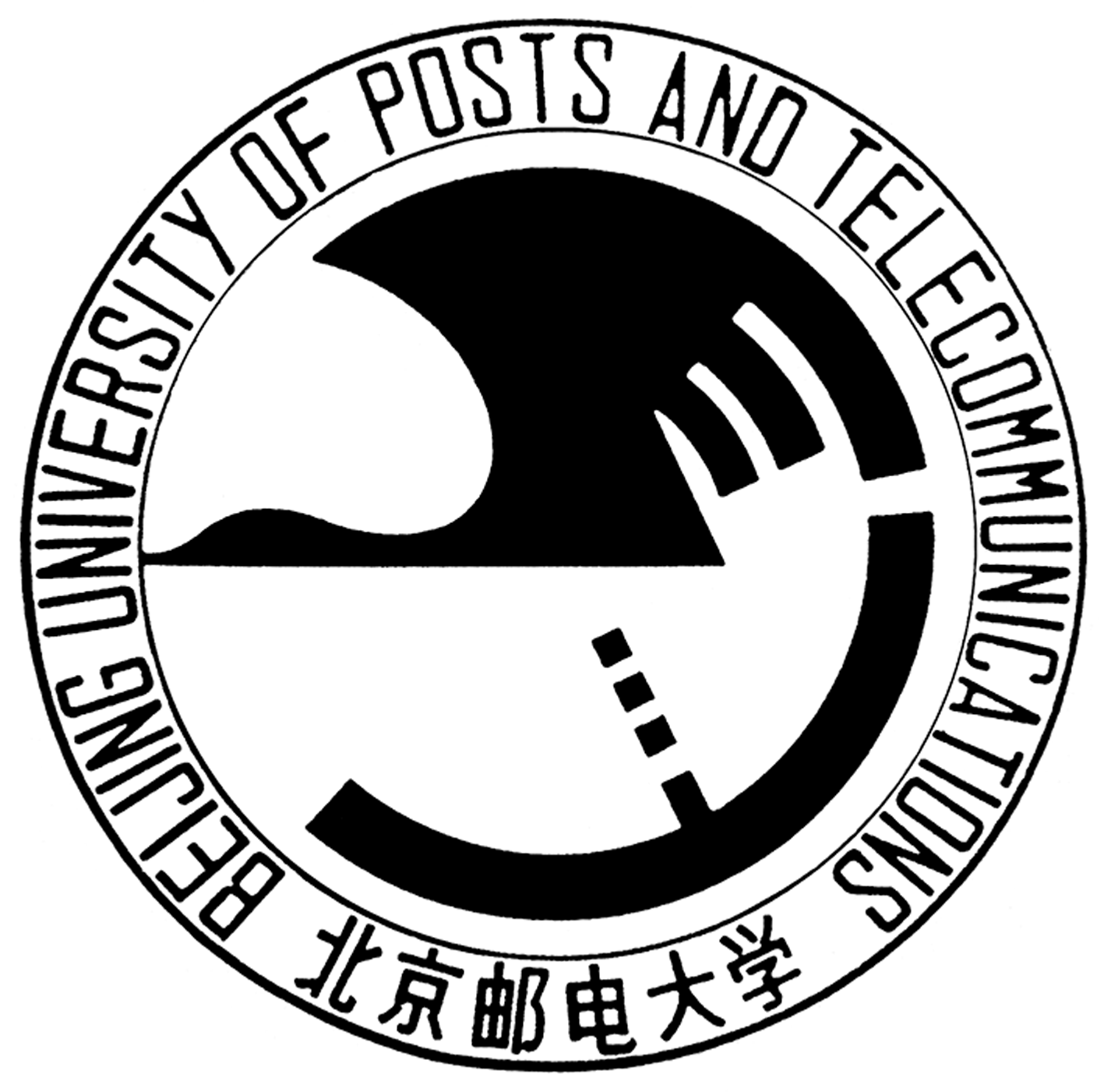
# 移动应用开发实验报告



|  |  |
| --- | --- |
| 实验名称： | 移动应用开发 |
| 姓 名： | 计子毅 |
| 学 号： | 2023212872 |
| 学 院(系)： | 计算机院 |
| 专 业： | 计算机科学与技术 |
| 指导教师： | 卢向群 |

2024年 12 月 4 日

# 实验一 快速入门-摇一摇登录校园网

# 实验目的

1. 了解App Inventor 2开发环境
2. 理解App Inventor 2开发的基本特点
3. 熟悉App Inventor 2开发过程
4. 掌握创建项目、设计界面、编写代码块、测试调试、打包应用等基本开发流程和操作
5. 了解App Inventor 2中基本组件（控制、逻辑、文本、变量、过程、事件处理程序、按钮等用户界面组件、界面布局、多媒体音效、加速度传感器等）的用法。

# 实验环境

1. 硬件环境

PC微机

1. 软件环境

Windows操作系统

App Inventor2018离线版（AppInventor2018PersonalEdition\_Win）

桌面版AI伴侣

1. 如果使用Android手机进行测试，还需要Android手机和安装在手机上的MIT AI2 Companion App

# 实验要求

1. 创建一个登录北邮校园网的使用App。
2. 基本功能：支持登录、注销、显示登录状态3个基本操作。
3. 实现摇一摇手机自动登录
4. 登录和注销时有提示音效

预期的运行界面如下图所示：



# 实验原理

首先，我们要理解基本的浏览网页的原理，如下图所示：

图示

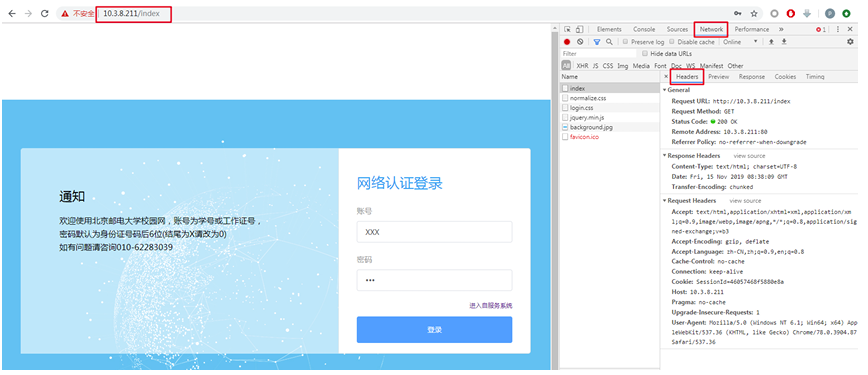
描述已自动生成

在网页登录时，注意到认证页面（http://10.3.8.211/login），如果尚未登录，输入用户名、密码，然后点击确定可以登录校园网；如果已经登录，会显示登陆成功的信息；如果已经登录成功，点击注销（http://10.3.8.211/logout），会注销成功，重新回到认证页面。

我们使用chrome 浏览器自带的网页开发工具来找出为了登录，我们需要发送给校园网网关的数据和应答的数据。

如何找到登录、注销时用的网址和发送数据的格式？

在认证页面（http://10.3.8.211/index）按下 F12，打开 chrome 开发者工具，选中开发者工具的network菜单-headers选项卡：



在页面输入用户名+密码，点击确定，会弹出以下界面，

图形用户界面, 应用程序, 网站

描述已自动生成

并且调试工具显示：Request URL: http://10.3.8.211/login

http://10.3.8.211/login 就是登录发送的网址。

在下部有Form data处点击view source，显示格式是user=XXX&pass=YYY。

如果输入正确的用户名+密码后，点击response菜单，可以看到登录成功时，网关返回的网页的源文件。

文本

低可信度描述已自动生成

通过这番操作，我们可以知道，登录时，可以用 http 的 Post 方法往网关http://10.3.8.211/login 提交用户名（user=用户名）和密码（pass=密码），然后从返回的网页中匹配某个特征字符串，就可以得知是否登录成功。例如，登录成功时，会返回“You have successfully logged in”，而失败则没有。

同理，通过用http 的 Get 方法请求网关 http://10.3.8.211/，从返回的网页中匹配某个特征字符串（successfully logged in），就可以得知登录的状态。

注销时，需要http的Get方法请求http://10.3.8.211/logout，然后浏览器会跳转到 http://10.3.8.211/index 页面，通过匹配此页面的特征字符串（如” 输入账号”） ，就可以得知注销的状态。

设计可理解为从现存状态到目标状态的一个行动方案。我们主要采用分解综合的方式进行分析，可以从目标从上往下分析，也可从现有的资源状态从下往上综合，一般是根据设计进度的反馈2种方法交替使用。为清晰起见，以下以思维导图的方式，从目标出发，逐步细化，一直到将目标分解为开发环境直接支持的功能或者我们可以直接把握的基本功能。之后我们需要做的，就是利用开发环境提供的这些基础功能，设计界面、编写代码进行组合，达到最终的目标（实验内容 ）。

下图是分析过程的示例：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

从以上分析可以看出，为做好实验和设计，对目标的准确理解和开发环境基本资源与功能的熟悉，是非常重要的。

所有实验的目的也是这2个方面：学会如何将目标分解到基本功能，学会如何使用工具与代码将基本功能组合成目标功能。

# 实验步骤

1. 打开开发环境

打开C:\CxcyCourse\AppInventor2018PersonalEdition\_Win\ 启动AppInventor.cmd，启动离线版的AppInventor2。

打开chrome浏览器，输入 localhost:8888

1. 新建项目

点击Poejects->Start new project，在弹出的对话框中输入项目名称，例如EXP1

1. 设计界面
2. 拖入 VerticalArrangement 组件： 用鼠标点 中 Layout 下的VerticalArrangement，拖动到右侧 Viewer 面板的 Screen1 中。在 Components 面板中能看到默认命名为 VerticalArrangement1 的组件在Screen1 下。

点击 viewer 面板的 VerticalArrangement1 的图标或者 Components 中的VerticalArrangement1，最右侧的 Properties 将显示点中的 VerticalArrangement1 的属性，需要修改部分属性以满足实验要求：

设置Properties-》AlignHorizontal 和 AlignVertical 都为 center，表示在水平和垂直方向都按照中心对齐。

设置Height 属性为80 percent。表示占纵向空间的80%，使得整个组件在屏幕的中上方。

设置Width属性为Fill parent。表示充满整个横向空间。

1. 按照同样的方法，拖入TableArrangement1 到 VerticalArrangement1 中，并设置属性为2行2列

图形用户界面

描述已自动生成

1. 拖入2个Label 标签组件（用户名、密码）：用鼠标点中User Interface下的Label，拖动到右侧viewer中TableArrangement1 中相应的位置。
2. 拖入1个TextBox文本框组件：用鼠标点中User Interface下的TextBox，拖动到右侧viewer中TableArrangement1 中相应的位置。
3. 拖入1个PasswordTextBox文本框组件：用鼠标点中User Interface下的PasswordTextBox，拖动到右侧viewer 中 TableArrangement1 中相应的位置。
4. 拖入3个Button按钮组件（登录、注销、显示状态）：用鼠标点中User Interface 下的 Button，拖动到右侧 viewer 中 TableArrangement1 中相应的位置。
5. 设置组件所显示的文本。在 Components 面板中选中上面拖入的组件，然后在右侧properties 的 Text 框，将各组件默认显示的文本改成上图所示文本
6. 重命名组件：在 Components 面板中选中上面拖入的组件，点击下部的Rename 按钮，在弹出的对话框中输入新的更具描述性的名字。
7. 拖入1个 Web（Web 客户端）框组件：用鼠标点中Connectivity通信连接下的Web，拖动到右侧 Viewer 中。所有的非可视组件都显示在界面的下方。
8. 拖入1个 Notifier（对话框）组件：用鼠标点中User Interface用户界面下的Notifier，拖动到右侧Viewer中。
9. 拖入1个Sound（音效播放器）组件：用鼠标点中Media多媒体下的Sound，拖动到右侧Viewer中。
10. 拖入1个AccelerometerSensor（加速度传感器）组件：用鼠标点中Sensor传感器下的AccelerometerSensor，拖动到右侧Viewer中。
11. 添加素材文件：登录成功与失败时手机将发出相应的声音提醒用户，首先需要上传声音的素材文件。 App 的图标和背景是图片素材，也需要上传， 点击界面右下角Media面板，选择Upload File，弹出上传文件对话框，选择“选择文件”，在文件选择框找到实验提供的素材： 音效文件：logon.wav、logoff.wav 图片素材：Exp1\_Logo.png、Exp\_bg1.jpg。点击OK，完成素材的上传。
12. 设置Screen属性：为Screen1 修改全局属性，包括：应用名称、主题样式、App图标等。 点击Components-》Screen1，在右侧 Properties 中修改成自己喜欢的值，提供参考如下： AppName：Exp1\_BuptLogin Theme：Device Default Title：北邮校园网登录 BackgroundImage: Exp\_bg1.jpg
13. 添加组件行为
14. 切换到编程界面
15. 需要先定义一个全局变量ActionType 用于区分请求类型：因为“登录、注销、显示状态”3 个按钮事件都会使用 web 客户端向网关发送请求，目的是为了在web客户端的应答中区分是哪种请求。
16. 创建一个过程 Login：因为加速度传感器检测到晃动和点击“登录”按钮的操作都是发送登录请求，为防止代码冗余，将这2个相同的操作提取为一个过程Login。基本过程是：定义一个请求登录网关的函数Login，将ActionType设置1，将web客户端的url网址设置为http://10.3.8.211/login，将 web 客户端的 Post 参数设置为之前分析所得的格式
17. 最后，在Button\_login的Click事件中调用Login过程。
18. 处理注销Buttom点击事件，注销操作是使用web客户端向http://10.3.8.211/logout 发送get 请求，无需输入账户。

图片包含 图形用户界面

描述已自动生成

1. 处理显示状态Button点击事件，显示登录状态操作是使用web客户端向http://10.3.8.211发送get请求，无需输入账户。



1. 处理Web客户端Web1的响应事件，“登录、注销、显示状态”3个按钮事件都会使用web客户端向网关发送请求，当网关返回应答数据时，web客户端Web1的GotText 事件将被调用。我们可在此事件中根据请求的类型做响应的处理。

日程表

描述已自动生成

1. 处理加速度传感器晃动事件，晃动手机功能在实验室提供的桌面版 AI 伴侣中无法直接测试，需要使用Android 手机，有条件的同学可以选择完成。 加速度传感器AccelerometerSensor1 晃动时，产生 Shaking 事件，在此事件中调用Login过程，则可以触发登录操作。

图片包含 图形用户界面

描述已自动生成

1. 测试软件
2. 打开C:\CxcyCourse\AI2Companion\_Win7\_64bit\ailaunch.bat，启动桌面AI 伴侣。
3. 将AI伴侣链接地址改为localhost:8888
4. 在开发环境中，选择Connect-》AI Companion，弹出二维码，将6位code输入框中，并点击用编码进行连接。

# 实验结果与分析

输入网关账号与密码后点击登录，显示 have logged in^\_^ 登录成功

图形用户界面, 网站

描述已自动生成

点击显示状态后显示logged in表示已经登录

日历

低可信度描述已自动生成

点击注销，显示注销成功

日历

低可信度描述已自动生成

注销后，点击显示状态，显示haven’t logged in表示没有登陆，即已经注销。

图形用户界面

低可信度描述已自动生成

# 实验总结

通过本次实验，我熟练掌握了App Inventor 2开发环境的使用，包括新建项目、设计界面、编写代码块、测试调试等流程。

通过chrome浏览器的开发者工具，我深入分析了登录、注销和显示状态时的网址和数据格式，确保在编写代码时能够正确地向网关发送请求并处理返回的数据。

为了提高代码的可读性和可维护性，我采用了模块化的设计思路，将相同功能的代码块封装成过程，避免了代码的冗余和重复。

在实验过程中，我注重测试和调试环节，及时发现并修复了代码中的错误，确保了应用的稳定性和可靠性。

通过本次实验，我深刻体会到了移动应用开发的重要性和挑战性。在开发过程中，我遇到了许多困难和挑战，但通过不断学习和实践，我逐渐掌握了开发流程和技能，并成功实现了实验目标。这次实验不仅提高了我的专业技能和实践能力，还培养了我的耐心和解决问题的能力。

# 实验二 Flappy Bird 1 — 扬翅的小鸟

# 实验目的

1．熟悉App Inventor 2 开发环境和开发过程

2．理解事件处理程序在App Inventor 2开发中的重要作用

3．熟悉App Inventor 2 的组件布局、画布、定时器、音效等基本功能的使用

4．熟悉列表等基本数据结构的使用

5．熟悉判断等逻辑结构的使用

6．理解精灵动画的实现原理

# 实验环境

1. 硬件环境

PC 微机

2.软件环境

Windows 操作系统

App Inventor2018 离线版（AppInventor2018PersonalEdition\_Win）

桌面版AI伴侣（AI2Companion\_Win7\_64bit）

3.如果使用Android 手机进行测试，还需要Android 手机和安装在手机上的

MIT AI2 Companion App。

# 实验要求

1． 在屏幕上创建游戏的背景。

2． 并在屏幕中间创建一只扇动翅膀的小鸟。

3． 同时小鸟发出扇动翅膀的声音。

实验要点有：

1． 打开App Inventor 开发环境；

2． 使用组件设计器创建应用的外观；

3． 使用块编辑器为组件设定行为（做什么以及何时做）；

4． 使用组件设计器选择组件（在Android设备上，有些组件可以显示，有

些则不可见）；

5． 从本地计算机加载媒体文件（声音和图像），并添加到应用中；

6． 用块编辑器来组装程序块，以此来设定组件行为；

7． 使用定时器实现动画效果；

8． 用 App Inventor 的实时测试功能对应用进行测试。可以一边创建应用，

一边在手机上或者模拟器上看到它们外观以及运行情况；

9． 将应用打包并下载到Android设备上。

预期的实验结果：

图表

描述已自动生成

# 实验原理

本实验使用的动画技术可称为精灵动画（Sprite Animation）。

精灵动画是一种电子版的序列帧动画，是以快速连续播放不同的静止画面，来产生角色的动态效果，就像电影胶片一般，通过快速切换来产生动画效果。

所谓精灵，就是一张张大小相同的位图。通常使用带有透明通道的图片格式保存，每一个区域都代表这个精灵在某一静止时刻的动作体现。

获取精灵动画的位图资源之后，在程序中只需将资源按单个动作图像大小进行切割，并存于数组中，在程序循环中根据时间的改变，顺序的循环显示不同的精灵动作，就可以实现出角色的动画效果。

对本实验而言，可以使用3张位图表示小鸟翅膀的上中下3个位置，再配合

以位置的上下，就可以模拟出扇动翅膀的动画效果。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

根据以上分析，我们将使用一个列表存储小鸟在上中下中4个状态的位图与位置，然后定时器超时时轮流切换这4个状态，这样每4个超时后会完成了翅膀的一次扇动。

# 实验步骤

* + - 1. 打开App Inventor 开发环境C:\CxcyCourse\AppInventor2018PersonalEdition\_Win\启动 AppInventor.cmd
      2. 新建项目
      3. 使用组件设计器创建界面

1. 上传素材，在右下角的Media-》Upload Files 上传以下文件。

表格

描述已自动生成

1. 添加画布并设置画布属性
2. ：在Canvas1中添加地基精灵（ImageSprite），重命名为ImageSprite\_base，并设置好属性
3. 在Canvas1中添加小鸟精灵（ImageSprite），重命名为ImageSprite\_bird，并设置好属性。
4. 添加一个定时器（Clock），并设置好属性。
5. 添加一个音效播放器（Sound），并设置好属性。
   * + 1. 使用块编辑器设定程序行为
6. 为此先创建一个全局列表birdFlapStates，该列表的每一项又是一个列表，分别存储小鸟的位图与Y坐标偏移。
7. 然后添加一个轮换列表状态的函数rotateBirdStates。该函数的原理是把列表的第一个元素添加到列表末尾，然后删除第一个元素。以后每次都取列表的第一个元素，这就模拟了一次切换。
8. 添加屏幕初始化事件的响应代码，完成初始化。
9. 添加定时器超时事件的响应代码，完成小鸟扇动翅膀的动画与声音。

图表, 日程表

描述已自动生成

# 实验结果与分析

图表

描述已自动生成

实现了小鸟翅膀上中下扇动并且在每一次扇动时发出音效，营造出小鸟悬浮在空中的场景。

# 实验总结

在实验过程中，我深刻体会到了事件处理程序在开发中的重要性，学会了如何合理地使用定时器、音效等组件来增强应用的交互性和趣味性。

通过不断地调试和优化，我学会了如何发现和解决程序中的错误，提高了自己的逻辑思维和问题解决能力。

# 实验三 Flappy Bird 2 — 下坠的小鸟

# 实验目的

1． 掌握App Inventor 2 开发过程

2． 了解App Inventor 2 调试程序的方法

3． 熟悉使用定时器实现和控制动画的方法

4． 检测并处理用户输入的点击事件

5． 碰撞检测

6． 多屏程序的设计

7． 使用过程（procedure）等技术组织稍微复杂的逻辑

# 实验环境

1.硬件环境

PC 微机

2.软件环境

Windows 操作系统

App Inventor2018 离线版（AppInventor2018PersonalEdition\_Win）

桌面版AI伴侣（AI2Companion\_Win7\_64bit）

3.如果使用Android 手机进行测试，还需要Android 手机和安装在手机上的

MIT AI2 Companion App。

# 实验要求

本实验将要在实验1的基础上增加以下功能：

1． 增加启动游戏的图像，当用户点击时，切换到游戏交互界面。

2． 在游戏交互界面，小鸟将以一定的加速度做自由落体运动。

3． 用户点击屏幕后，小鸟将获得一个向上的恒定速度。

4． 小鸟在上升下坠过程中，需要模拟抬头和低头动作。

5． 小鸟在碰撞到地基后，停止游戏并显示游戏结束消息和碰撞音效。

实验要点有：

1． 使用组件设计器添加多个屏幕，设计相应屏幕的外观；

2． 使用全局变量记录小鸟的状态，并根据用户输入修改小鸟状态；

3． 使用定时器实现小鸟扇动翅膀和抬头低头的动画效果；

4． 使用程序控制屏幕元素的布局，在屏幕之间传递参数；

5． 使用程序模拟物理世界行为的方法；

6． 响应精灵的碰撞检测事件。

# 实验原理

本实验在上一个实验的基础上增加功能。主要是小鸟除了扇动翅膀外，将获得一个向下的恒定加速度，需要用户点击屏幕，给予一个向上的初始速度，才能不因为碰到地基而导致游戏失败。

文本

描述已自动生成

本实验需要注意的是Y方向上加速度与初始速度对小鸟运动的影响。为此，可以增加一个全局变量记录小鸟的速度，因为超市间隔是一定的，所以位移的变化量就是速度的变化量，这样就可以用速度直接计算在每个超时间隔的 Y 方向位移值。

# 实验步骤

1. 使用组件设计器设计界面
2. 修改 Screen1 界面
3. 上传素材，在右下角的Media-》Upload Files 上传以下文件。

表格

描述已自动生成

1. 在Canvas1中添加精灵（ImageSprite），重命名为ImageSprite\_message，并设置好属性（主要是Picture源图像和XY坐标）。
2. 增加 Screen2 界面
3. 创建 Screen2。在开发环境上部点击 Add Screen，增加一块屏幕Screen2
4. 按照上一个实验的3.4.3 增加以下组件（这些组件在 Screen2 中与

Screen1 中的属性也都相同）：

Canvas1

ImageSprite\_base

ImageSprite\_bird

Clock1

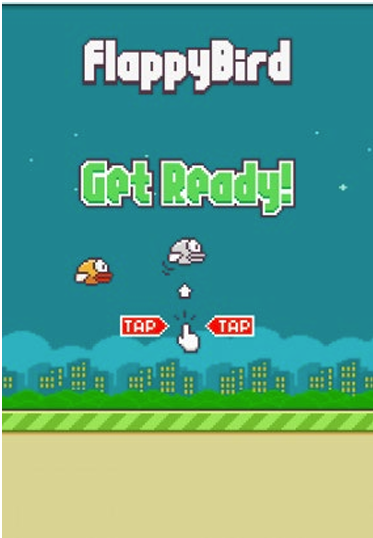
Sound1

1. 在Canvas1中添加精灵（ImageSprite），重命名为ImageSprite\_over，并设置好属性。
2. 使用块编辑器设定程序行为
3. 修改 Screen1 的程序行为

本实验的Screen1只在上一个实验基础上增加了ImageSprite\_message，只需要处理其touched事件，打开第二个屏幕即可。

1. 修改 Screen2 的程序行为
2. 还是先创建一个全局列表birdFlapStates，以及轮换列表状态的函数rotateBirdStates。
3. 添加Screen2初始化事件的响应代码，完成初始化。
4. 添加定时器超时事件的响应代码，完成小鸟扇动翅膀、改变位移、改变抬头角度的动画。
5. 添加屏幕点击事件的响应代码，完成赋予小鸟初始速度、抬头、发出翅膀声音的效果。
6. 添加碰撞检测事件的响应代码，显示游戏结束、改变小鸟位置与朝向、停止定时器、播放碰撞声音。

# 实验结果与分析



图表

中度可信度描述已自动生成

图形用户界面, 图表

描述已自动生成

# 实验总结

通过本次实验，我成功地实现了小鸟的自由落体运动、用户点击屏幕后小鸟上升的效果、小鸟扇动翅膀和抬头低头的动画效果以及碰撞检测功能。实验过程中，我遇到了一些挑战，如如何准确地计算小鸟的位移和速度变化，以及如何处理碰撞检测事件。但通过不断地调试和优化，我最终克服了这些困难，并实现了预期的实验结果

# 实验四 Flappy Bird 3 — 避障的小鸟

# 实验目的

1． 掌握使用画布、精灵、定时器实现动画的方法

2． 碰撞检测与边界检测

3． 随机数的使用

4． 使用过程（procedure）等技术组织复杂的逻辑

# 实验环境

1.硬件环境

PC 微机

2.软件环境

Windows 操作系统

App Inventor2018 离线版（AppInventor2018PersonalEdition\_Win）

桌面版AI伴侣（AI2Companion\_Win7\_64bit）

3.如果使用Android 手机进行测试，还需要Android 手机和安装在手机上的

MIT AI2 Companion App。

# 实验要求

本实验将要在实验2基础上增加以下功能：

1． 在屏幕2中随机生成上下2根柱子

2． 柱子以恒定的速率从右往左运动

3． 用户通过点击控制小鸟的速度从柱子之间穿过

4． 穿过中如果碰到柱子则游戏失败，停止游戏并显示游戏结束消息和碰撞音效。

实验要点有：

1． 使用随机数控制2根柱子的起始位置

2． 以柱子的从右往左移动模拟小鸟的从左往右飞

3． 使用全局变量记录游戏的状态，包括小鸟柱子等

4． 响应精灵的碰撞检测事件。

# 实验原理

本实验在上一个实验的基础上增加功能。主要是需要随机生成2根柱子，并控制小鸟从柱子穿过。采用的办法是让柱子从右往左运动，模拟小鸟的穿柱子操作。整体难度比上一个没增加多少，只是功能更丰富。

表格

描述已自动生成

# 实验步骤

1. 修改Screen2 界面, 在Screen2中增加2根柱子
2. 上传柱子的图像素材文件：pipe-green\_down.png和pipe-green\_up.png。
3. 增加 2 个精灵表示柱子，设置图像、是否显示等属性。ImageSprite\_pipe1up 和 ImageSprite\_pipe1down
4. 添加Screen2 组件程序行为
5. 修改屏幕2的初始化函数，计算随机生成的上下2根柱子的XY坐标和高度。具体实现时建议先将柱子的XY坐标和高度设置成一个固定的数值，在调试好后续的小鸟与柱子的碰撞检测后，再将XY坐标设置随机数。
6. 在定时器的超时事件处理函数中增加柱子从右往左移动的功能。 每次超时时，柱子的x坐标减去固定的数值模拟匀速运动
7. 在精灵的碰撞检测函数中增加柱子与小鸟碰撞的判断代码，如果发生碰撞，停止游戏，显示游戏结束信息

# 实验结果与分析

图形用户界面, 应用程序

描述已自动生成

点击手指所指位置，将从screen1跳转到screen2，即游戏开始

图片包含 游戏机, 房间

描述已自动生成

点击屏幕，小鸟会升高

图片包含 图形用户界面

描述已自动生成

当小鸟碰到柱子时，会头朝下坠落，并且显示game over的字样表示游戏失败

# 实验总结

实验结果表明，本次实验成功地在Flappy Bird游戏的基础上增加了避障功能。游戏运行流畅，柱子能够随机生成并以恒定的速率从右往左运动，小鸟能够通过点击屏幕升高并尝试从柱子之间穿过。当小鸟碰到柱子时，游戏会停止并显示“game over”字样，表示游戏失败。

通过本次实验，我深刻理解了使用画布、精灵、定时器实现动画的方法，掌握了碰撞检测与边界检测的实现原理，并学会了如何应用随机数来增加游戏的随机性和挑战性。同时，我也体会到了使用过程（procedure）等技术组织复杂逻辑的重要性。

# 实验五 Flappy Bird 4 — 飞扬的小鸟

# 实验目的

1． 综合实验1、2、3中使用过各种编程元素：的画布、精灵、定时器、音

效等组件，判断、循环等程序控制手段，列表、全局变量、随机数等数

据手段，初始化、定时器、碰撞检测等事件处理，实现一个较为完整的

游戏。

2． 使用变量、过程、和事件处理程序组织复杂的逻辑。

3． 熟悉程序状态的控制。

4． 了解如何保存数据到本地数据库或从本地数据库中读取数据。

# 实验环境

1.硬件环境

PC 微机

2.软件环境

Windows 操作系统

App Inventor2018 离线版（AppInventor2018PersonalEdition\_Win）

桌面版AI伴侣（AI2Companion\_Win7\_64bit）

3.如果使用Android 手机进行测试，还需要Android 手机和安装在手机上的

MIT AI2 Companion App。

# 实验要求

本实验将要在实验3基础上增加以下功能：

1． 在屏幕2中随机生成上下2根柱子，当柱子将到屏幕中央时，再随机生成上下3根柱子，之后屏幕上一般将有2组上下2根共4根柱子。到柱子碰到屏幕左侧（柱子“消失”）时，在屏幕右侧再次随机生成 2 根柱子，如此循环不止。

2． 柱子的XY位置、高度和间隔都可以设计为随机的，这样柱子出现的时机、位置和彼此间的间隔都带有随机性。或者只将柱子的Y坐标设置成随机的，这样程序更容易控制。

3． 增加计分功能，每次小鸟穿过一组柱子，就得一分。

4． 增加保存最高得分的功能，游戏结束时将最高得分从本地数据库读入，并与当前得分比对，将较大的结果存入本地数据库。

5． 穿过中如果碰到柱子则游戏失败，播放碰撞音效，小鸟继续自由下落，直到摔落到地基上，播放失败音效，停止游戏并显示游戏结束消息，结束消息中包括本地得分与最高得分，并提示用户再来一盘或者结束游戏。

6. 声音可开关，游戏可暂停按键

7. 三种游戏难度选择1、2、3

实验要点有：

1． 使用随机数、列表和全局变量通过4根柱子模拟无限柱子的功能

2． 小鸟位置、初始速度、加速度、柱子的随机位置、高度、间隔等对游戏难度的影响。

3． 游戏状态的控制。

4． 理解精灵动画游戏通过静态帧循环的方式模拟动画的原理。

5． 记录与保存得分与最高得分。

# 实验原理

本实验在上一个实验的基础上增加较多功能。主要是为了形成一个完整的游戏而提出的要求。

文本

中度可信度描述已自动生成

# 实验步骤

1. 修改Screen2 界面，在第二屏界面中再增加2根柱子
2. 上传柱子的图像素材文件
3. 增加2个精灵表示柱子，设置图像、是否显示等属性
4. 添加Screen2 组件程序行为
5. 修改屏幕2的初始化函数，计算随机生成的2组上下4根柱子的XY坐标和高度。因为开始时只显示一组柱子，当第一组柱子移动到差不多画布中央，第二组柱子才开始从右侧进入画布，所以要注意2组柱子X坐标的设置。
6. 在精灵的碰撞检测函数中增加柱子与小鸟碰撞的判断代码，如果发生碰撞则游戏失败，播放碰撞音效，小鸟继续自由下落，直到摔落到地基上，播放失败音效，停止游戏并显示游戏结束消息，结束消息中包括本地得分与最高得分，并提示用户再来一盘或者结束游戏。
7. 拖入两个精灵，命名为“pause”，图片素材pause.png；命名“begin”, 图片素材get\_ready.png。 并刚开始时显示pause，点击这个图标screen2就会处于暂停，并且图标会变为“begin”，再次点击屏幕开始，并变为pause
8. 拖入三个精灵，图片分别为1.png,2.png,3.png，点击其中一个时，小鸟的速度就会改变为对应的难度，并且该图标会隐藏，其他两个图标会可见

# 实验结果与分析

手机屏幕的截图

中度可信度描述已自动生成

初设置难度为1，所以1消失，并点击右上角暂停游戏

手机屏幕的截图

低可信度描述已自动生成

点击左上角喇叭，出现斜杠则声音关闭，点击数字2，难度切换成2

图形用户界面, 应用程序

描述已自动生成

失败后，出现弹窗显示本局游戏分数和历史最高分数，并可以选择退出或重新开始游戏

文本

描述已自动生成

点击重新开始后进入此页面，点击get ready即可重新开始游戏

# 实验总结

实验结果表明，本次实验成功实现了一个较为完整的Flappy Bird游戏，包括随机生成的柱子、计分功能、保存最高得分、声音开关、游戏暂停以及三种游戏难度选择等功能。游戏运行流畅，柱子能够随机生成并以恒定的速率从右往左运动，小鸟能够通过点击屏幕升高并尝试从柱子之间穿过。当小鸟碰到柱子或摔落到地基上时，游戏会停止并显示得分与最高得分，同时提供重玩或退出选项。

通过本次实验，我深刻理解了使用App Inventor进行游戏开发的过程，掌握了如何综合运用各种编程元素和数据手段实现复杂的游戏逻辑。同时，我也体会到了游戏开发中测试与调试的重要性，通过多次调整和优化，最终实现了预期的游戏效果。