期末大作业\_7组

# The code Odyssey

## 作者：

黄悦 10224102408

魏 畇 钊 10222140414

王文宇 10224511407

### 摘要

本报告涵盖了一个使用 Pygame 库开发的横版闯关游戏项目的代码分析。该项目实现了包括角色移动、攻击、跳跃、与物品和怪物的交互等游戏机制。游戏具有城市和关卡两个场景，并包括商店、音乐框、故事框等功能。以下是对项目结构和关 键算法的分析。

**一.介绍**

## 1.1

**游戏主要内容介绍**

**故事情节：** 游戏设定了有趣的故事背景，从前，在算法和代码的世界中…小白羊，一位聪明而充满雄心的计算机科学学生， 踏上了一场探索编程和软件开发广阔领域的旅程。以计算机科学学生的冒险为主线，为游戏增加了一层叙事。以键盘为剑， 以IDE为盾，打败了一个又一个bug怪兽，书写自己的代码史诗。

**动态关卡设计：** 游戏中的关卡设计具有动态性，通过不同的关卡切换和怪物生成，为玩家提供了多样化的游戏体验。

**角色与怪物设计：** 打败怪兽不断闯关的小白羊象征着初到计算机学院对计算机相关知识掌握并不多的我们大一学生，打败怪兽的过程也是debug的过程，希望我们能像打这款游戏一样勇敢向前冲，不惧前方的困难。

**角色与怪物行为：** 游戏中的角色和怪物具有丰富的行为设计，包括攻击动画、跳跃动作、移动模式等，增加了游戏的趣味性。怪物的行为不仅包括简单的移动，还包括反转方向、攻击等智能行为，使得游戏中的怪物具有挑战性。

**道具系统：** 游戏中引入了武器系统，包括主角的攻击动作和怪物的攻击机制，为游戏战斗元素增色。游戏中有金币、药水等道具，通过与商店的交互，玩家可以获得加速、恢复生命等效果，增加了游戏的策略性。

**动画效果：** 游戏中运用了动画效果，包括角色和怪物的动画，以及其他游戏元素的动态效果，提高了游戏的可视化水平。

## 1.2

**实现游戏的软件介绍**

**PyCharm：** import pygame, 负责游戏的核心代码逻辑、图形界面、游戏物体的动画等方面的开发和调试工作**Github：** 创建仓库并将源码开源，本源码仓库的地址是 <https://github.com/UiUmi/T.git>

**GitKraken：** 使用GitKraken协同管理GitHub上的仓库，确保团队成员能够同时编写代码而不发生冲突。负责解决代码合并的问题，协助团队保持代码同步。

**markdown：** 编写报告工具，使排版更加美观

## 1.3

### 规则：

1. **主城的按键用法：**

**商店：** 点击商店购买物品，购买的物品在背包中查看

**背包：** 要想使用购买的商品，要在背包中点击使用（比如在商店中购买的人物形象当时只是查看换装效果，并没有实现人物切换，要在背包中使用）

**光盘：** 打开和关闭音乐

**story 书本:** 里面讲述了我们故事发生的背景**MAP：** 进行关卡的选择

1. **药水的使用说明：** H红药水：healthy 药水，一瓶可以补充一颗心S蓝药水：speed 药水，可加速

J绿药水：jump 药水，可跳的更高A黄药水：攻击药水，伤害+1

1. **人物操作说明：** WZAD分别控制人物上下左右走动按键盘J键实现人物攻击敌人

**二.基本架构 (主函数介绍)**

### 游戏初始化：

通过调用pygame.init()函数初始化Pygame库以及一些全局变量的初始化。初始化Pygame和其他相关设置，包括屏幕大小、背景图像、字体等。

定义和初始化一些全局变量，如玩家坐标、生命值、金币数量等。

载入游戏中需要使用的图片和音频资源：使用pygame.image.load()函数加载游戏需要使用的图片资源，并使用pygame.transform.scale()函数调整图片的大小。

使用pygame.mixer.init()函数初始化音频模块，使用pygame.mixer.music.load()函数加载音乐文件，使用pygame. mixer.Sound()函数加载音效文件，并使用play()函数播放音乐。

### 主城工具栏： 商店页面：

在商店页面，根据商品数量动态生成商品的显示位置，并通过鼠标点击购买商品。商店的页面切换通过上一页和下一页 按钮实现。

### 故事框和规则框：

处理故事框的显示，包括文字渲染、逐字显示等。规则框的显示和交互，提供游戏规则和相关信息。**音乐框**

通过点击音乐框按钮，控制音乐的播放和暂停。这些框体的显示和隐藏通过标志位实现。 **背包系统**

### 背包初始化：

背包对象包含一个列表，用于存储不同类型的物品。**物品购买和添加：**

购买成功后，将相应类型的物品对象添加到玩家的背包中。**渲染背包框和物品：**

如果背包框打开，首先渲染背包框的背景图像。遍历玩家背包中的物品列表，计算每个物品在背包框中的位置。使用pygame.image.load()函数加载物品的图像，并使用pygame.transform.scale()函数调整图像的大小。使用blit()函数 将物品图像绘制到屏幕上，并使用font.render()函数渲染物品数量的文本。检查鼠标点击是否在背包中的物品上，并执 行相应的逻辑，如更新主角形象、增加血量等。

### 物品使用与移除：

玩家可以通过背包界面选择要使用的物品。 不同类型的物品有不同的效果，如增加生命值、提高速度等。使用物品后，更新玩家的状态，例如生命值、速度等。

消耗的物品移除后，更新背包中的物品列表。

### 药水使用

加速状态和跳跃药水效果：如果加速状态开启，根据加速持续时间计算剩余时间。当加速持续时间结束时，关闭加速状态， 并恢复主角的正常速度。跳跃药水效果类似，根据跳跃药水的持续时间和原始跳跃速度计算剩余时间，并在结束时恢复正常 跳跃速度。

### 主循环:

游戏的主要逻辑在循环 while running 中执行。 处理游戏中的各种事件，包括鼠标点击、按键操作等。根据事件的类型，执行相应的逻辑，例如切换场景、打开商店、执行角色移动等。

更新游戏状态，包括玩家位置、怪物位置、背景滚动等。 渲染游戏界面，绘制背景、玩家、怪物、UI等。

1. **角色移动和交互** 通过检测键盘按键状态，实现角色的移动和攻击。玩家可以通过按键实现左右移动、攻击等基本操作。同时，角色还可以执行跳跃操作。

### 怪物生成和移动：

定义怪物类，包括怪物的属性（位置、速度、伤害等）。 在不同关卡生成怪物，控制怪物的移动和攻击行为。 处理怪物和玩家的碰撞，更新生命值和其他相关状态。

怪物生成：根据所选的关卡（selected\_level）和当前关卡是否开始（current\_level\_start），选择合适的怪物列表

（exist\_monsters），并设置怪物存在标志（is\_monster\_exist）为True。

怪物移动：根据怪物的位置（m.pos）和移动方向（m.is\_moving和m.is\_facing\_right），更新怪物的位置。 怪物攻击：如果玩家距离怪物较近且一定时间内没有被攻击过，怪物开始攻击。根据怪物的状态（m.is\_attacking和m. is\_facing\_right），更新怪物的位置，并在一定时间后结束攻击动画。如果玩家处于攻击状态（player\_is\_attacking） 并且怪物在攻击范围内，玩家对怪物造成伤害。如果怪物的血量小于等于0，怪物被移除并玩家获得奖励。

### 攻击和碰撞检测：

处理玩家和怪物的攻击动作，包括攻击动画、伤害计算等。 碰撞检测，判断玩家是否与怪物发生碰撞，以及子弹与玩家的碰撞。 处理玩家受伤和怪物受伤后的状态更新。

### 游戏状态控制：

控制游戏的状态，例如切换关卡、进入城市、打开故事框等。 处理特殊物品（如药水）的使用，更新玩家状态。 处理游戏结束条件，例如玩家生命值耗尽。

### 背景滚动和关卡切换：

实现背景滚动效果，让游戏有动态感。 控制关卡切换逻辑，包括生成新的怪物群、切换背景图等。游戏包含城市场景和关卡场景，通过点击地图图标实现场景切换。不同场景有不同的背景和怪物生成逻辑。

### UI和视觉反馈：

绘制并更新UI元素，如生命值、金币数量等。

提供视觉反馈，例如攻击动画、怪物死亡动画等。处理游戏中的事件，例如打开商店、查看地图等。

1. **子弹和子弹移动：** 控制子弹的生成和移动，处理子弹与玩家、怪物的碰撞。 更新子弹的位置，实现子弹的飞行效果。

**三.核心算法、技术、美工、界面部分**

## 一、核心算法

### 角色移动和交互算法

* 1. 角色移动 玩家角色的移动通过检测键盘按键状态实现。按下左右键时，更新玩家的 x 坐标，实现左右移动。同时， 在没有按下左右键时，玩家停止移动。移动时通过切换奔跑动画帧，实现动画效果。
  2. 角色攻击

玩家攻击通过检测 J 键状态实现。按下 J 键时，启动攻击动画，快速移动角色位置，根据角色朝向渲染相应的攻击动画帧。攻击动画持续时间为 0.2 秒 1.3 角色跳跃

角色跳跃分为起跳和下落两个阶段。按下 W 键时，检测是否在地面上，是则进入起跳阶段。在起跳阶段，角色 y 坐标逐渐减小，模拟向上跳跃。下落阶段通过增加 y 坐标模拟重力效果。

### 场景切换技术

* 1. 场景切换逻辑

通过点击地图图标实现场景切换。根据选择的关卡更改背景图片，实现场景的切换。在不同场景下，怪物的生成和移动 逻辑也不同。此外，为实现场景过渡效果，添加了背景图的移动效果。

* 1. 背景图移动

为实现场景过渡效果，背景图的移动通过更新背景的 x 坐标实现。不同场景下背景图的移动速度可以调整，通过两张背景图的交替移动，实现平滑的过渡效果。

### 商店和物品交互技术

* 1. 商店页面

商店页面通过动态生成商品显示位置，实现多页商品展示。鼠标点击上一页和下一页按钮实现页面切换。通过检测商品 位置，判断鼠标点击的是哪个商品，实现购买商品的逻辑。

* 1. 商品购买逻辑

商品购买逻辑包括更新主角的属性和状态，例如增加生命值、速度等。通过检测购买的商品类型，执行相应的逻辑。例 如，购买生命药水时增加主角的生命值，购买速度药水时增加主角的移动速度，并在一定时间内启动加速状态。

### 人物攻击设定

* 1. 设置伤害

给player\_damage赋值即可

* 1. 设置攻击动画

用pygame.time自带的计时工具计时0.5s，在这0.5s内人物想朝向的方向每帧移动10像素，每帧在人物前渲染player attack animation，实现人物攻击

### 怪物设定

* 1. 怪物初始化

创建Monster类，来初始话不同的怪物，目前实现两种小怪和一个Boss，设定其伤害，生命值，速度之类的属性。

* 1. 创建怪物列表

用来存储现在存在的怪物，死亡的pop出去，列表元素是一个类，类包含怪物类别，以及这个怪物的实时数据，坐标， 攻击时间间隔等，以便后来对其操作的判定

5.3 怪 物 攻 击 monster1和玩家类似的攻击方式，持续0.5s，10像素的冲击，monster2则是在人物头顶头像炸弹，对炸弹也生成一 个列表进行检测判定和坐标位置移动，Boss则相当于一个生成桩，boss存在时隔一段时间就会刷出一些怪物。 5.4怪物运动逻辑

设置了怪物的探查范围，当玩家在探查范围外时，怪物会来回随机的运动和停留，当玩家被怪物察觉时，怪物就会冲向 玩家，并在玩家适当范围内开始攻击

### 受伤判定

* 1. 人物

被bullet或被怪物冲撞，人物被攻击后会有0.5s的无敌时间，且人物会被击飞一下

* 1. 怪物

被人物攻击并扣血并被击飞

### 关卡选择

* 1. 点击map选择关卡

检测鼠标点击逻辑，渲染map，并检测点击在哪个关卡上，改变select\_level的值进行进入关卡操作

* 1. 进入关卡

首先替换背景，停止渲染rule等无用的东西，关卡中map也不渲染，然后将初始化好的level1-1的怪物copy进入exiet\_monsters列表

* 1. 场景循环后移动

在level1每个小节怪物死完后，背景循环并进入下一小节

### 关卡选择

* 1. 设置is\_in\_city变量

初始化时设置为True，记录人物是否在主城，进入关卡后设置为False

* 1. 死亡和关卡结束结算

人物死亡后设置为True，每一level结束后也会设置为True

* 1. 当is\_in\_city为True

另外设置背景图为city.png即可

## 二、技术

### 软件技术：

**Pygame库：** 游戏的主要开发基于Pygame库，这是一个专门用于制作2D游戏的Python库。 Pygame提供了处理图形、事件、声音等方面的模块，方便游戏的开发和交互。

### Git版本控制：

使用Git进行版本控制，确保多人协同开发时代码的同步和管理。 GitKraken等可视化工具可用于更直观也更加方便地管理仓库、分支、合并等操作。可以实时看到代码的修改部分，以及方便地上传各类文件。通过将项目代码存储在GitHub仓库 中，我们轻松地跟踪和管理代码的变更，并与成员进行协作开发。GitHub还提供了一系列强大的功能，如问题跟踪、合并 请求和代码审查等，有助于提高开发效率和代码质量。 **Markdown报告：** 使用Markdown编写游戏报告，清晰地记录项目的开发过程、分工情况、亮点、技术细节等。 Markdown提供了简洁的语法和良好的可读性，适合用于撰写文档和报告。**代码技术：** ———

### 游戏循环：

游戏通过一个主循环来运行。每次循环中，检测用户输入、更新游戏状态、绘制图形等。 主循环是保持游戏运行的核心， 通过控制每一帧的逻辑，实现游戏的流畅运行。

### 角色和怪物动画：

游戏中的角色和怪物都通过图像来呈现。动画效果通过在一定时间内切换不同图像帧来实现。 Pygame提供了图像处理和转换的功能，使得角色和怪物的动画效果能够顺利展示。

### 碰撞检测：

玩家与怪物、玩家与子弹之间的碰撞检测是游戏中一个重要的方面。通过检测对象之间的相对位置和尺寸，判断它们是否发 生碰撞。

碰撞检测用于触发攻击、处理伤害、控制关卡切换等关键逻辑。**状态管理：**

游戏中有多个状态，如关卡状态、城市状态、商店状态等。使用状态管理来控制游戏在不同场景之间的切换。 状态管理还包括角色状态、怪物状态，用于控制动画播放、生命值变化等。

### 时间管理与定时器：

游戏中许多效果都涉及到时间，例如攻击动画的持续时间、速度加成的持续时间等。通过Pygame提供的定时器功能来管理 这些效果的时间。

### 事件处理：

游戏通过监听和处理事件来响应用户的输入，例如按键、鼠标点击等。 事件处理用于控制角色移动、攻击触发、物品使用等用户交互的逻辑。

### 数学计算：

游戏中涉及到一些数学计算，例如角色移动的位置计算、子弹的轨迹计算等。 数学计算用于处理游戏中的位置关系、速度变化等。

**美工技术：**

图形资源和设计： 游戏中的图形资源，包括角色、怪物、背景等，需要进行图形设计和美术制作。 游戏设计中要考虑图形的统一风格，以及不同元素之间的协调。 以上是对游戏实现中一些关键技术和实现细节的详细介绍。这些技术共同构成了一个完整的横版闯关游戏。

**三、 美工**

**确定游戏风格和氛围：** 在开始设计之前，了解游戏的整体风格（代码普及、休闲闯关）和目标受众（对代码或闯关游戏感兴趣）。根据游戏的类型（冒险、休闲等）以及故事背景，确定温暖舒适的氛围和风格。

**颜色选择：** 黄色作为一种暖色调能够给人带来温暖、轻松等感受，因此选择黄色为主色调，从而增强游戏带给玩家的亲切舒适感。运用莫兰迪配色涂画主要人物形象。

**创造背景：** 使用PS和画图工具，根据游戏风格和黄色调的要求，构建出一个暖色背景。其中涉及植物、森林、沙漠和学校等元素，并使用黄色搭配其他暖色调进行渲染，以创造出温暖舒适的视觉效果。

**设计人物形象：** 根据游戏的角色设定和风格要求，设计出与游戏背景相符的主角、敌人、NPC等角色人物形象，并运用莫兰迪配色。。

**制作标识：** 设计游戏的标识，游戏的Logo、按钮图标等。这些标识与游戏的整体风格相契合又有突出，便于玩家在界面中快速识别和操作。

**完善细节和效果：** 在设计完成后，对细节进行修饰和优化。会添加光影效果、纹理、阴影等元素，以增强场景和角色的逼真感和立体感。

**四、 界面部分**

游戏的界面包括开始按钮、退出按钮、商店按钮、音乐按钮等。通过 Pygame 的绘制功能实现这些按钮的显示和交互。同时，通过文本渲染实现故事文本和商品价格的显示。

### 背景渲染：

游戏背景通过background\_image加载，使用pygame.transform.scale调整其大小以适应窗口。背景在游戏中会根据角色 和怪物的移动而滚动，以呈现出场景变化的效果。

### 角色和怪物渲染：

主角和怪物的渲染通过SCREEN.blit实现。角色的朝向和怪物的移动方向通过图像的翻转来呈现。主角和怪物的图像通过加 载相应的图片，并使用pygame.transform.scale调整大小。 怪物的生成和行为逻辑在游戏中通过多个关卡的设计来展现。怪物有不同的种类，包括Boss，每种怪物都有独特的属性。

### 图标和文本渲染：

生命值和金币数量通过文本渲染，使用font.render创建文本表面，然后通过SCREEN.blit放置在窗口上。游戏中还包含一些 图标，如心形图标表示生命值，金币图标表示金币数量。 游戏中可能包含其他元素，如商店、音乐框按钮等，它们也通过SCREEN.blit实现。

**子弹渲染：** 游戏中包含子弹，子弹的渲染通过SCREEN.blit实现。子弹的移动和与主角的碰撞逻辑在游戏中得以展示。**背包渲染：**

背包的渲染包括背包框、物品图片以及物品数量的渲染。每个物品通过图片呈现，其数量通过文本表面渲染。渲染的坐标和 排列通过遍历背包物品列表来确定，以保证整齐排列。

### 故事文本和规则框渲染：

游戏中包含故事文本和规则框，它们在游戏进行中根据一定的触发条件出现。故事文本通过字幕的方式逐行显示，规则框则 作为背景显示。

总体上，界面部分通过Pygame提供的渲染函数实现，根据游戏逻辑和状态的变化，动态更新各个元素的位置和状态，使得 整体呈现出生动有趣的游戏画面。

**四.分工以及发展历程心得**

## 分工：

初始规划：黄悦、魏畇钊、王文宇

算法设计与代码编写：黄悦（主要负责界面，如商店功能，背包功能，故事框，音乐框规则框，背景循环，以及坐标数 值计算数值）、魏畇钊（负责玩家人物和怪兽的动作，交互，金币血量更新）

美工设计：王文宇、魏畇钊

报告撰写：黄悦（一、介绍、二、基本架构、三中技术界面）、魏畇钊（三中核心算法）、王文宇（三中美工，发展历程）

## 发展历程：

初始概念和规划阶段：确定了将编程与冒险游戏相结合的核心概念。进行小范围调研，挖掘了编程需求和冒险游戏的经 典与情感，为游戏的整体规划和设计奠定基础。

软件选定：下载pycharm并import pygame;考虑到团队合作的便捷性，选择用github,学习相关知识；下载github,

GitKraken，注册账号并登录

美术设计和角色设定：着重于界面设计和角色设定，创造了游戏的背景世界和人物形象，旨在通过精心设计和细致绘画 为游戏创造引人入胜的视觉体验。选定背景音乐。

编程和关卡设计：进行算法设计、编程和关卡设计工作，实现游戏的功能和交互性。设计不同类型的关卡并变换难度， 增加游戏的挑战性和乐趣。初始化游戏窗口，进行主界面的搭建，商店、背包、故事框等的搭建。初始化人物形象，实 现商店购买物品，背包存放和使用物品的功能。丰富人物，优化人物动作，增加人物行为；初始化怪兽，增加人物攻击 功能。初始化更多类的怪兽，实现人物与怪兽交互。进行背景循环，进行关卡控制和设计。

测试和优化：在游戏开发的后期阶段，进行大量测试和优化工作，邀请志愿者玩家参与测试，收集反馈意见，并对游戏 进行调整和改进，确保流畅性、平衡性和可玩性。

游戏开发是一个复杂而迭代的过程，通过不断规划、设计、测试和更新，《代码之旅》逐步完善，为玩家提供优质游戏体 验。

**心得**

# 黄悦

制作这个游戏是一次非常有趣和有收获的经历。通过深入参与代码阅读和分析，我对游戏开发中的一些关键概念和技术有了 更深的理解。

在团队协作方面，GitKraken的使用为团队提供了高效的代码管理和版本控制。每个成员在协同工作中能够专注于自己 的任务，通过GitKraken轻松同步代码。但是在探索和学习github和GitKraken的过程中，其实也遇到一些困难，比如 最开始同时接触vscode,github,码云以及命令行控制，我会把他们混在一起，当时弄不清楚他们之间应该是怎样的协作 关系，以及记不住那些命令。经过一番摸索和小组队员的交流后，我们发现了更适合新手用的GitKraken。同时明白了 在github上创建仓库，用GitKraken管理源码，像pull,push以及merge,同时命令行其实和GitKraken管理源码功能类 似，只是GitKraken管理源码更加直观方便。首先，我发现了良好的代码组织和模块化设计是保持代码清晰和易维护的 关键。这个游戏中，不同功能和部分都被封装成了独立的函数，使得整体代码结构清晰可读。这样的设计让团队能够更 容易地理解代码，也使得后续的修改和扩展变得更加方便。

在编写代码方面，我体会到了对于一个功能的实现，有时不能看表面的效果。比如商店的翻页，最初我苦想，该怎么实 现商店的页面的动态设置呢。最后恍然大悟，其实箭头开关的作用与其说是控制不如说是对应，点击左箭头，看似是向 左划了，其实是左箭头对应第一张商店图片，右箭头也同理。在实现很多效果的过程中都有这种体会。这对不爱打游戏 的我来说，是一中很奇妙的体验，有点看穿游戏背后的秘密的初体验。

总的来说，这次游戏制作是一次很好的学习和实践机会。通过参与其中，我不仅学到了更多关于游戏开发的知识，还深刻体 会到了团队协作和代码规范对项目成功的重要性。希望能够继续深入学习游戏开发，参与更多有趣的项目。

# 魏畇钊

首先，我发现了良好的代码组织和模块化设计是保持代码清晰和易维护的关键。这个游戏中，不同功能和部分都被封装成了 独立的函数，使得整体代码结构清晰可读。这样的设计让团队能够更容易地理解代码，也使得后续的修改和扩展变得更加方 便。

其次，多关卡设计为游戏的深度和可玩性提供了极大的提升。每个关卡都有独特的怪物和背景，给玩家带来不同的挑战和体 验。这种多样性使得游戏更富有趣味性，玩家可以不断地发现新的内容。

另外，游戏中的图形界面和交互设计也是吸引玩家的关键因素。Pygame库提供了便捷的图形渲染功能，使得游戏元素的表 现更加生动。而商店、背包等交互元素的设计增加了玩家与游戏的互动性，使得游戏更具吸引力。

# 王文宇

在美工方面，游戏团队着重于界面设计和角色设定，创造了游戏的背景世界、角色形象，希望通过精心的设计和细腻的绘画 能为游戏创造了一个引人入胜的视觉体验。游戏的美工设计不是一蹴而就的，而是一个复杂而迭代的过程，通过不断地规 划、设计、测试和更新，希望The Code Odyssey得以逐步完善，并提供给玩家们一个优质的游戏体验。

## 不足与反思

**视觉优化：** 游戏画面可能有提升的空间，更精美的视觉效果可以让玩家更沉浸于游戏世界中。考虑增加更多的细节、特效或动画来提升游戏的视觉吸引力。

例如，主城区中商店，背包，故事框等收回时需再次点击框，这有一定的不便捷性

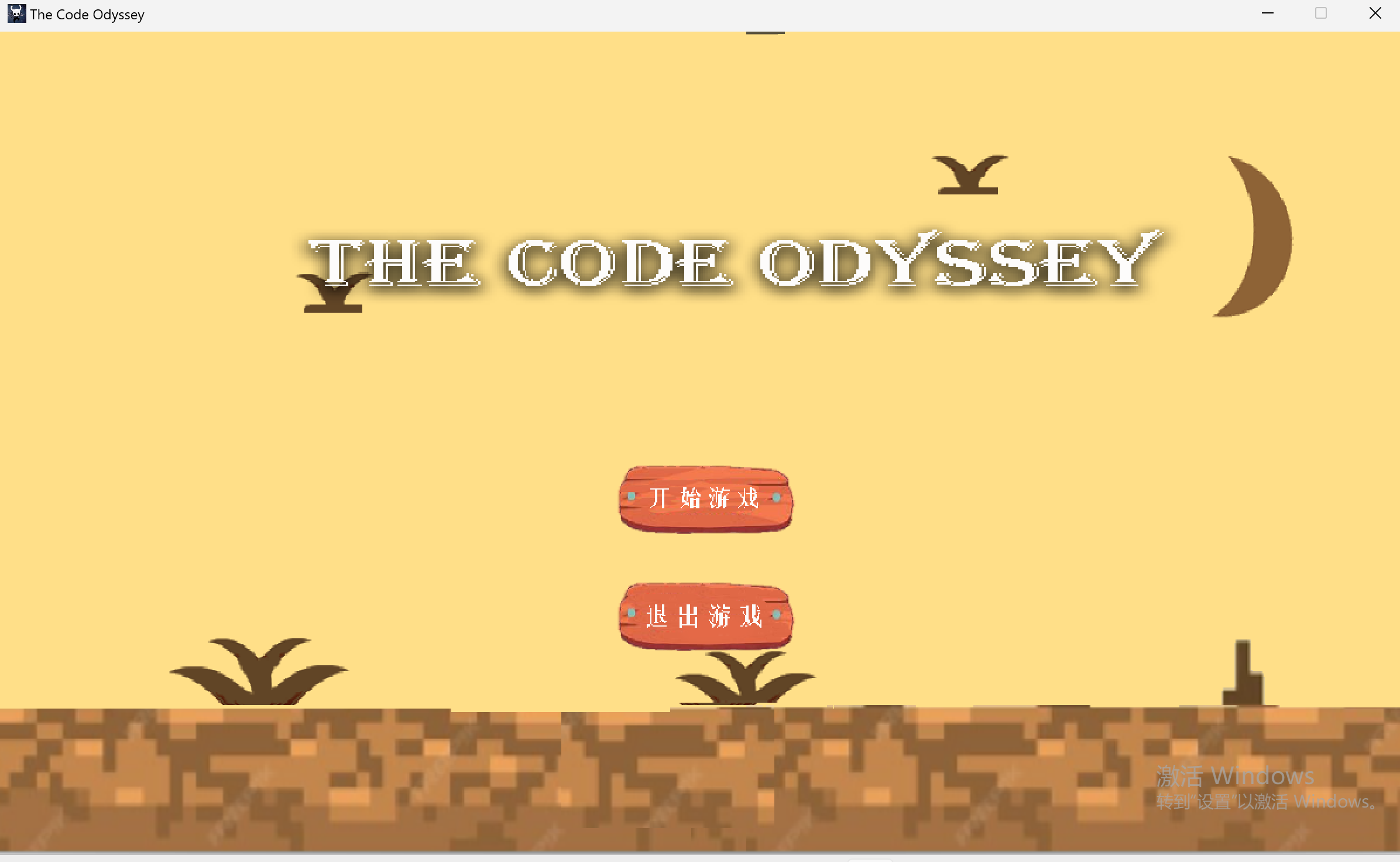
**关卡限制：** 关卡数量过于局限，可能不能满足玩家的游戏体验，缺乏一些障碍物和任务等。 例如，可以添加更多的关卡、调整关卡之间的逻辑关系、或者根据玩家的成绩来解锁新的关卡等。

**功能多样性限制：** 目前只有一个攻击动作，而且攻击对怪物的影响也比较简单。当前游戏的功能也较为局限

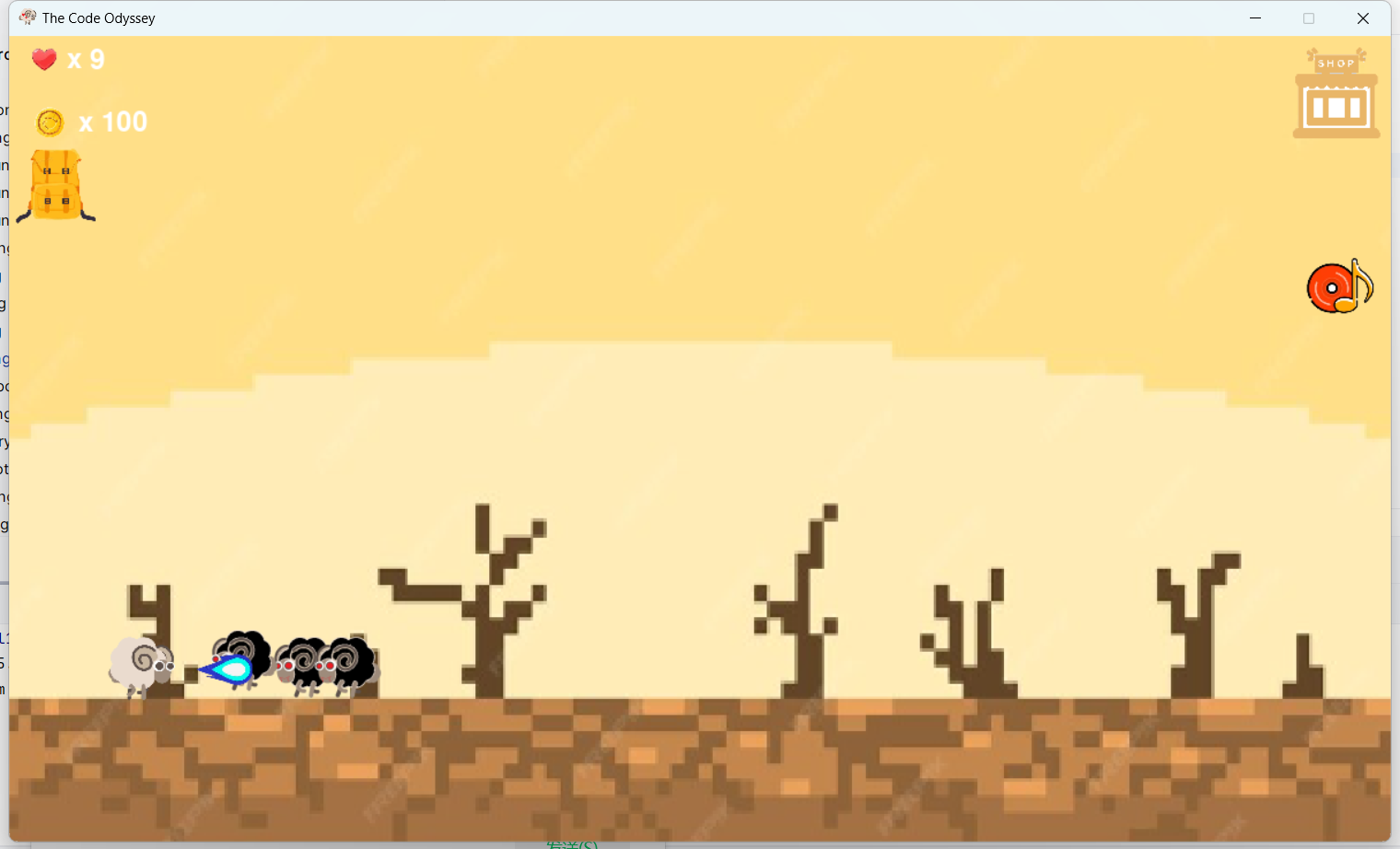
**代码结构和可读性：** 代码中的一些部分比较冗长，可能会影响可读性。 考虑将代码模块化，将相关的功能封装成函数或类，以提高代码的组织性和可维护性。

一些图示

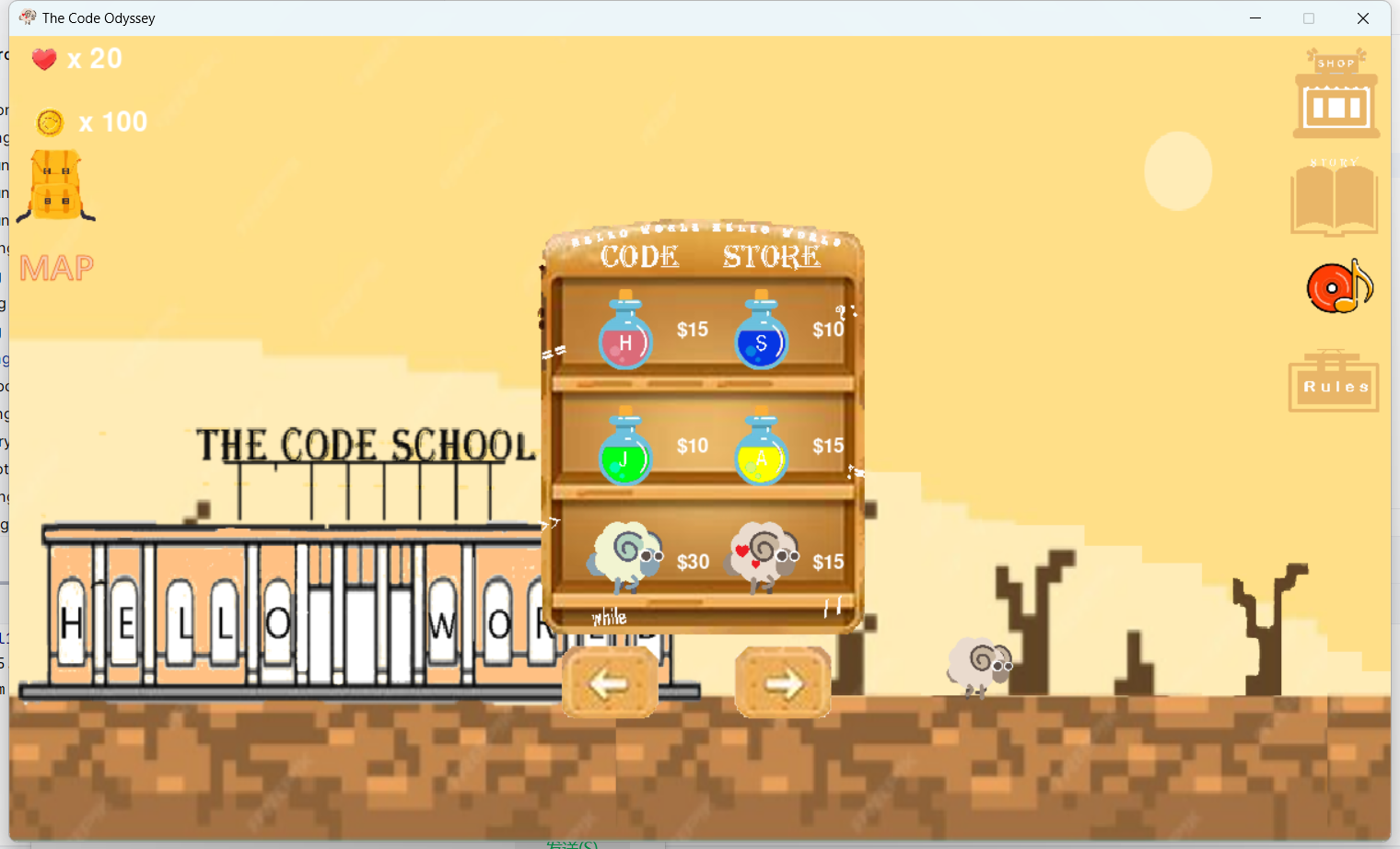
（开始菜单）



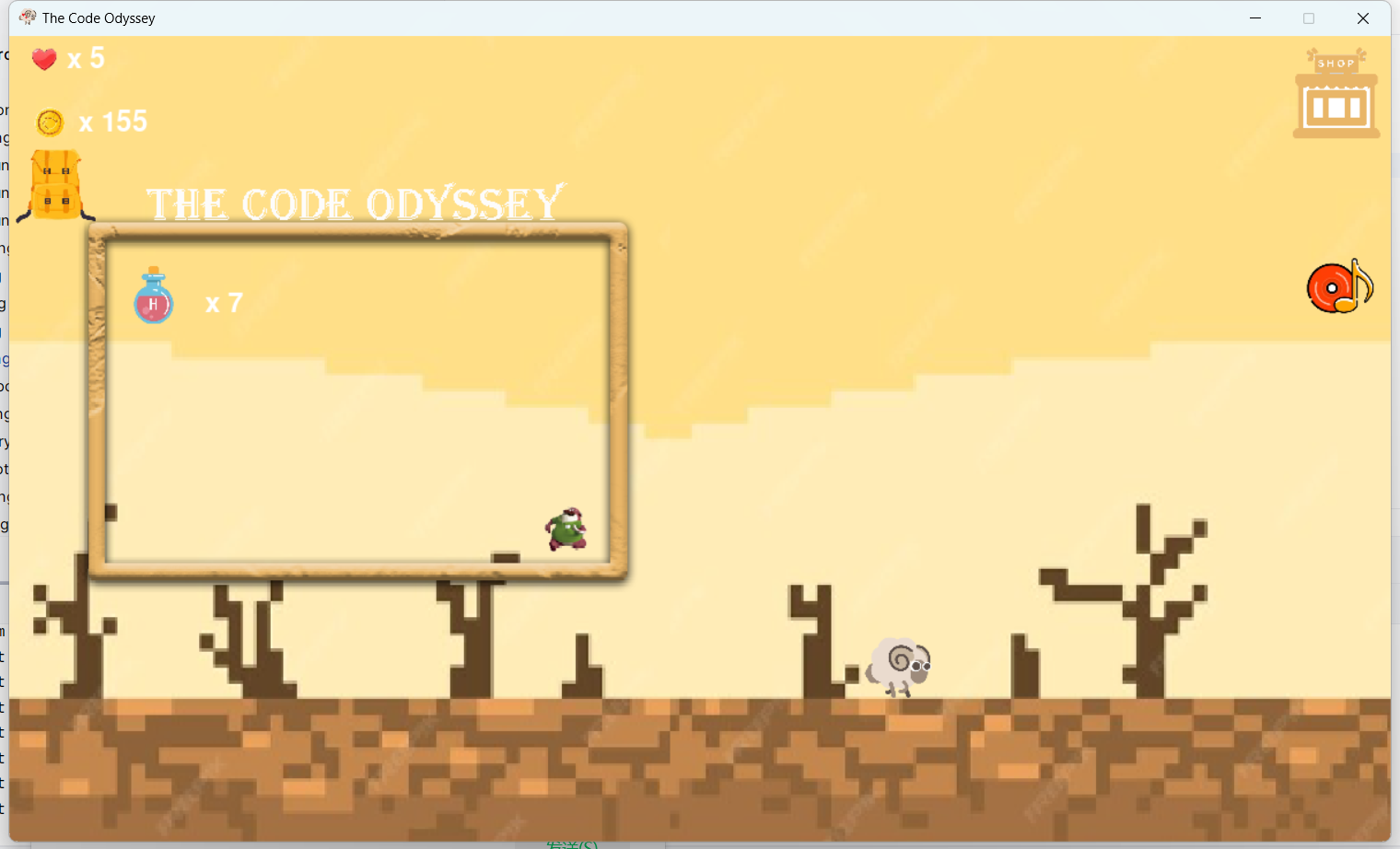
（攻击）

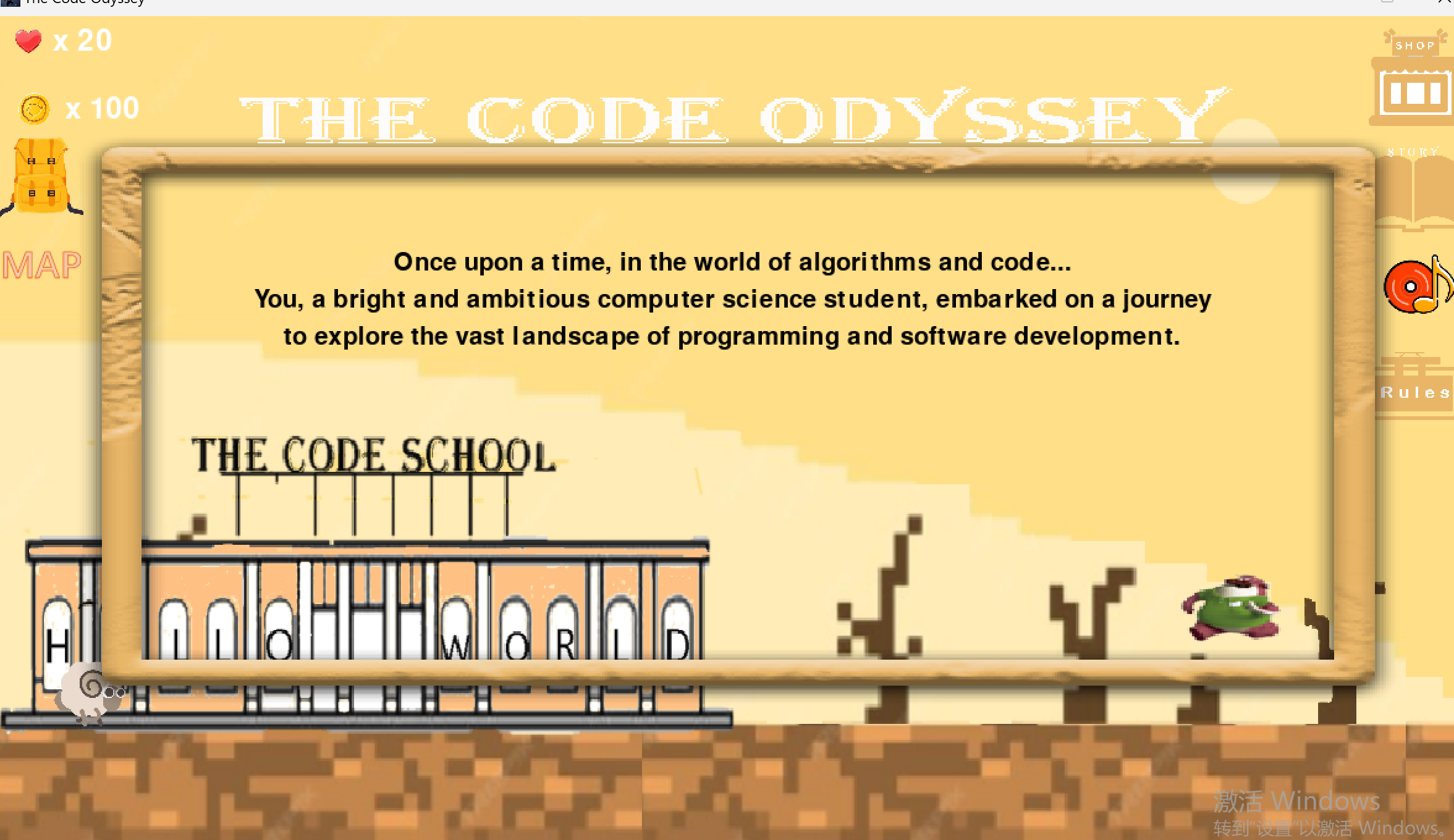


（主城区界面）

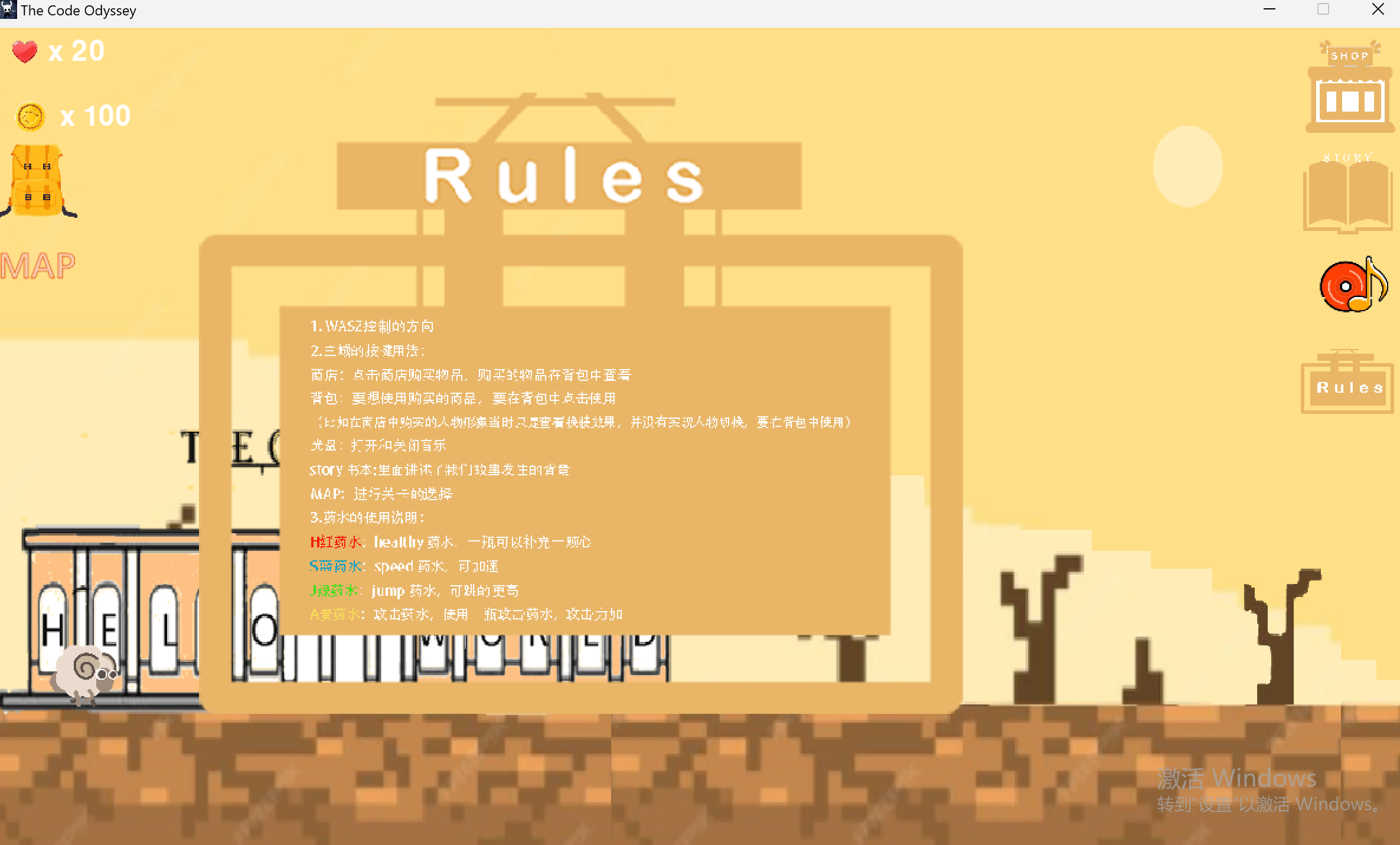


（背包）

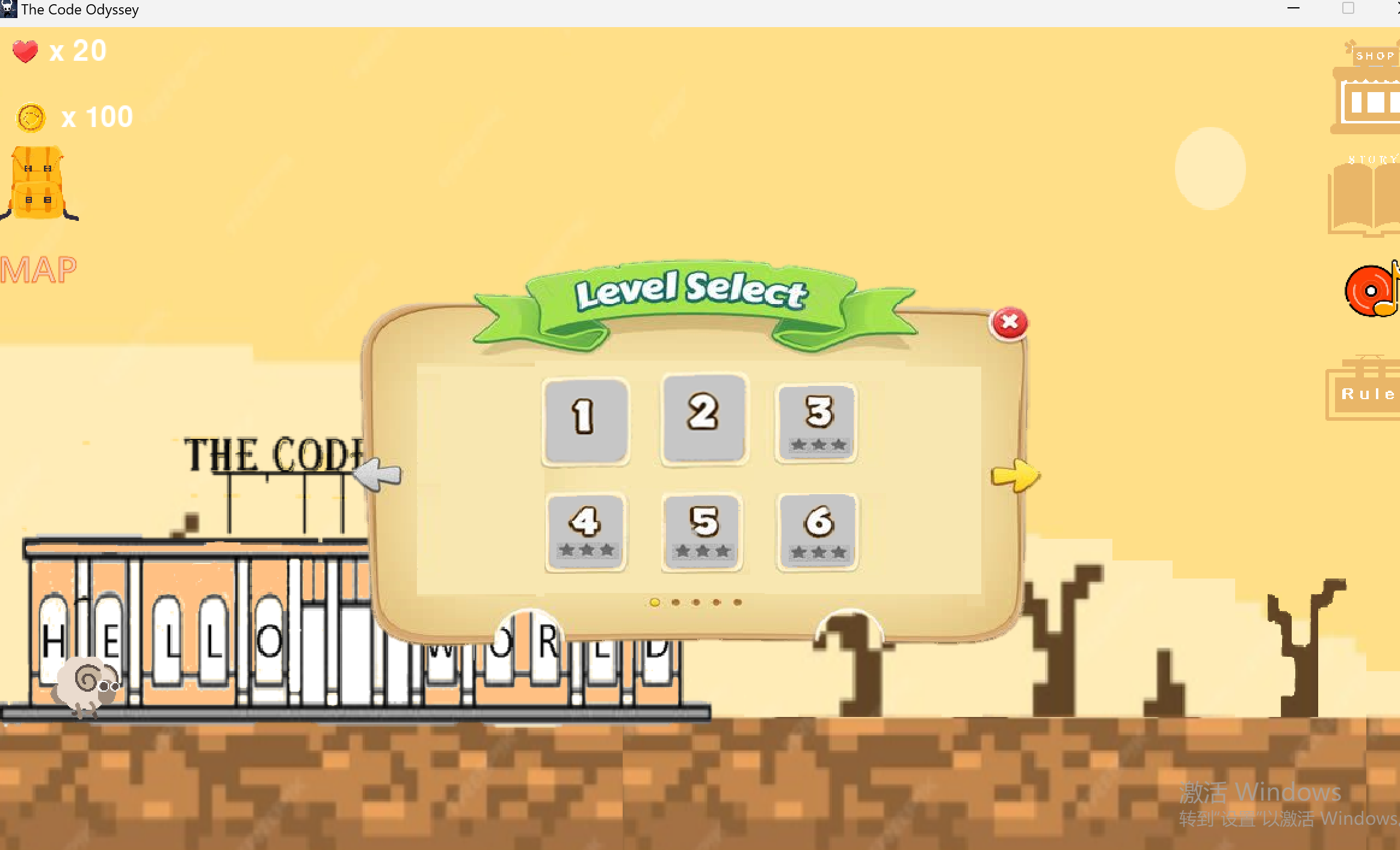


（故事框按钮）

（规则框按钮）



（关卡选择框）



（第二关界面）

