【开发者说】重塑经典，如何在 HarmonyOS 手机上还原贪吃蛇游戏

本文作者：

Kiruba Pradeep, Tech Lead, currentlybased at L&T Technology Services

本期我们为大家带来的是由印度开发者 Kiruba Pradeep 投稿的童年经典游戏，通过 JS 模板在 HarmonyOS 手机上一步步开发呈现，简单有趣的小游戏 demo，希望给你的 HarmonyOS 开发之旅多一点启发。

贪吃蛇，一款经典的小游戏，勾起了多少人的童年回忆。小时候，我们在游戏中快乐玩耍，现如今，让我们在开发中慢慢回溯吧！基于此，Kiruba Pradeep 总结了一套开发贪吃蛇游戏的要点秘诀，并在一行行代码中重塑经典。

在游戏中，玩家须通过导航按钮控制蛇的运动，以便吃掉随机出现的食物，来获得相应积分并增加蛇身长度，当蛇发生碰撞或反向移动时，则游戏结束。首先，让我们一起梳理一下开发的要点：

1) 绘制固定组件：包括画布、导航按钮、得分栏等内容的绘制

2) 设计动态组件：包括蛇体与食物的形状、大小、颜色的设计

3) 动态规则：设定食物随机出现的位置，蛇的运动方式及形态

4) 运行机制：当蛇消耗食物后，增加积分和蛇的长度，并绘制新的食物

5) 违规判定：当蛇碰撞到自身或画布边缘，或者反向移动，则判定违规，提示用户“game

over”，并自动重新开始游戏

在正式开始敲代码之前，开发者们需要先下载安装 Huawei DevEco Studio，如果对于这个流程不甚熟悉，可以参照官网的教程操作。

· HuaweiDevEco Studio 安装指南：

https://developer.harmonyos.com/cn/docs/documentation/doc-guides/software_install-0000001053582415

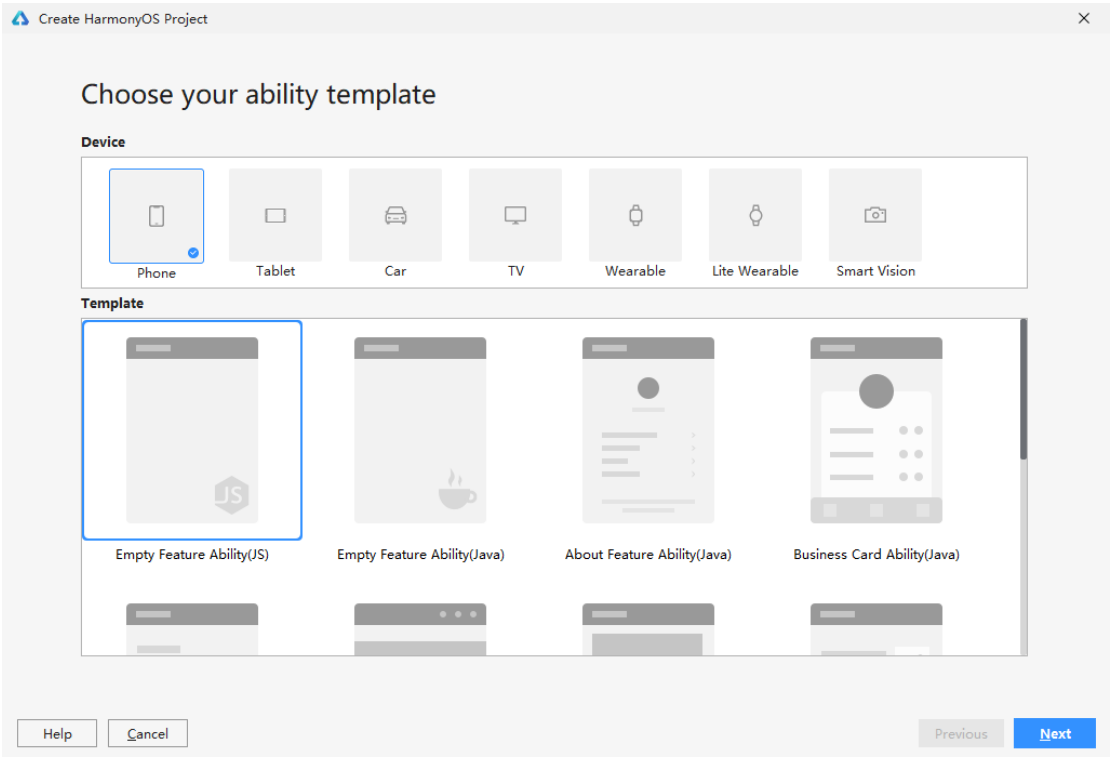
请注意，本次 demo 主要针对逻辑梳理和要点讲解，限于篇幅长度不会展示完整代码，开发者们可打开以下链接获取完整代码哦~

https://gitee.com/harmonyos-developer/kaifazheshuo

下面我们就进入项目开发环节，逐一攻破以上要点，一起探寻贪吃蛇之旅：

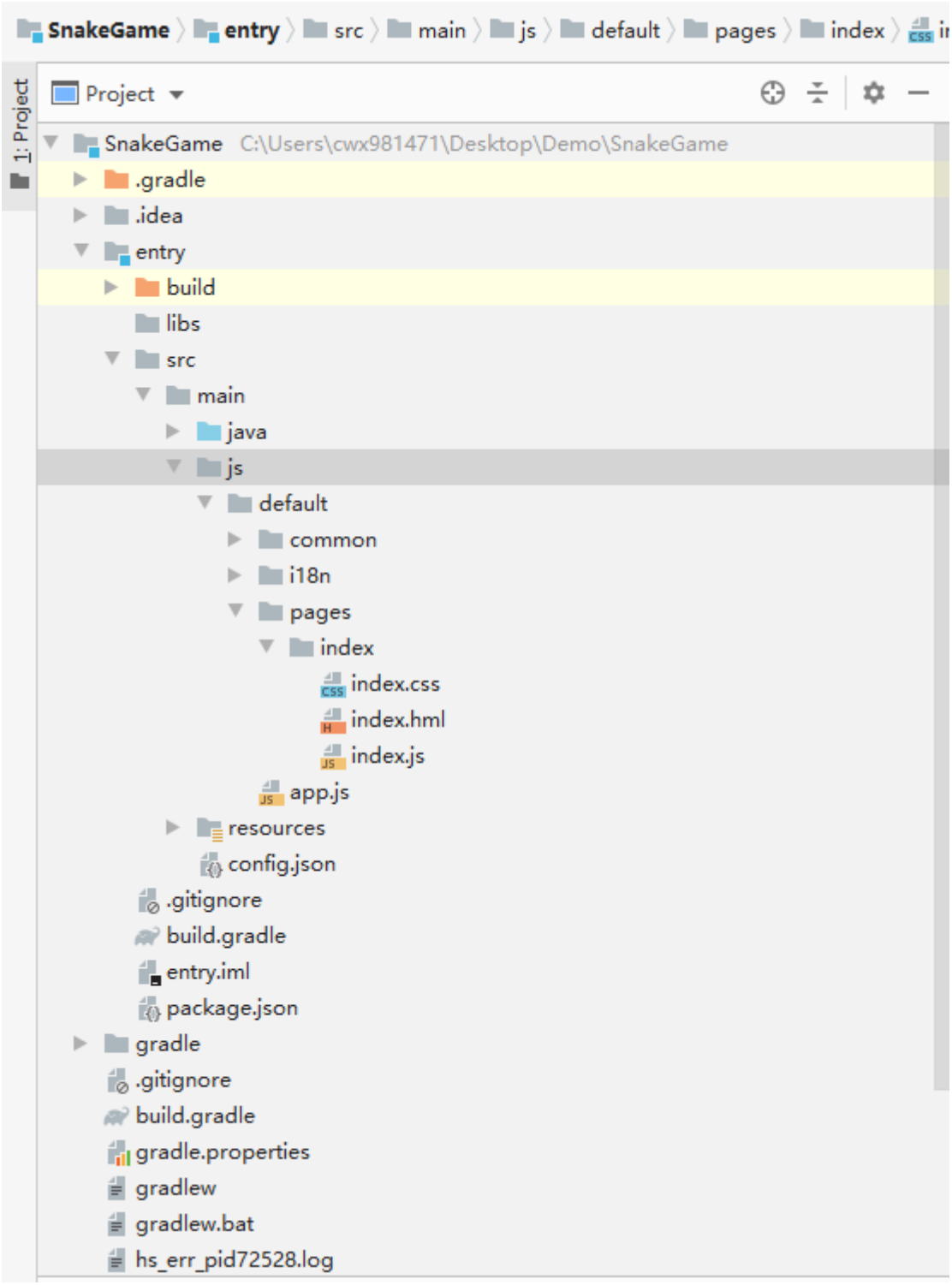
创建项目

打开 DevEco Studio，创建新项目（New Project），选择 Phone 设备，点击 Empty Feature Ability (JS)模板，注意项目命名须不带中文或特殊字符（如此处的 MyPhoneGame），最后点击 Finish。



项目创建后，我们先了解一下目录结构：

来源：HarmonyOS 开发者微信号 https://mp.weixin.qq.com/s/v4Qy_42bCxE4rp1uaSXsyA



- index.hml 文件：描述页面布局
- index.css 文件：描述页面样式
- index.js 文件：处理页面和用户之间的交互
- app.js 文件：管理全局 JavaScript 逻辑和应用程序生命周期

来源：HarmonyOS 开发者微信号 https://mp.weixin.qq.com/s/v4Qy_42bCxE4rp1uaSXsyA

- pages 目录：存储所有组件页面
- java 目录：存放与项目相关的 java 文件

其中，app.js 文件、pages 目录和 java 目录内容均由 JS 模板自动提供。

1）绘制固定组件

① 绘制画布，在 index.hml 文件中，设定画布的颜色和尺寸；

index.hml

```
<div class="container" onswipe="touchMove">
    <text class="title">Snake Game</text>
    <canvas ref="canvasref" style="width: 600px; height: 600px;
background-color:#000000" >
</canvas>
</div>
```

在 index.css 文件中，可以描述画布的呈现样式；

index.css

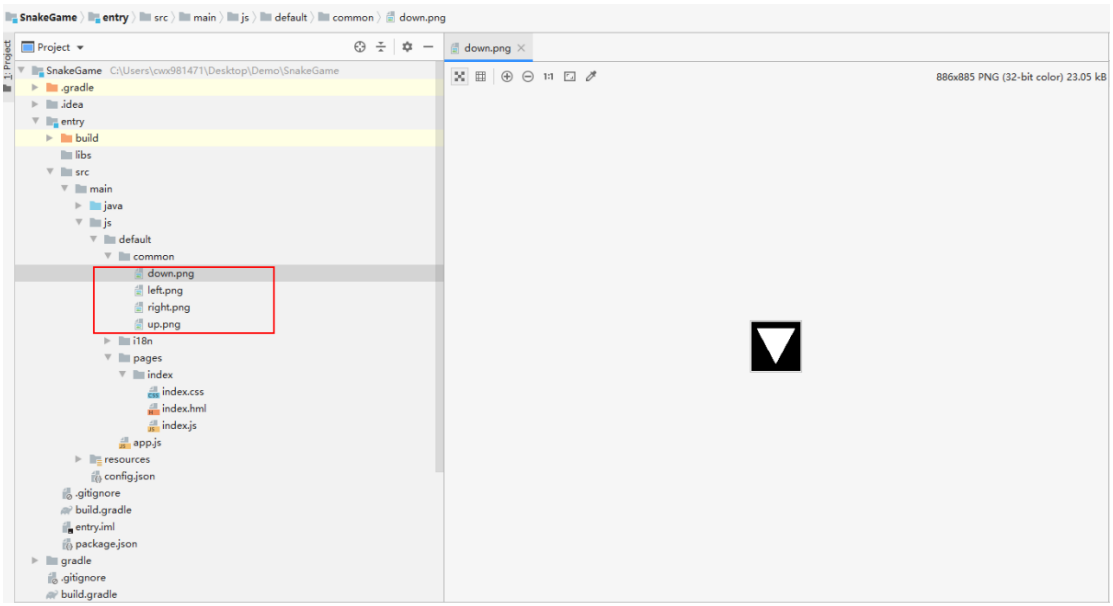
```
.container {
    flex-direction: column;
    justify-content: center;
    align-items: center;
    background-color: #d6d8d8;
}
```

在 index.js 文件中，引用画布$refs.canvasref，保障画布被蛇和食物图像填充。

index.js

```
const canvas = this.$refs.canvasref;
this.ctx = canvas.getContext("2d");
```

② 绘制导航按钮，我们先在 entry/src/main/js/default/common 目录下，添加 png 格式的导航按钮图片。



来源：HarmonyOS 开发者微信号 https://mp.weixin.qq.com/s/v4Qy_42bCxE4rp1uaSXsyA

然后，在 index.hml 文件中，创建“上下左右”四个导航按钮，并将方向保存在变量“this.direction”中；

index.hml

```
<image src='/common/up.png' class="backBtnup" onclick="onStartGame(1)"></image>
<div class="directsecond">
    <image src='/common/left.png' class="backBtnleft" onclick="onStartGame(2)"></image>.
    <image src='/common/down.png' class="backBtncenter" onclick="onStartGame(3)"></image>
    <image src='/common/right.png' class="backBtnright" onclick="onStartGame(4)"></image>
</div>
```

在 index.css 文件中，设置导航按钮大小、位置、边框和颜色等参数；

Index.css

```
.backBtnup, .backBtncenter, .backBtnleft, .backBtnright {
    width: 100px;
    height: 100px;
```

```
    margin-bottom: 20px;
    margin-top: 20px;
    border-radius: 10px;
    background-color: #000000;
}

.directsecond {
    flex-direction: row;
    justify-content: center;
    align-items: center;
}

.backBtnup {
    margin-top: 80px;
}
.backBtncenter {
    margin-left: 40px;
    margin-right: 40px;
```

在 index.js 文件中，通过 if 判断，调用“this.direction”，实现导航按钮控制蛇的运动方向。

index.js

```
onStartGame(direct){
    if (direct == 1) {
        this.direction = 'up'
    } else if (direct == 2) {
        this.direction = 'left'
    } else if (direct == 3) {
        this.direction = 'down'
    } else if (direct == 4) {
        this.direction = 'right'
    }
},
```

③ 在 index.hml 文件中，创建“得分”名称；在 index.css 文件中，设置“得分栏”大小和顶部边距；在 index.js 文件中，设定原始得分为“0”。

index.hml

```
<text  if="{{!gameOver}}"class="scoretitle">Score: {{score}}</text>
```

Index.css

```
.scoretitle {
    font-size: 50px;
    margin-top: 30px;
}
```

index.js

```
data: {
    title: "",
    snakeSize: 10,
    w: 600,
    h: 600,
    score: 0,
    snake : [],
    ctx: null,
    food: null,
    direction: '',
    gameOver: false,
    tail: {
        x: 0,
        y: 0
    },
```

2）设计动态组件

接下来，我们在 index.js 文件中创建蛇体，设定蛇的初始位置（0,0），蛇的长度为 10。

index.js

```
drawSnake() {
    var len = 7;
    var snake = [];
    for (var i = len - 1; i >= 0; i--) {
        snake.push({
            x: 0,
            y: i
        });
    }
    this.snake = snake;
},
```

在 index.js 文件中，为蛇绘制食物，设定食物的颜色和风格。

index.js

```
cookie(x, y) {
    var ctx= this.ctx;
    ctx.fillStyle = '#e28743';
    ctx.fillRect(x * this.snakeSize, y * this.snakeSize, this.snakeSize, this.snakeSize);
    //border color of the cookie
    ctx.fillStyle = '#e28743';
    ctx.fillRect(x * this.snakeSize + 1, y * this.snakeSize + 1, this.snakeSize - 2,
this.snakeSize - 2);
    this.ctx = ctx;
}
```

3）动态规则

在 index.js 文件中，我们通过随机函数 Math.random()，设定食物位置的随机值，同时须加 if 判断，以确保食物不会出现在蛇的身体上。

index.js

```
createFood() {
```

```
    this.food = {
        x: Math.floor((Math.random() * 30) + 1),
        y: Math.floor((Math.random() * 30) + 1)
    }
    for (var i = 0; i > this.snake.length; i++) {
        var snakeX = this.snake[i].x;
        var snakeY = this.snake[i].y;

        if (this.food.x === snakeX && this.food.y === snakeY || this.food.y ===
snakeY && this.food.x === snakeX) {
            this.food.x = Math.floor((Math.random() * 30) + 1);
            this.food.y = Math.floor((Math.random() * 30) + 1);
        }
    }
}
```

在 index.js 文件中，我们可以设定蛇自动跑，每跑一步的时间间隔为“500”。

index.js

```
var ctx = this.ctx;
ctx.fillStyle = '#e28743';
ctx.fillRect(x * this.snakeSize, y * this.snakeSize, this.snakeSize, this.snakeSize);
```

```
//border color of snake
ctx.strokeStyle = '#063970';
ctx.strokeRect(x * this.snakeSize, y * this.snakeSize, this.snakeSize, this.snakeSize);
this.ctx = ctx;
```

在 index.js 文件中，判断按键导航的方向，并沿此方向增加蛇头，减少蛇尾。

index.js

```
if (this.direction == 'right') {
    snakeX++;
}
else if (this.direction == 'left') {
    snakeX--;
}
else if (this.direction == 'up') {
    snakeY--;
} else if (this.direction == 'down') {
    snakeY++;
}
```

4）运行机制

在 index.js 文件中，我们设定每当蛇消耗食物后，“得分栏”就增加 5 分；

index.js

```
if(snakeX == this.food.x && snakeY== this.food.y) {
        this.tail = {x: snakeX, y: snakeY}; //Create a new head instead of moving thetail
      this.score = this.score+5;

      this.createFood(); //Create new food
} else {
      this.tail = this.snake.pop(); //pops out the last cell
      this.tail.x = snakeX;
      this.tail.y = snakeY;
}
```

与此同时，增加蛇的长度，并绘制新的食物。

index.js

```
this.snake.unshift(this.tail); //putsback the tail as the first cell
for(var i = 0; i < this.snake.length; i++) {
      this.bodySnake(this.snake[i].x, this.snake[i].y);
```

```
}

this.cookie(this.food.x, this.food.y);
```

5）违规判定

我们设定假如蛇碰撞自身或画布边缘，或者反向移动，则判定违规，提示用户“game over”，并自动重新开始游戏。

在 index.js 文件中，检查蛇是否碰撞自身或画布边缘：

index.js

```
checkCollision(x, y, array) {
    for(var i = 0; i < array.length; i++) {
        if(array[i].x === x && array[i].y === y)
        return true;
    }
    return false;
}
```

在 index.js 文件中，检查蛇是否反向移动：

index.js

```
if (snakeX == -1 || snakeX == this.w / this.snakeSize ||snakeY == -1 || snakeY ==this.h / this.snakeSize ||this.checkCollision(snakeX, snakeY, this.snake)) {
    ctx.clearRect(0,0,this.w,this.h); //cleanup the canvas
    this.restart()
    return;
}
```

在 index.hml 文件中，设定当游戏结束时，得分栏显示“game over”；在 index.js 文件中，设定重启游戏，重新绘制区域、蛇和食物，得分清零：

index.hml

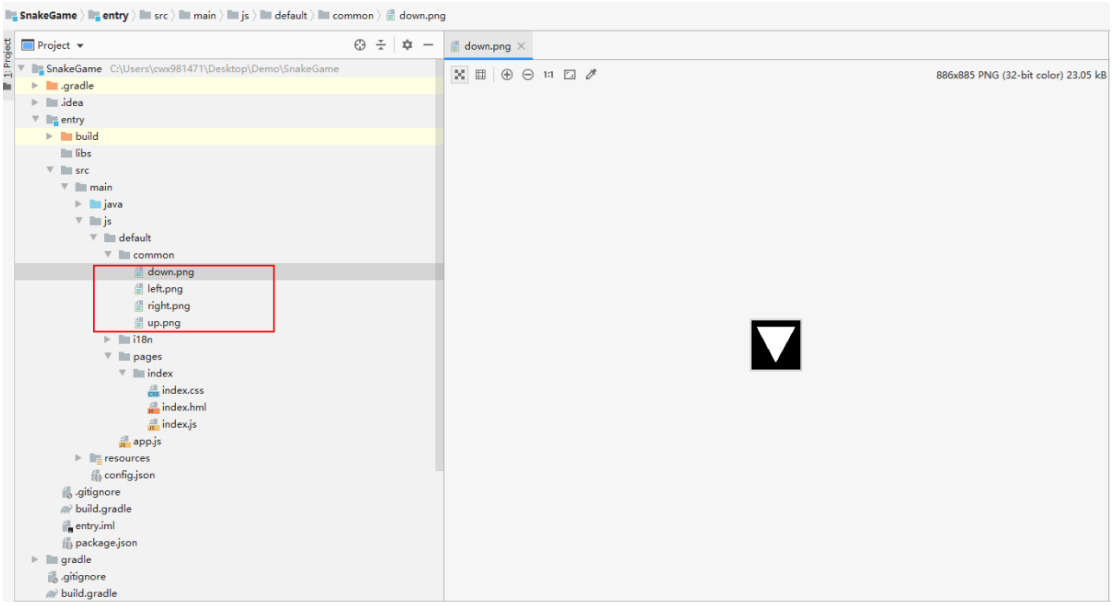
```
<text if="{{gameOver}}" class="scoretitle">Game Over!!</text>
```

index.js

```
restart() {
    this.drawArea()
    this.drawSnake()
    this.createFood()
    this.gameOver = true
    this.score = 0
```

},

到此，我们已经完成了贪吃蛇游戏的开发啦！



最终实现的效果，奥利给

回顾整个开发过程，难点在于当蛇消耗食物后，增加得分及蛇的长度，同时随机出现新的食物，如何判定违规并提示等方面，当我们一步步实操起来，所有难题都会迎刃而解。

来源：HarmonyOS 开发者微信号 https://mp.weixin.qq.com/s/v4Qy_42bCxE4rp1uaSXsyA