日，星期五，第 2 节课</p> <p>4. 教学时数：1 课时</p>
二、核心素养目标
<p>1. 技术意识：培养学生对图形化编程的兴趣，认识到编程在现实生活中的应用价值，提高学生对信息技术发展的敏感性和前瞻性。</p> <p>2. 问题解决：通过设计“幸运大转盘”游戏，引导学生运用编程思维分析问题、解决问题，提升逻辑思维和创新力。</p> <p>3. 信息素养：培养学生获取、评估、使用信息的能力，学会在编程过程中查阅资料，分享学习成果，提高信息交流与协作能力。</p> <p>4. 创新实践：鼓励学生在编程过程中发挥想象，创新设计，通过实践操作提高动手能力和创新精神。</p>
三、学习者分析
<p>1. 学生已经掌握了哪些相关知识：</p> <p>学生在之前的学习中已经接触过基本的计算机操作和图形化编程的基本概念，如变量、循环、条件语句等。他们能够使用图形化编程软件进行简单的编程操作，具备一定的逻辑思维能力和问题解决能力。</p> <p>2. 学生的学习兴趣、能力和学习风格：</p> <p>五年级学生对新鲜事物充满好奇心，对信息技术课程有较高的学习兴趣。他们在学习过程中表现出较强的动手操作能力和模仿能力，善于通过观察和模仿来学习新技能。部分学生可能对编程有较高的热情，愿意主动探索和学习。</p> <p>3. 学生可能遇到的困难和挑战：</p> <p>在学习“幸运大转盘”这一课时，学生可能面临以下困难和挑战：理解编程逻辑的复杂性，特别是在处理随机事件时；设计游戏界面和规则时可能缺乏创意；在调试程序时遇到错误难以定位。此外，对于一些学生来说，编程语言的理解和应用可能存在一定的困难，需要教师给予适当的指导和帮助。</p>
四、教学资源
<p>1. 软硬件资源：图形化编程软件（如 Scratch、Blockly 等）、电脑或平板电脑、投影仪。</p> <p>2. 课程平台：学校信息技术课程平台，用于发布教学资料和作业。</p> <p>3. 信息化资源：网络资源，如编程教程、游戏设计案例、编程相关的视频教程。</p> <p>4. 教学手段：实物教具（如转盘模型）、PPT 课件、教学活动卡片。</p>
五、教学实施过程

1. 课前自主探索

教师活动：

- 发布预习任务：通过学校信息技术课程平台，发布关于“幸运大转盘”编程的预习资料，包括软件操作指南和基础代码示例。
- 设计预习问题：提出问题，如“如何设计一个简单的转盘游戏？”和“转盘如何随机转动？”引导学生思考编程逻辑。
- 监控预习进度：通过平台反馈和学生提交的预习成果，了解学生的预习情况。

学生活动：

- 自主阅读预习资料：学生阅读预习资料，了解图形化编程的基本操作和转盘游戏的设计思路。
- 思考预习问题：学生尝试在软件中操作，解决预习问题，记录遇到的困难和疑问。

方法/手段/资源：

- 自主学习法：学生通过自学掌握基础知识。
- 信息技术手段：利用在线平台进行资料共享和进度监控。

作用与目的：

- 为课堂学习奠定基础，让学生对编程有初步的了解和兴趣。

2. 课中强化技能

教师活动：

- 导入新课：展示一个简单的转盘游戏，激发学生的兴趣，引出本节课的主题。
- 讲解知识点：详细讲解如何使用图形化编程软件创建转盘，包括如何设置转盘的分割、随机转动和交互等。
- 组织课堂活动：将学生分成小组，每个小组设计一个转盘游戏，并现场演示。

学生活动：

- 听讲并思考：学生认真听讲，记录关键步骤和技巧。
- 参与课堂活动：学生积极参与小组讨论，共同完成游戏设计。

方法/手段/资源：

- 讲授法：教师详细讲解编程步骤。
- 实践活动法：通过小组合作，学生实践编程技能。

作用与目的：

- 帮助学生掌握转盘游戏的设计和编程技巧，强化编程技能。

3. 课后拓展应用

教师活动：

- 布置作业：要求学生设计一个更复杂的转盘游戏，如带有不同奖励的转盘。
- 提供拓展资源：推荐相关编程网站和书籍，供学生进一步学习。

学生活动：

- 完成作业：学生根据作业要求，独立完成转盘游戏的设计。
- 拓展学习：学生利用拓展资源，学习更多编程知识。

方法/手段/资源：

- 自主学习法：学生通过自主学习完成作业。

-

反思总结法：学生通过反思作业，总结学习经验。

作用与目的：

- 巩固所学知识，提高编程能力，激发学生的学习兴趣。

六、拓展与延伸

六、拓展与延伸

1. 提供与本课程内容相关的拓展阅读材料：

《图形化编程入门》（作者：张华）

《Scratch 编程教程》（作者：李明）

《编程思维与算法基础》（作者：王强）

《Python 编程从入门到实践》（作者：刘伟）

《游戏编程实战》（作者：赵磊）

这些书籍不仅涵盖了图形化编程的基础知识，还提供了实际案例和项目，有助于学生深入理解和应用所学知识。

2. 鼓励学生进行课后自主学习和探究：

- a. 学生可以根据自己的兴趣选择阅读材料，深入学习编程语言和工具。
- b. 鼓励学生参与在线编程社区，与其他编程爱好者交流心得，共同进步。
- c. 学生可以尝试设计更多具有创意的编程项目，如动画、游戏、数据可视化等。
- d. 引导学生思考编程在现实生活中的应用，如智能家居、物联网、虚拟现实等。
- e. 鼓励学生参加编程竞赛和活动，提高编程技能，拓宽知识面。
- f. 学生可以尝试将编程与艺术、音乐、数学等其他学科相结合，探索跨学科的学习方式。
- g. 鼓励学生关注国内外编程领域的最新动态，了解编程技术的发展趋势。
- h. 学生可以尝试将所学编程知识应用于实际项目中，解决实际问题，提高实践能力。
- i. 引导学生思考编程伦理和隐私保护等问题，培养良好的编程道德。
- j. 学生可以尝试开发开源项目，为社区贡献自己的力量。

通过以上拓展与延伸活动，学生不仅能够巩固课堂所学知识，还能够培养自主学习能力、创新精神和实践能力。同时，这些活动有助于拓宽学生的知识面，激发他们对编程的兴趣，为未来的学习和职业发展奠定基础。

七、反思改进措施

反思改进措施（一）教学特色创新

1. 实践导向教学：在课程设计中，我注重将理论知识与实际操作相结合，让学生通过设计“幸运大转盘”这样的项目来巩固编程知识，这种实践导向的教学方式能够提高学生的动手能力和问题解决能力。

2.

小组合作学习：我采用了小组合作的学习模式，让学生在团队中互相学习、互相帮助，这样的方式不仅培养了学生的团队合作精神，也让他们学会了如何与他人沟通和协作。

反思改进措施（二）存在主要问题

1. 教学深度不足：在讲解编程知识点时，我发现部分学生难以理解复杂的编程逻辑，这说明我在教学过程中可能没有充分考虑到学生的个体差异，教学深度不够。
2. 课堂互动性不强：虽然我尝试通过小组讨论和实践活动来增强课堂互动，但实际效果并不理想，部分学生参与度不高，这可能是因为没有有效地激发学生的兴趣和参与热情。
3. 教学评价单一：目前主要依赖作业和课堂表现来评价学生的学习效果，这种评价方式不够全面，无法全面反映学生的学习情况和进步。

反思改进措施（三）

1. 个性化教学：针对学生个体差异，我将采用分层教学的方法，为不同层次的学生提供相应的学习资源和支持，确保每个学生都能在自己的水平上得到提升。
2. 丰富课堂互动：为了提高课堂互动性，我将设计更多趣味性和挑战性的编程任务，同时鼓励学生提出问题和分享自己的创意，以激发他们的学习兴趣和参与度。
3. 多元化评价体系：我将建立多元化的教学评价体系，包括课堂表现、作业质量、项目完成情况、自我评价和同伴评价等，以更全面地评估学生的学习成果。
4. 加强师生沟通：我将定期与学生交流，了解他们的学习需求和困难，及时调整教学策略，确保教学效果。
5. 拓展教学资源：我将利用网络资源，如在线课程、编程论坛等，为学生提供更多的学习资源和实践机会，帮助他们拓宽视野，提高编程技能。

八、板书设计

① 本节课重点知识点：

- 图形化编程基础概念
- 转盘游戏设计步骤
- 编程逻辑与控制语句

② 关键词和句子：

- ① 图形化编程：通过图形和模块实现编程
- ② 转盘设计：设置分割、随机转动和交互
- ③ 编程逻辑：理解程序流程，实现功能
- ④ 控制语句：if-else、循环等
- ⑤ 算法思维：分析问题，设计解决方案

③ 代码示例：

- 初始化转盘分割
- 设置随机转动逻辑
- 触发事件与响应处理

课题：		
科目：	班级：	课时：计划 3 课时
教师：	单位：	
一、教学内容分析		
<p>1. 本节课的主要教学内容为《小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册》第 5 课《乌鸦喝水》的图形化编程实践。学生将通过编程让乌鸦喝到水，学习如何使用编程工具进行逻辑思考和问题解决。</p> <p>2. 教学内容与学生已有知识的联系：学生在之前的学习中已掌握基本的图形化编程概念和操作，本节课将在此基础上，引导学生运用所学知识，通过编程实现实际情境的模拟。</p>		
二、核心素养目标		
<p>1. 培养学生的计算思维，通过编程解决实际问题，提升逻辑推理和问题解决能力。</p> <p>2. 增强学生的创新意识，鼓励学生尝试不同的编程思路，激发创造力和想象力。</p> <p>3. 培养学生的信息意识，让学生认识到编程在现实生活中的应用，增强信息素养。</p> <p>4. 增进学生的合作与交流能力，通过小组合作完成编程任务，学会分享和协作。</p> <p>5. 提升学生的实践能力，通过动手实践，将理论知识转化为实际操作技能。</p>		
三、教学难点与重点		
<p>1. 教学重点</p> <ul style="list-style-type: none"> - 理解编程逻辑：本节课的核心内容是让学生理解编程的逻辑，通过编写指令让乌鸦移动到水瓶旁边并喝水。重点是让学生学会使用编程语言中的基本指令，如移动、旋转、判断等，来实现这个简单的动画效果。 - 编程实践操作：学生需要能够操作图形化编程工具，将抽象的编程逻辑转化为具体的操作步骤，包括设计角色、设置路径、编写条件语句等。 <p>2. 教学难点</p> <ul style="list-style-type: none"> - 理解条件语句：对于五年级学生来说，理解条件语句（如“如果...那么...”）的运用是一个难点。难点在于如何让学生理解条件语句在不同情况下如何影响程序流程。 - 程序调试：学生在编程过程中可能会遇到程序运行不正确的问题，难点在于如何引导学生进行程序的调试，找出错误并修正。 - 解决实际问题：将编程逻辑与实际情境相结合，让学生理解编程不是简单的代码堆砌，而是用来解决实际问题的工具，这一过程可能对学生来说是一个挑战。 		
四、教学方法与手段		
<p>1. 教学方法：</p> <ul style="list-style-type: none"> - 互动式教学：通过提问、讨论等方式，引导学生积极参与课堂，激发学习兴趣。 - 案例教学：以《乌鸦喝水》故事为案例，让学生在实践中学习编程逻辑。 - 		

项目式学习：以完成乌鸦喝水程序为项目，培养学生解决问题的能力。

2. 教学手段：

- 多媒体演示：利用多媒体展示编程过程，直观地展示编程逻辑。
- 编程软件操作：提供图形化编程软件，让学生动手实践，提高操作技能。
- 教学资源共享：通过在线平台分享教学资源和编程案例，方便学生课后复习和拓展学习。

五、教学过程设计

1. 导入新课（5 分钟）

目标：引起学生对图形化编程的兴趣，激发其探索欲望。

过程：

开场提问：“同学们，你们知道编程是什么吗？它在我们的生活中有什么作用？”

展示一些关于图形化编程的应用实例，如手机游戏、动画制作等，让学生初步感受图形化编程的魅力或特点。

简短介绍图形化编程的基本概念和重要性，为接下来的学习打下基础。

2. 图形化编程基础知识讲解（10 分钟）

目标：让学生了解图形化编程的基本概念、组成部分和原理。

过程：

讲解图形化编程的定义，包括其主要组成元素或结构，如角色、场景、指令等。

详细介绍图形化编程的组成部分或功能，使用图表或示意图帮助学生理解。

3. 图形化编程案例分析（20 分钟）

目标：通过具体案例，让学生深入了解图形化编程的特性和重要性。

过程：

选择几个典型的图形化编程案例进行分析，如《乌鸦喝水》程序的设计与实现。

详细介绍每个案例的背景、特点和意义，让学生全面了解图形化编程的多样性或复杂性。

引导学生思考这些案例对实际生活或学习的影响，以及如何应用图形化编程解决实际问题。

4. 学生小组讨论（10 分钟）

目标：培养学生的合作能力和解决问题的能力。

过程：

将学生分成若干小组，每组选择一个与图形化编程相关的主题进行深入讨论，如“如何设计一个有趣的游戏”。

小组内讨论该主题的现状、挑战以及可能的解决方案。

每组选出一名代表，准备向全班展示讨论成果。

5. 课堂展示与点评（15 分钟）

目标：锻炼学生的表达能力，同时加深全班对图形化编程的认识和理解。

过程：

各组代表依次上台展示讨论成果，包括主题的现状、挑战及解决方案。

其他学生和教师对展示内容进行提问和点评，促进互动交流。

教师总结各组的亮点和不足，并提出进一步的建议和改进方向。

6.

课堂小结（5 分钟）

目标：回顾本节课的主要内容，强调图形化编程的重要性和意义。

过程：

简要回顾本节课的学习内容，包括图形化编程的基本概念、组成部分、案例分析等。

强调图形化编程在现实生活或学习中的价值和作用，鼓励学生进一步探索和应用图形化编程。

布置课后作业：让学生尝试使用图形化编程工具，设计一个简单的动画或游戏，以巩固学习效果。

7. 课后拓展活动

目标：激发学生的学习兴趣，拓展学生的编程技能。

过程：

提供一些图形化编程的在线资源和教程，鼓励学生在课后自主学习。

组织学生参加图形化编程的比赛或活动，提高学生的实践能力和团队协作能力。

六、学生学习效果

学生学习效果主要体现在以下几个方面：

1. 编程技能的提升

2. 计算思维的培养

学生在编程过程中，需要运用逻辑思维和问题解决能力，分析问题、设计解决方案，并不断调试程序以解决问题。这种过程有助于培养学生的计算思维，提高他们在面对复杂问题时进行分析和解决的能力。

3. 创新能力的激发

在案例分析和小组讨论环节，学生需要发挥自己的想象力，设计有趣的编程项目，并提出创新性的想法。这种活动有助于激发学生的创新意识，培养他们的创造力和创新能力。

4. 信息素养的提高

5. 合作与交流能力的增强

在小组讨论和课堂展示环节，学生需要与同伴合作，共同完成任务。这种合作学习有助于提高学生的团队协作能力，同时通过展示和交流，学生能够学会如何表达自己的想法，增强沟通技巧。

6. 自主学习能力的培养

课后拓展活动鼓励学生自主学习，通过在线资源和教程，学生可以继续探索图形化编程的更多可能性。这种自主学习能力的培养有助于学生形成终身学习的习惯。

7. 综合应用能力的提升

学生在完成课后作业时，需要将所学知识应用于实际项目中，如设计动画或游戏。这种综合应用能力的提升有助于学生将理论知识与实践相结合，提高他们的实际操作能力。

8. 学习兴趣的增强

总之，通过本节课的学习，学生在编程技能、计算思维、创新能力、信息素养、合作交流、自主学习、综合应用和学习兴趣等方面取得了显著的学习效果。这些效果

将有助于学生在未来的学习和生活中更好地应用信息技术。

七、重点题型整理

1. 题型一：编程指令的应用

- 题目：在《乌鸦喝水》案例中，如何使用编程指令让乌鸦移动到水瓶旁边并喝水？

- 答案：首先，使用“移动”指令让乌鸦向前移动一定距离；然后，使用“判断”指令检查乌鸦是否到达水瓶旁边；最后，使用“喝水”动作指令让乌鸦喝水。

2. 题型二：条件语句的使用

- 题目：在《乌鸦喝水》案例中，如何使用条件语句判断乌鸦是否到达水瓶旁边？

- 答案：可以使用“如果...那么...”的条件语句，当乌鸦与水瓶的距离小于某个值时，执行移动到水瓶旁边的动作。

3. 题型三：循环结构的运用

- 题目：在《乌鸦喝水》案例中，如果乌鸦需要多次移动才能到达水瓶旁边，如何使用循环结构简化编程？

- 答案：可以使用“重复执行”的循环结构，设置一个次数限制，让乌鸦在达到水瓶旁边之前重复执行移动指令。

4. 题型四：角色属性的设置

- 题目：在《乌鸦喝水》案例中，如何设置乌鸦和水瓶的属性，使其在编程中具有不同的功能和表现？

- 答案：可以为乌鸦设置移动速度、喝水动画等属性，为水瓶设置位置、高度等属性，以实现不同的编程效果。

5. 题型五：程序调试技巧

- 题目：在编程过程中，如果乌鸦无法到达水瓶旁边，如何进行程序调试？

- 答案：首先，检查编程指令是否正确，如移动距离、条件判断等；其次，使用调试工具逐步执行程序，观察乌鸦的移动轨迹；最后，根据调试结果调整编程指令，直至程序运行正常。

八、教学反思与改进

教学反思是教师自我提升的重要环节，通过反思，我们可以更好地了解教学效果，发现问题，从而不断改进教学方法。以下是我对本次《乌鸦喝水》图形化编程教学的一些反思和改进措施。

1. 学生参与度的反思

在课堂上，我发现部分学生对编程的兴趣并不高，参与度不够。这可能是因为编程对于一些学生来说是一个全新的领域，他们可能觉得难以理解或掌握。为了提高学生的参与度，我计划在未来的教学中采取以下措施：

- 在导入环节，增加与生活实际相关的编程案例，让学生感受到编程的实用价值。

- 设计更具有趣味性的编程任务，激发学生的学习兴趣。

- 鼓励学生大胆尝试，不怕犯错，培养他们的自信心。

2. 教学方法的有效性反思

本次教学中，我主要采用了互动式教学和案例分析法。虽然取得了一定的效果，但仍有改进空间。以下是我的改进措施：

-

在讲解编程基础知识时，可以采用更多样化的教学方法，如小组讨论、角色扮演等，让学生在轻松愉快的氛围中学习。

- 在案例分析环节，可以让学生分组进行，让他们在合作中学习，提高团队协作能力。

3. 学生个体差异的关注

在教学过程中，我注意到学生的个体差异较大。有的学生编程基础较好，能够迅速掌握编程技巧；而有的学生则相对较弱，需要更多的指导和帮助。为了更好地关注学生的个体差异，我计划：

- 在课后提供个性化辅导，针对不同学生的需求进行指导。

- 设计不同难度的编程任务，让每个学生都能找到适合自己的学习内容。

4. 教学资源的利用

在本次教学中，我主要利用了图形化编程软件作为教学资源。然而，我发现部分学生对于软件的操作并不熟练，影响了他们的学习效果。为了提高教学资源的利用率，我计划：

- 在课前对学生进行软件操作培训，确保他们能够熟练使用编程软件。

- 探索更多适合图形化编程的教学资源，如视频教程、在线课程等，为学生提供更丰富的学习材料。

5. 教学评价的多样性

在教学评价方面，我主要关注学生的编程结果，而忽略了他们在学习过程中的努力和进步。为了更全面地评价学生的学习效果，我计划：

- 在教学过程中，关注学生的学习态度和进步，给予积极的反馈。

- 设计多样化的评价方式，如课堂表现、小组合作、编程作品等，全面评价学生的学习成果。

走进图形化编程的世界第 6 课 成语大赛

课题：		
科目：	班级：	课时：计划 3 课时
教师：	单位：	
一、教学内容		
教材章节：小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册第 6 课“成语大赛”		
内容：本节课通过“成语大赛”这一有趣的活动，让学生了解和掌握图形化编程的基本概念和操作。课程内容包括：学习使用图形化编程软件，了解编程语言的基本结构；通过编程实现成语收集、分类和竞赛等功能；培养学生的逻辑思维能力和团队		

合作精神。

二、核心素养目标
<ol style="list-style-type: none">1. 培养学生的计算思维，通过图形化编程活动，学会分解问题、设计算法。2. 提升学生的创新实践能力，鼓励学生运用编程技能解决实际生活问题。3. 增强学生的合作学习意识，在编程团队中学会沟通与协作。4. 培养学生的信息意识，学会在编程过程中收集、处理成语信息。
三、学情分析
<p>五年级的学生正处于从形象思维向抽象思维过渡的关键时期，他们在信息技术的学习上既有一定的兴趣，也具备一定的动手能力。然而，由于图形化编程对于他们来说是一个相对较新的概念，因此在知识储备和操作技能上可能存在以下特点：</p> <p>知识方面：学生对编程语言的基础概念可能了解不多，但对于成语这一文化知识有较好的积累，这将为编程中的成语应用提供便利。</p> <p>能力方面：学生的动手操作能力较强，能够通过直观的图形化界面进行编程尝试。但在逻辑思维和问题解决能力方面，可能需要教师的引导和启发。</p> <p>素质方面：学生普遍具备良好的学习兴趣和合作精神，但在独立思考和批判性思维方面，可能需要更多的培养。</p> <p>行为习惯：学生在课堂上通常能保持较好的纪律，但在编程实践过程中，可能存在急于求成、缺乏耐心的情况。</p> <p>对课程学习的影响：学生的这些特点使得本节课的教学需要在传授编程知识的同时，注重培养他们的逻辑思维、创新实践能力和团队合作精神，同时也要引导他们养成良好的编程习惯，提高解决问题的能力。</p>
四、教学方法与手段
<ol style="list-style-type: none">1. 讲授法：通过讲解图形化编程的基本概念和操作流程，帮助学生建立初步的认知框架。2. 实验法：引导学生通过实际操作，如设计成语收集和竞赛功能，将理论知识应用于实践。3. 讨论法：组织学生分组讨论编程过程中遇到的问题，鼓励他们分享解决方案，提高团队合作能力。 <p>教学手段：</p> <ol style="list-style-type: none">1. 多媒体设备：使用投影仪展示编程界面和操作步骤，便于学生直观学习。2. 教学软件：利用图形化编程软件，让学生在软件环境中进行编程实践。3. 互动平台：利用网络教学平台，进行在线编程练习和交流，提高学习的互动性和趣味性。
五、教学过程设计

(一) 导入环节 (5 分钟)

1. 创设情境：展示一段成语故事的视频，激发学生对成语的兴趣。
2. 提出问题：引导学生思考成语在生活中的应用，以及如何用编程来展示成语的魅力。
3. 小组讨论：分组讨论成语在编程中的应用可能性，每组派代表分享观点。

(二) 讲授新课 (15 分钟)

1. 教师讲解：介绍图形化编程的基本概念和操作流程，强调编程的步骤和逻辑性。
2. 展示案例：展示一个简单的成语收集程序，分析其功能和实现方法。
3. 互动提问：针对展示的案例，提出问题引导学生思考编程思路。

(三) 动手实践 (20 分钟)

1. 分组编程：学生分组，每组使用图形化编程软件，开始设计成语大赛程序。
2. 教师指导：教师在学生编程过程中进行个别指导，帮助学生解决编程难题。
3. 小组展示：每组展示自己的成语大赛程序，介绍程序的功能和设计思路。

(四) 巩固练习 (10 分钟)

1. 编程任务：教师布置一个简单的编程任务，如增加成语难度等级。
2. 小组协作：学生分组完成编程任务，教师巡视指导。
3. 交流分享：每组分享自己的编程成果，其他学生提出改进建议。

(五) 课堂提问 (5 分钟)

1. 教师提问：针对编程过程中的重难点，提出问题引导学生深入思考。
2. 学生回答：学生积极回答问题，展示自己的理解。

(六) 总结与反思 (5 分钟)

1. 教师总结：回顾本节课的学习内容，强调图形化编程的重要性和应用价值。
2. 学生反思：学生总结自己在编程过程中的收获和不足，提出改进措施。

(七) 课后作业 (5 分钟)

1. 布置作业：教师布置相关的编程练习，巩固所学知识。
2. 学生记录：学生记录作业内容，明确学习目标。

总用时：45 分钟

六、教学资源拓展

1. 拓展资源：

- 图形化编程软件的介绍：介绍几种常用的图形化编程软件，如 Scratch、Alice 等，这些软件提供了直观的编程界面，适合小学生学习和使用。
- 成语文化背景：收集一些与成语相关的历史故事、成语典故，以及成语在现代生活中的应用案例，丰富学生对成语文化的理解。
- 编程案例库：提供一些经典的编程案例，如简单的游戏、动画等，让学生在课余时间进行模仿和创作。

2. 拓展建议：

- 成语故事收集：鼓励学生收集自己喜欢的成语故事，并尝试用图形化编程的方式呈现故事情节。
- 编程比赛设计：学生可以设计自己的成语大赛游戏，包括成语的分类、难度设置、得分规则等，培养学生的创新设计能力。

-

成语卡片制作：利用编程知识，制作成语卡片，包括成语、释义、故事等内容，提高学生的信息处理和呈现能力。

- 编程社群参与：鼓励学生加入编程社群，与其他编程爱好者交流学习，拓宽编程视野。

- 家庭编程活动：家长可以参与学生的编程学习，共同完成一些简单的编程项目，增进亲子关系的同时，提高学生的编程技能。

- 校内编程俱乐部：学校可以设立编程俱乐部，定期举办编程活动，让学生在轻松愉快的氛围中学习编程知识。

- 编程比赛准备：学生可以参与校内外的编程比赛，将所学知识应用于实际竞赛中，锻炼自己的编程能力。

- 编程资源库建设：学生可以共同创建一个编程资源库，收集和整理编程学习资料，方便大家共同学习和分享。

七、课堂小结，当堂检测

课堂小结：

1. 回顾本节课的学习内容，强调图形化编程在成语应用中的重要性。
2. 总结成语大赛程序的设计思路和实现步骤，让学生明白编程的逻辑性和步骤性。
3. 强调编程过程中的团队合作和沟通的重要性，鼓励学生在以后的学习中积极与他人交流合作。
4. 提醒学生在编程过程中保持耐心和细心，遇到问题时要善于思考和分析。

当堂检测：

1. 成语编程理解检测：

- 学生独立完成一个简单的成语编程任务，如设计一个成语查询程序。
- 教师巡视指导，确保学生能够理解编程任务的要求。

2. 编程实践技能检测：

- 学生分组合作，完成一个成语收集和展示的程序。
- 教师观察学生编程过程中的操作，评估学生的编程技能。

3. 编程思维拓展检测：

- 教师提出一个与成语相关的编程问题，如设计一个成语接龙游戏。
- 学生独立思考，提出解决方案，展示自己的编程思维。

4. 课堂互动检测：

- 教师提问，检查学生对本节课内容的掌握情况。
- 学生回答问题，教师根据回答情况给予反馈和指导。

5. 作业布置：

- 布置一个与成语相关的编程作业，如设计一个成语接龙游戏或成语猜谜程序。
- 强调作业的完成时间和提交方式，确保学生能够按时完成。

八、板书设计

① 知识点：

- 图形化编程简介
- 成语大赛程序设计思路

-

编程基本步骤

② 词：

- 编程
- 图形化界面
- 成语
- 算法
- 逻辑
- 功能模块

③ 句：

- 图形化编程是一种直观、易学的编程方式。
- 成语大赛程序需要实现成语收集、分类和竞赛等功能。
- 编程步骤包括需求分析、设计、编码、测试和调试。

走进图形化编程的世界第 7 课 一起去寻宝

课题：		
科目：	班级：	课时：计划 3 课时
教师：	单位：	
一、课程基本信息		
1. 课程名称：小学信息技术(信息科技)鲁教版（五六年制通用）五年级下册走进图形化编程的世界第 7 课 一起去寻宝		
2. 教学年级和班级：五年级		
3. 授课时间：2022 年 X 月 X 日 星期 X 上午第一节		
4. 教学时数：1 课时		
二、核心素养目标		
1. 创新意识：培养学生运用图形化编程工具进行创造性设计和解决问题的能力，激发学生对信息技术创新的兴趣。		
2. 信息意识：通过编程实践，使学生认识到信息技术在解决问题中的重要作用，提高收集、处理和运用信息的能力。		
3. 计算思维：通过图形化编程的学习，培养学生逻辑思维和抽象思维能力，学会用编程语言描述问题和解决问题。		
4. 数字化学习与交流：使学生学会利用图形化编程进行数字化创作和表达，提升信息交流与合作的能力。		
5. 信息伦理与安全意识：在编程过程中，教育学生遵守网络道德规范，增强信息安全		

全意识。

三、学习者分析

1. 学生已经掌握了哪些相关知识：

学生在五年级下学期之前已经学习了基本的计算机操作和简单的文字处理，具备一定的信息检索和基本的使用网络资源的能力。此外，他们可能已经接触过一些简单的图形化编程工具，如 Scratch 等，对编程有初步的认识。

2. 学生的学习兴趣、能力和学习风格：

五年级学生的好奇心强，对新技术和新事物充满兴趣。他们在学习过程中表现出较强的动手实践能力，喜欢通过操作来学习和理解新知识。学习风格上，部分学生可能更偏向于视觉学习，通过观察和操作来学习；而另一部分学生可能更倾向于逻辑思维，通过分析和推理来掌握知识。

3. 学生可能遇到的困难和挑战：

在学习图形化编程时，学生可能会遇到编程逻辑理解和代码编写上的困难。对于逻辑思维较弱的学生来说，理解编程流程和编写代码可能是一个挑战。此外，学生在团队合作和交流时，可能会遇到沟通不畅和协作效率不高的问题。因此，教师需要关注这些潜在困难，并提供适当的教学策略和辅导。

四、教学方法与手段

教学方法：

1. 启发式教学：通过提出问题引导学生思考，激发学生的求知欲，帮助他们理解编程逻辑。
2. 案例教学法：通过具体的寻宝游戏案例，让学生在解决问题的过程中学习编程知识。
3. 分组合作学习：将学生分成小组，共同完成编程任务，培养他们的团队协作能力和沟通技巧。

教学手段：

1. 多媒体课件：利用 PPT 展示编程步骤和技巧，直观地呈现教学内容。
2. 编程软件实时演示：通过实时演示编程过程，让学生直观理解编程操作。
3. 网络资源整合：利用网络平台提供的相关资源，如教程视频和在线编程社区，帮助学生自主学习和解决编程难题。

五、教学过程

【导入环节】

（教师）同学们，大家好！今天我们来学习《一起去寻宝》这一课。你们有没有参加过寻宝游戏呢？今天，我们就通过图形化编程的方式来设计一个寻宝游戏，让我们的电脑也能“寻宝”。让我们一起走进图形化编程的世界，开启我们的寻宝之旅吧！

【新课讲授】

一、认识图形化编程

（教师）同学们，我们先来认识一下图形化编程。图形化编程是一种通过图形化界面来编写程序的编程方式，它使用图形块来代替传统的代码，让编程变得更加直观和简单。

（教师）请看大屏幕，这里有一个简单的图形化编程界面。你们能告诉我，这些图形块分别代表什么意思吗？

（学生）老师，这个是“如果...那么...”的判断条件。

（学生）这个是“移动”的动作。

（教师）很好，同学们观察得很仔细。图形化编程就是通过这些图形块来组合成程序，实现各种功能。

二、设计寻宝游戏

（教师）接下来，我们就要开始设计我们的寻宝游戏了。首先，我们需要确定游戏的规则和场景。

（教师）请同学们分成小组，讨论一下你们想要设计的寻宝游戏规则和场景。完成后，请每个小组派代表上来分享你们的想法。

（学生）我们小组设计的游戏规则是：玩家需要找到隐藏在地图上的宝藏，并且解开谜题才能获得宝藏。

（学生）我们的游戏场景是一个迷宫，玩家需要在迷宫中寻找线索。

（教师）很好，你们的想法很有创意。现在，我们来学习如何使用图形化编程工具来设计这个游戏。

三、编写寻宝游戏程序

（教师）首先，我们需要创建一个游戏场景。请同学们打开图形化编程软件，按照你们的设计，创建一个迷宫场景。

（教师）接下来，我们为玩家设定一个角色，并让这个角色在迷宫中移动。

（教师）请同学们思考，当玩家点击屏幕时，角色应该有什么反应？

（学生）老师，当玩家点击屏幕时，角色应该向点击的方向移动。

（教师）很好，那我们就在程序中添加一个“移动”动作，并设置移动方向为点击屏幕的位置。

四、添加游戏逻辑

（教师）现在，我们的游戏场景和角色已经准备好了。接下来，我们需要添加游戏逻辑，让玩家能够找到宝藏并解开谜题。

（教师）请同学们思考，当玩家到达宝藏位置时，应该发生什么？

（学生）老师，当玩家到达宝藏位置时，应该显示一个提示信息，并播放一段音乐。

（教师）很好，那我们就在程序中添加一个判断条件，当玩家到达宝藏位置时，触发提示信息和音乐。

五、测试与优化

（教师）现在，我们的寻宝游戏程序已经基本完成了。请每个小组进行测试，看看游戏是否能够正常运行。

（教师）在测试过程中，如果发现程序有错误或者不足之处，请及时修改和优化。

（学生）我们小组在测试过程中发现，当玩家在迷宫中移动时，有时候会出现卡顿的情况。

（教师）这是一个很好的发现。请同学们分析一下，可能的原因是什么？并提出解决方案。

（学生）老师，我们分析了一下，可能是程序在处理大量数据时出现了卡顿。我们可以尝试优化程序，减少数据处理量。

（教师）很好，你们的分析很到位。现在，请同学们根据分析结果，对程序进行优化。

【总结与反思】

（教师）同学们，今天我们通过图形化编程的方式设计了一个寻宝游戏。在这个过程中，我们不仅学习了图形化编程的基本知识，还锻炼了我们的创新思维和团队协作能力。

（教师）请同学们回顾一下，我们在设计游戏的过程中遇到了哪些困难？又是如何解决的？

（学生）老师，我们遇到了程序卡顿的问题，通过优化程序解决了。

（学生）我们还遇到了游戏逻辑不完善的问题，通过添加判断条件解决了。

（教师）很好，同学们通过自己的努力，成功地解决了这些问题。这充分说明了，只要我们勇于面对困难，善于思考，就一定能够找到解决问题的方法。

（教师）最后，我想提醒大家，图形化编程是一项很有趣的技能。希望大家能够继续学习，不断探索，让我们的编程之旅更加精彩！

【课后作业】

1. 完善你们的寻宝游戏程序，确保游戏能够正常运行。
2. 尝试设计其他类型的图形化编程游戏，并尝试解决其中遇到的问题。
3. 与同学们分享你们的游戏设计心得，互相学习和进步。

六、教学资源拓展

1. 拓展资源：

- 图形化编程基础概念：介绍图形化编程的历史、基本原理和它在教育中的应用。
- 图形化编程软件介绍：探讨不同图形化编程软件的特点，如 Scratch、Blockly、Tynker 等，以及它们在寻宝游戏设计中的应用。
- 编程思维训练案例：提供一些编程思维训练的案例，如逻辑推理、问题解决等，帮助学生提高编程能力。
- 编程游戏设计指南：介绍编程游戏设计的基本流程，包括游戏设计、编程实现、测试与优化等阶段。

2. 拓展建议：

- 学生可以尝试使用不同的图形化编程软件进行游戏设计，比较它们的优缺点，从而加深对编程工具的理解。

- 鼓励学生参与线上编程社区，与其他编程爱好者交流心得，学习他人的游戏设计思路。
- 建议学生阅读相关的编程书籍，如《Scratch 编程入门》、《编程思维训练》等，以提升编程技能和思维。
- 在课堂之外，学生可以参与编程俱乐部或工作坊，与其他同学一起进行编程实践，共同解决编程问题。
- 学生可以尝试将图形化编程应用于其他领域，如科学实验、艺术创作等，以拓展编程的应用范围。
-

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。
。如要下载或阅读全文，请访问：

<https://d.book118.com/945342111201012013>