移动应用开发实验报告

实验一 摇一摇登录 bupt 校园网

1 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

2 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、桌面版 AI 伴侣

3 实验要求

1.创建一个登录北邮校园网的实用 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

4 实验原理

5 实验步骤

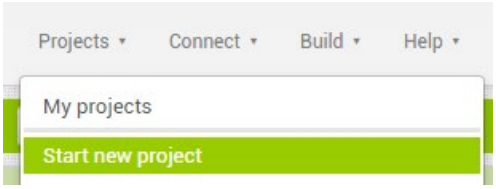
5.1 打开开发环境

打开AppInventor.cmd，启动离线版的AppInventor2
打开chrome浏览器，输入localhost:8888
默认进入开发环境的设计视图

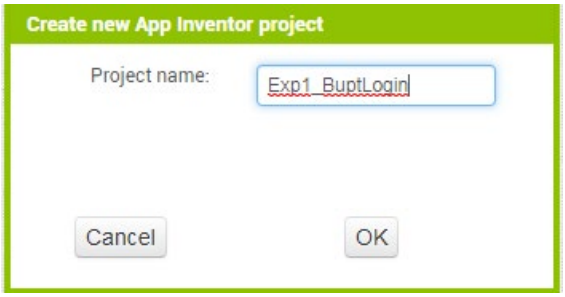
点击右上角的语言切换菜单，选择 English，实验将以英文进行编程

5.2 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_BuptLogin

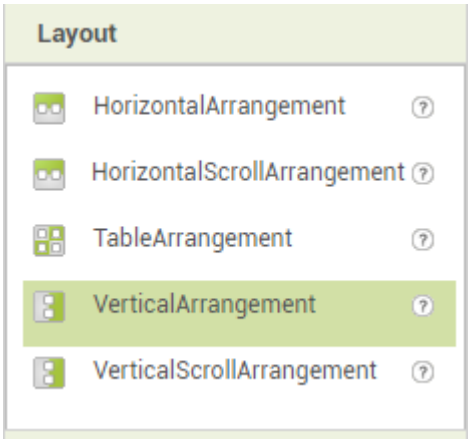


5.3 设计界面

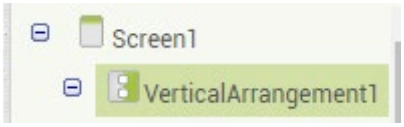
首先进行界面设置

5.3.1添加Layout 组件并设置属性

1. 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

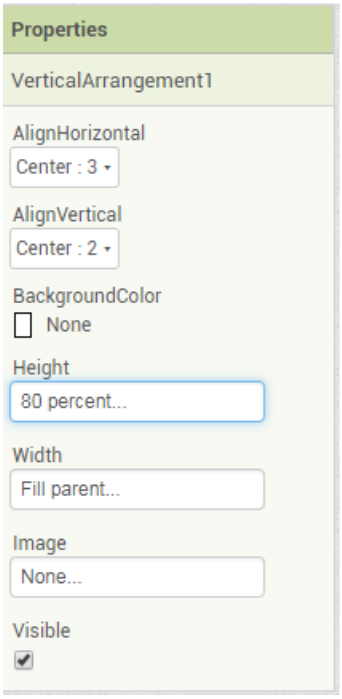


点击 viewer 面板的 VerticalArrangement1 的图标或 Components 中的 VerticalArrangement1，最右侧的 Properties 将显示点中的 VerticalArrangement1 的属性，需要修改部分属性以满足实验要求：

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。

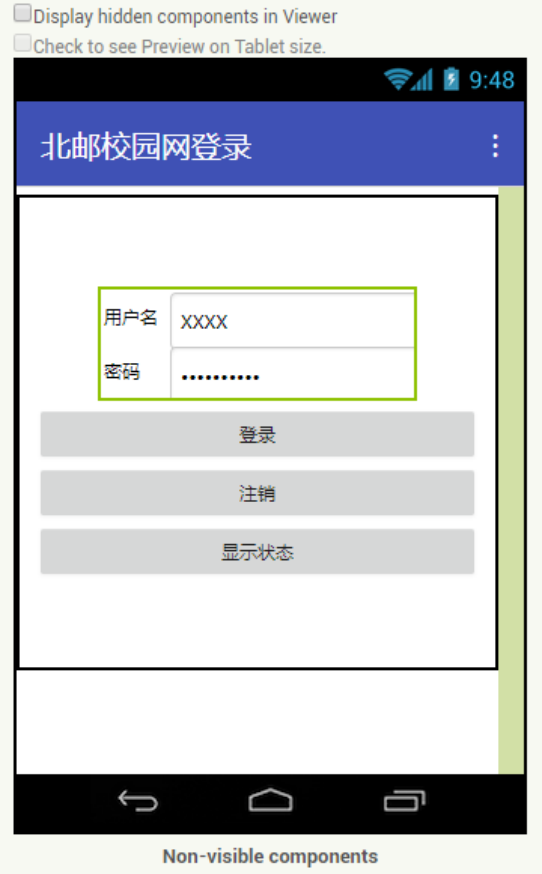
设置 Width 属性为 Fill parent。表示充满整个横向空间



2.按照同样的方法，拖入 TableArrangement1 到 VerticalArrangement1 中，并设置属性如下

5.3.2 添加Button和TextBox等可视组件并设置属性

按照下图所示位置，拖入相应的可视组件，作用是提供用户操作的界面：



3. 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

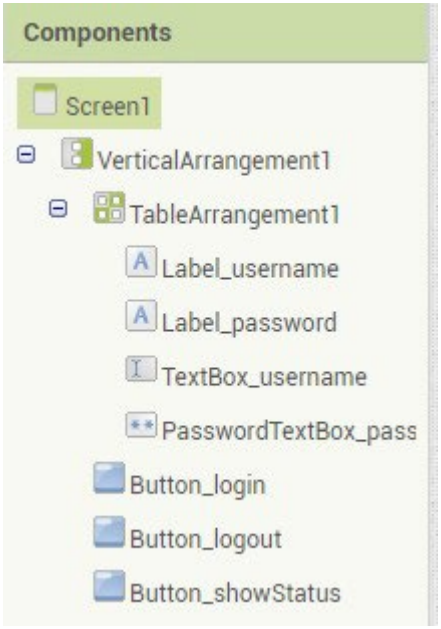
4. 拖入 1个 TextBox 文本框组件：用鼠标点中 User Interface下 的 TextBox，拖动到右侧 viewer 中 TableArrangement1 中相应的位置

5. 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。

6. 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

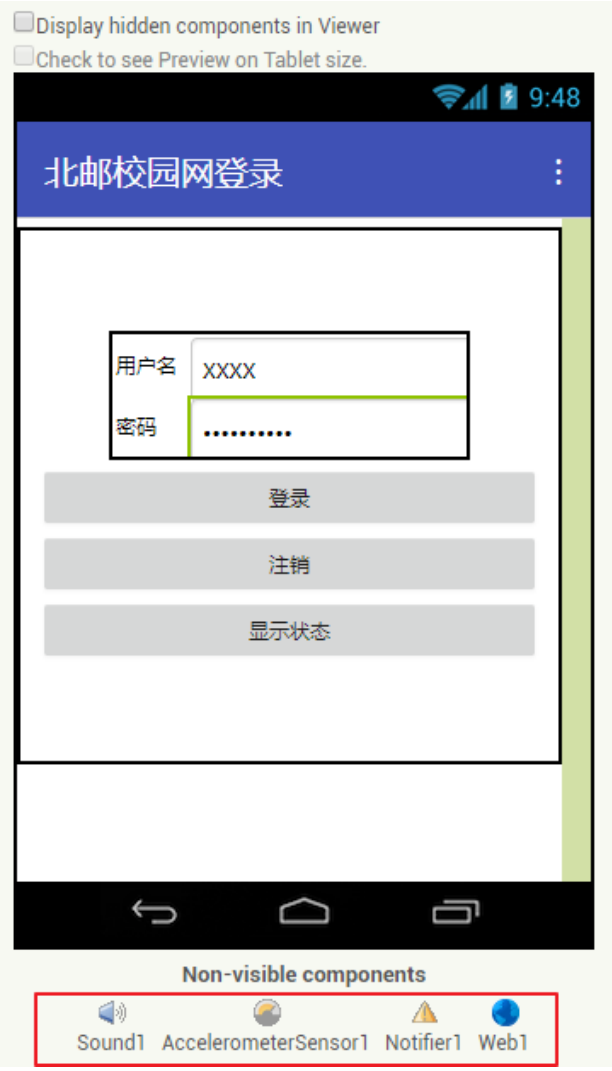
7. 设置组件所显示的文本。在 Components 面板中选中上面拖入的组件，然后在右侧 properties 的Text 框，将各组件默认显示的文本改成下图所示文本：

8.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。各个组件的重命名如下图所示：



5.3.3 添加Web客户端等非可视组件并设置属性

按照下图所示，拖入相应的非可视组件，作用是提供后台功能：



9. 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

10. 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

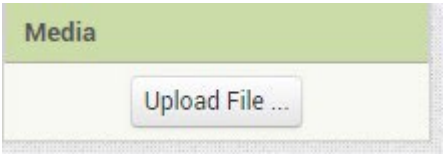
11. 拖入 1 个 Sound（音效播放器）组件：用鼠标点中 Media 多媒体下的 Sound，拖动到右侧 Viewer 中

12. 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。各个非可视组件的属性可保持默认，无需修改

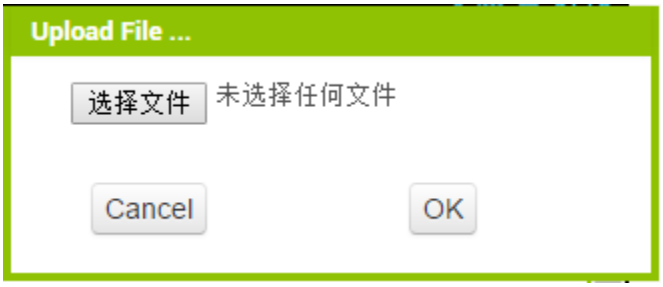
5.3.4 添加素材文件

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，



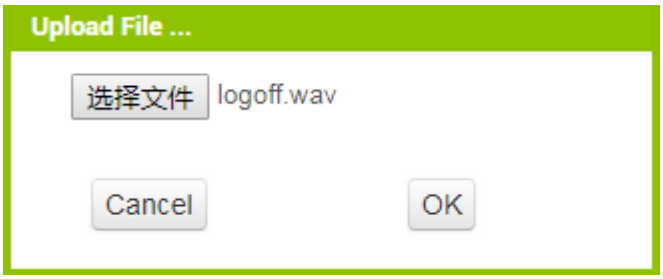
弹出上传文件对话框，选择“选择文件”，



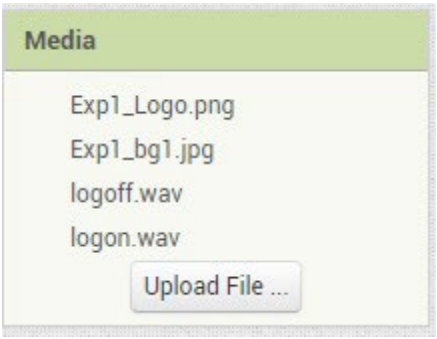
在文件选择框找到实验提供的素材：

音效文件：logon.wav、logoff.wav

图片素材：Exp1_Logo.png、Exp_bg1.jpg



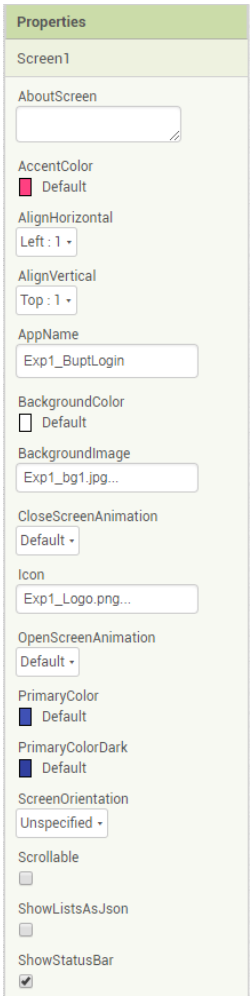
点击 OK，完成素材的上传



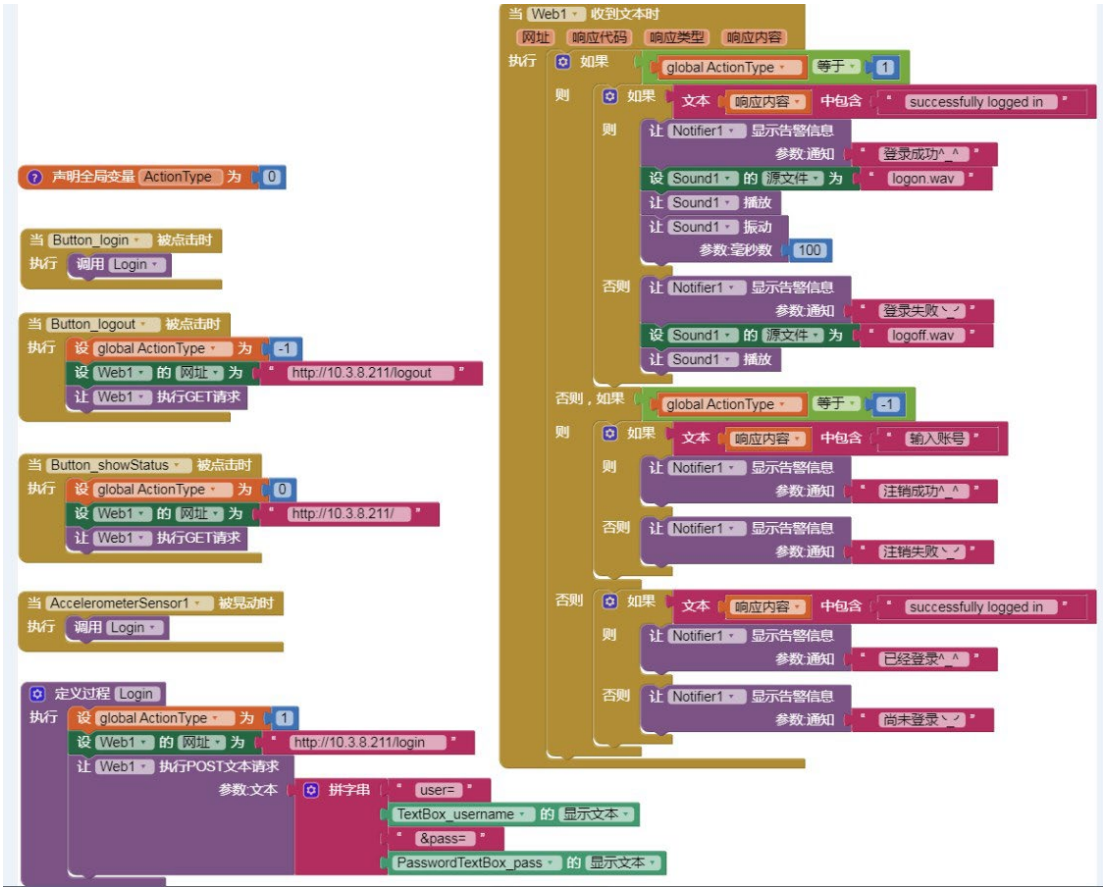
5.3.5 设置Screen属性

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：DeviceDefault

Title：北邮校园网登录
BackgroundImage:Exp_bg1.jpg



5.4 编程处理


声明全局变量 ActionType 为 0
当 Button_login 被点击时
执行 调用 Login
当 Button_logout 被点击时
执行 设 global ActionType 为 -1
设 Web1 的 网址 为 http://10.3.8.211/logout
让 Web1 执行GET请求
当 Button_showStatus 被点击时
执行 设 global ActionType 为 0
设 Web1 的 网址 为 http://10.3.8.211/
让 Web1 执行GET请求
当 AccelerometerSensor1 被晃动时
执行 调用 Login
定义过程 Login
执行 设 global ActionType 为 1
设 Web1 的 网址 为 http://10.3.8.211/login
让 Web1 执行POST文本请求
参数 文本 拼字串 user=
TextBox_username 的 显示文本
&pass=
PasswordTextBox_pass 的 显示文本
当 Web1 收到文本时
网址 响应代码 响应类型 响应内容
执行 如果 global ActionType 等于 1
则 如果 文本 响应内容 中包含 successfully logged in
则 让 Notifier1 显示告警信息
参数 通知 登录成功^_^
设 Sound1 的 源文件 为 logon.wav
让 Sound1 播放
让 Sound1 振动
参数 毫秒数 100
否则 让 Notifier1 显示告警信息
参数 通知 登录失败丶ノ
设 Sound1 的 源文件 为 logoff.wav
让 Sound1 播放
否则，如果 global ActionType 等于 -1
则 如果 文本 响应内容 中包含 输入账号
则 让 Notifier1 显示告警信息
参数 通知 注销成功^_^
否则 让 Notifier1 显示告警信息
参数 通知 注销失败丶ノ
否则 如果 文本 响应内容 中包含 successfully logged in
则 让 Notifier1 显示告警信息
参数 通知 已经登录^_^
否则 让 Notifier1 显示告警信息
参数 通知 尚未登录丶ノ

6 实验结果与分析

点击“显示状态”按钮，正常将显示登录状态。
输入用户名和密码，点击“登录”按钮，正常将提示登录成功。
点击“注销”按钮，正常将显示注销成功。

7 实验总结

实验二 Flappy Bird 1 — 扬翅的小鸟

1. 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

2. 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_32bit）
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 CompanionApp。

3. 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。

4. 实验原理

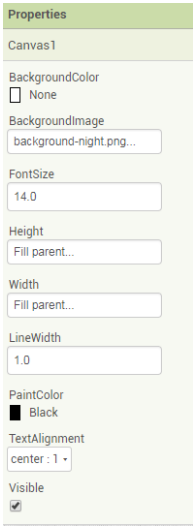
5. 实验步骤

步骤1：上传素材

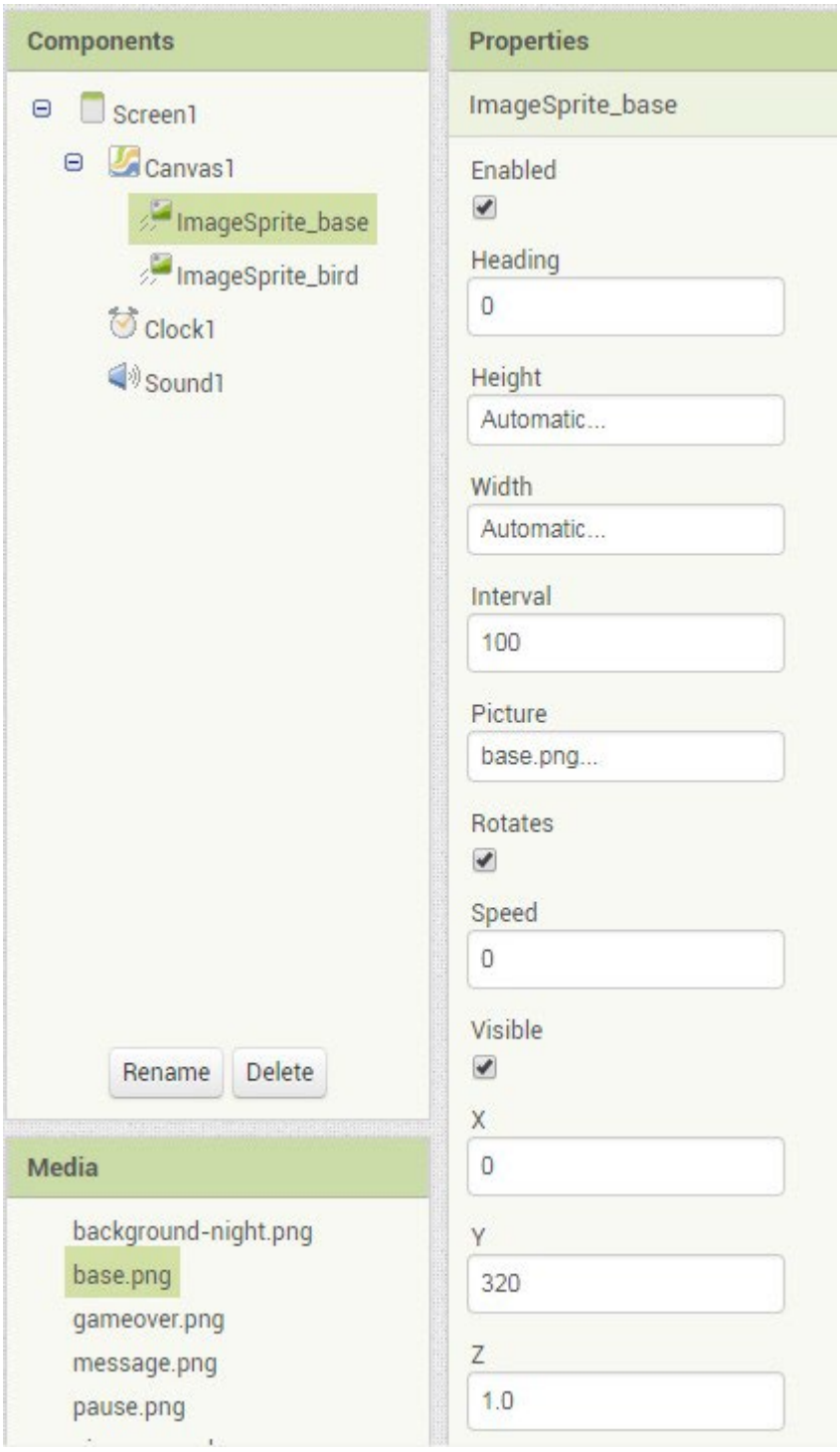
在右下角的 Media-》Upload Files 上传以下文件。

类型	文件名	作用
声音	wing.ogg	翅膀扇动的音效
图像	background-night.png	游戏背景
图像	base.png	地基
图像	yellowbird-midflap.png	小鸟在中间位置的位图
图像	yellowbird-upflap.png	小鸟向上扑腾的位图
图像	yellowbird-downflap.png	小鸟往下飞的位图

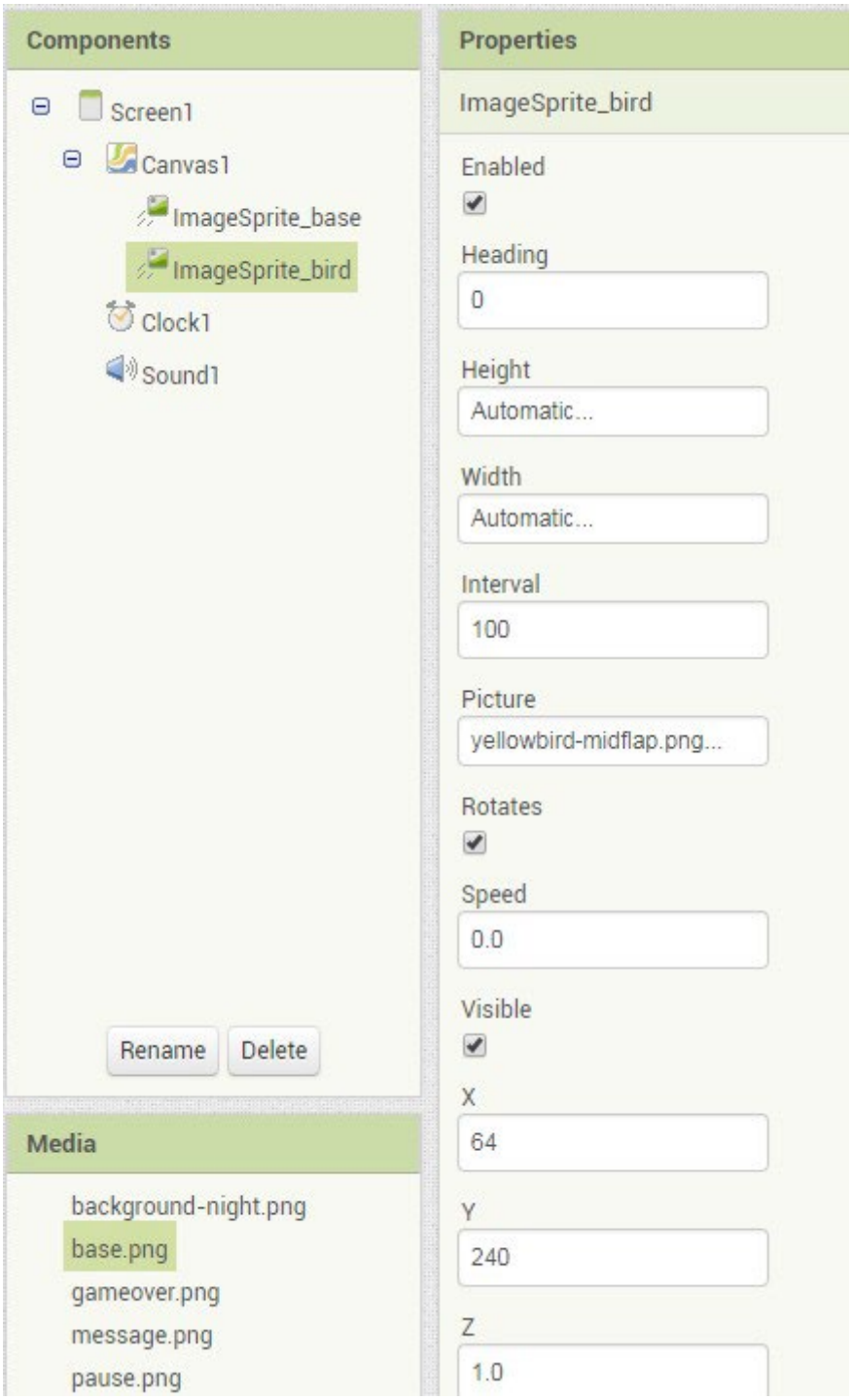
步骤2：添加画布（Canvas），并设置画布属性



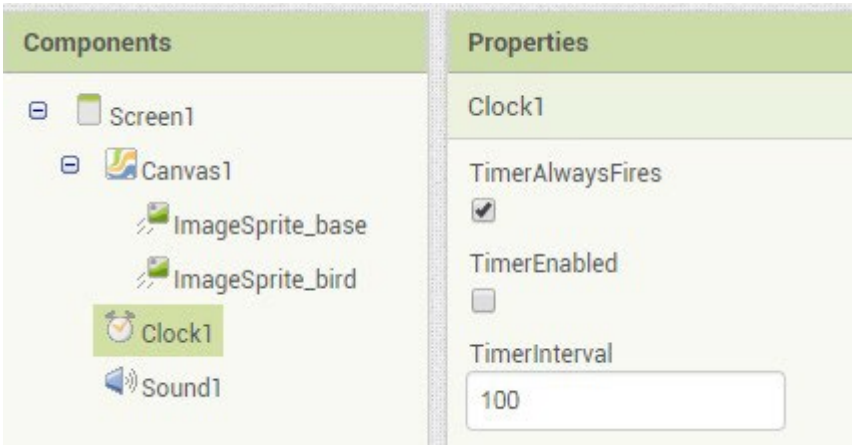
步骤 3：在 Canvas1 中添加地基精灵（ImageSprite）重命名为 ImageSprite_base，并设置好属性



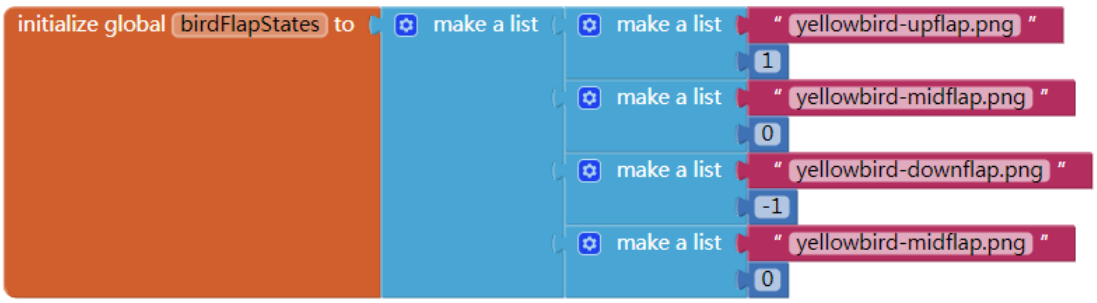
步骤 4：在 Canvas1 中添加小鸟精灵（ImageSprite），重命名为 ImageSprite_bird，并设置好属性。

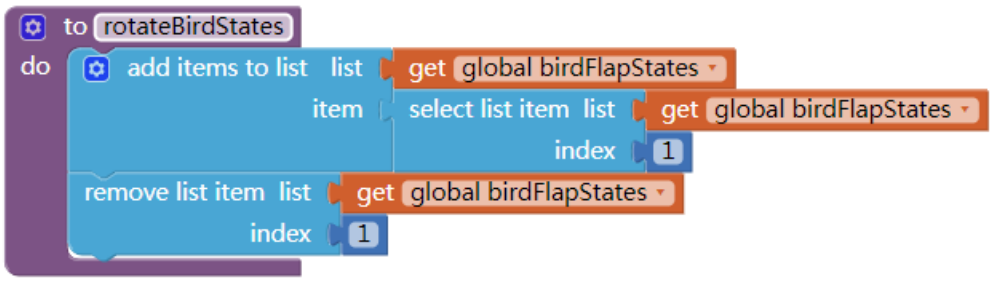


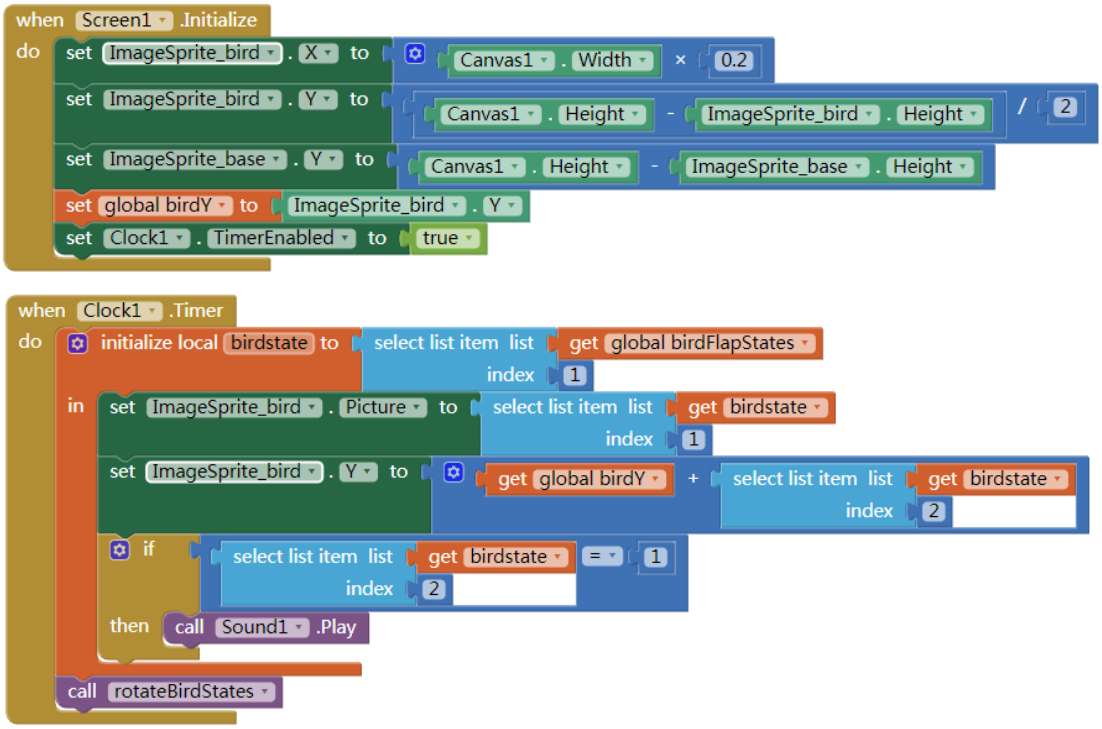
步骤 5：添加一个定时器（Clock），并设置好属性。



步骤 6：添加一个音效播放器（Sound），并设置好属性。







6. 实验结果与分析

小鸟扇动翅膀，并发出音效

7. 实验总结

实验三 Flappy Bird 2 — 下坠的小鸟

1. 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

2. 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_32bit）

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 CompanionApp。

3. 实验要求

本实验将要在实验 1 的基础上增加以下功能：

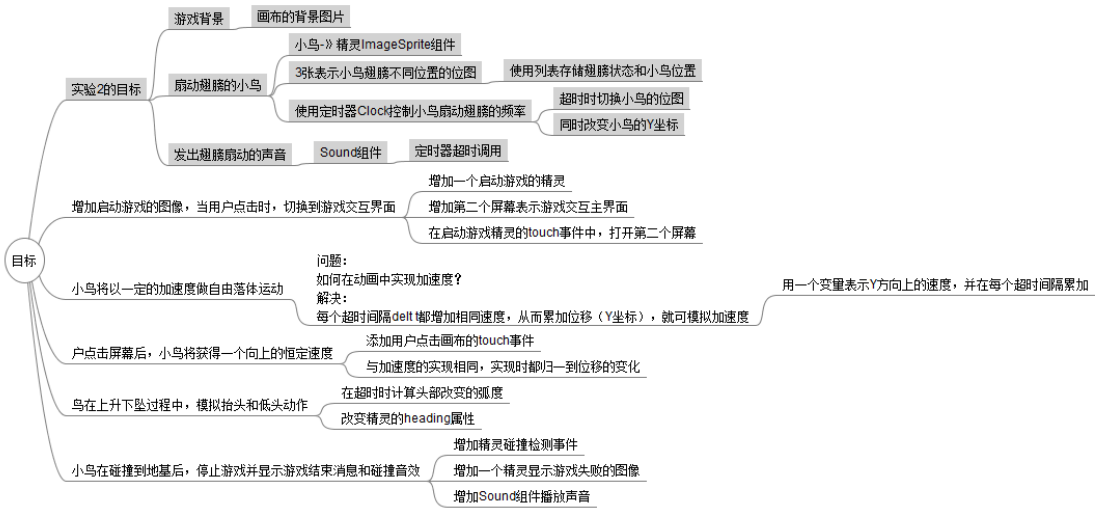
1. 增加启动游戏的图像，当用户点击时，切换到游戏交互界面。
2. 在游戏交互界面，小鸟将以一定的加速度做自由落体运动。
3. 用户点击屏幕后，小鸟将获得一个向上的恒定速度。
4. 小鸟在上升下坠过程中，需要模拟抬头和低头动作。
5. 小鸟在碰撞到地基后，停止游戏并显示游戏结束消息和碰撞音效。

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。

4. 实验原理



5. 实验步骤

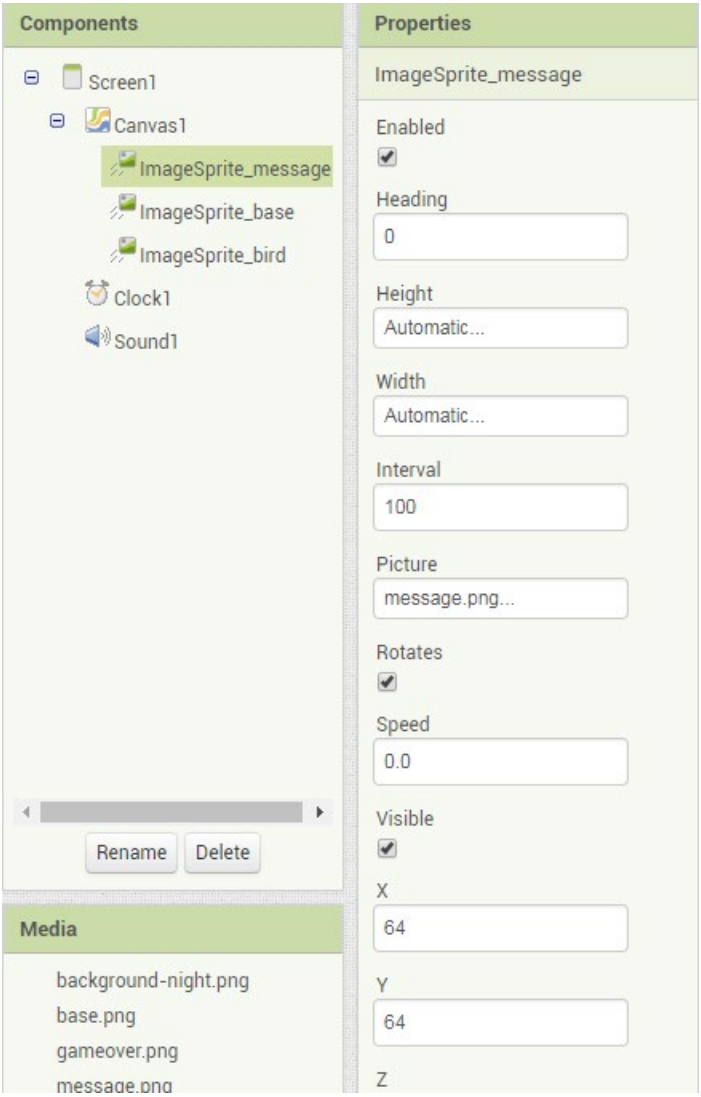
5.1 设置 Screen1 界面

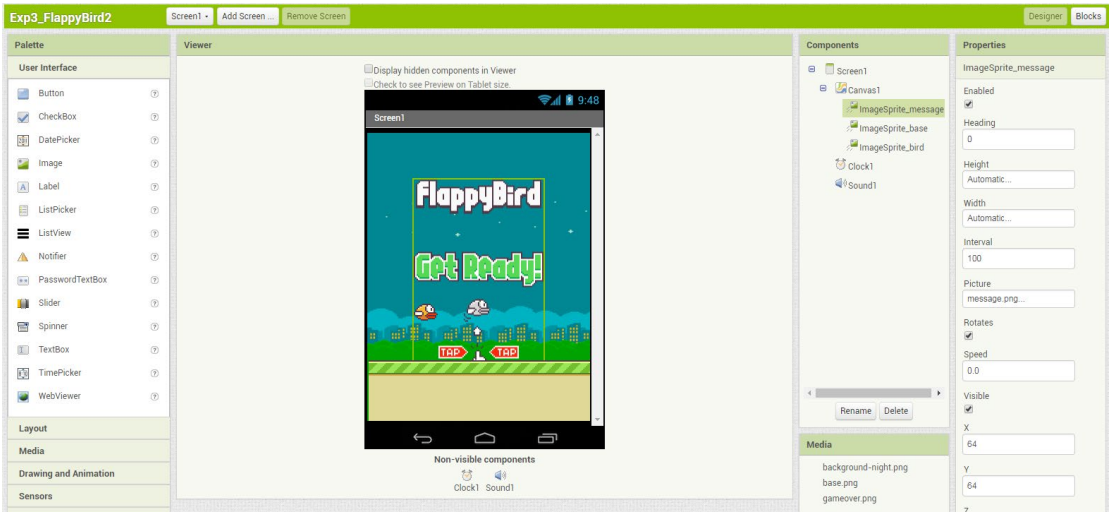
步骤 1：上传素材

在右下角的 Media-》Upload Files 上传以下文件。

类型	文件名	作用
图像	message.png	启动游戏的图片
图像	gameover.png	显示游戏结束的图片
声音	die.ogg	碰撞音效

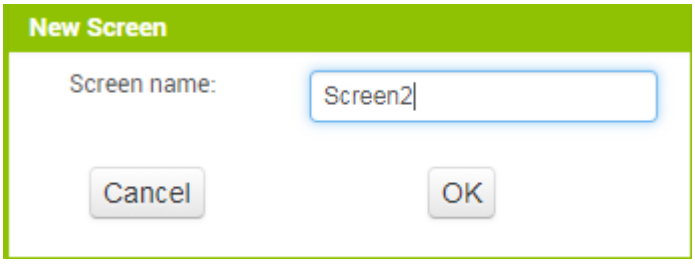
步骤 2：在 Canvas1 中添加精灵（ImageSprite），重命名为 ImageSprite_message，并设置好属性（主要是 Picture 源图像和 XY 坐标）。





5.2 增加 Screen2 界面

步骤 1：创建 Screen2。在开发环境上部点击 Add Screen，增加一块屏幕 Screen2



步骤 2：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

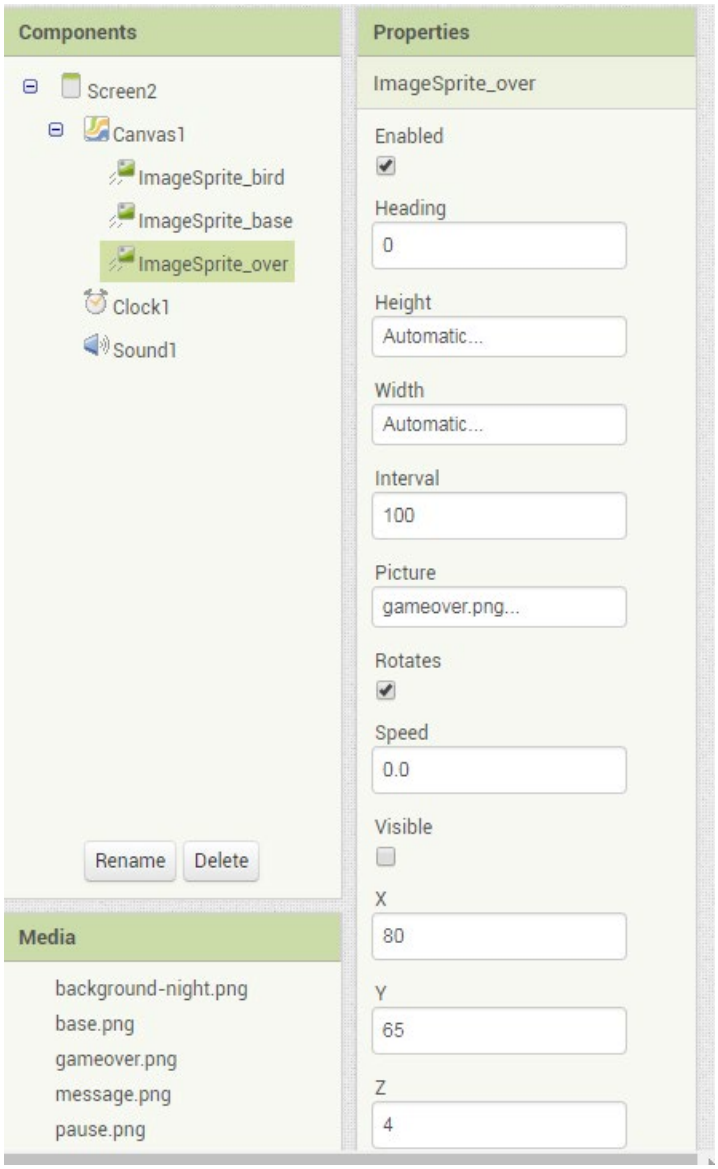
ImageSprite_base

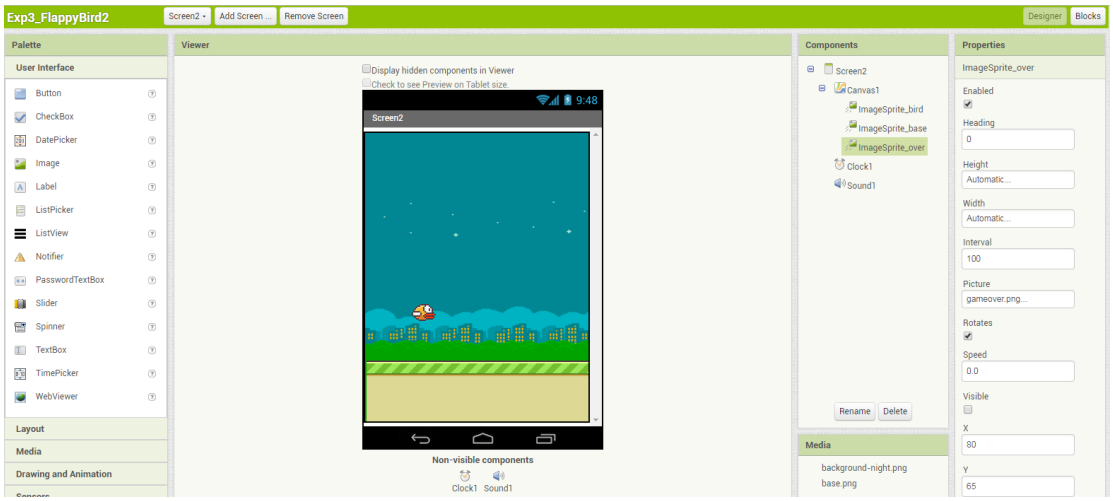
ImageSprite_bird

Clock1

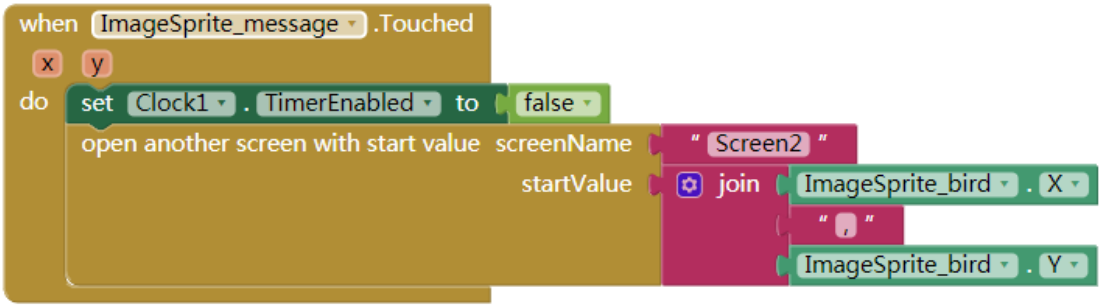
Sound1

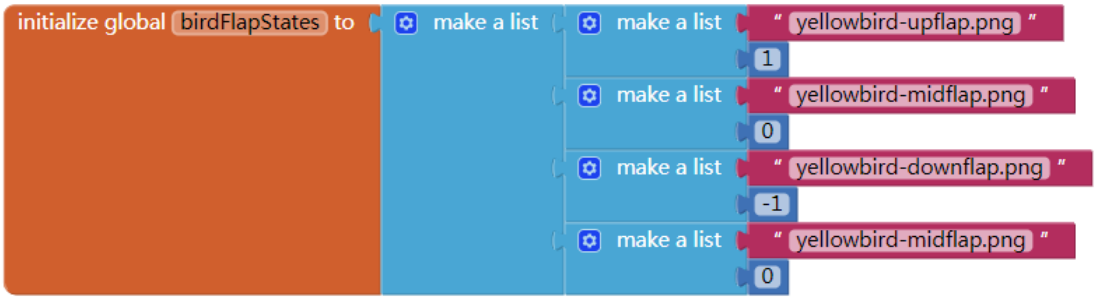
步骤 3：在 Canvas1 中添加精灵（ImageSprite），重命名为 ImageSprite_over，并设置好属性。

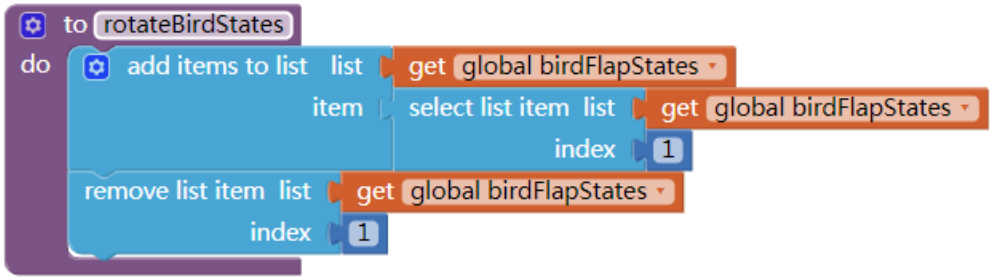

Components
Screen2
Canvas1
ImageSprite_bird
ImageSprite_base
ImageSprite_over
Clock1
Sound1
Rename
Delete
Media
background-night.png
base.png
gameover.png
message.png
pause.png
Properties
ImageSprite_over
Enabled
Heading
0
Height
Automatic...
Width
Automatic...
Interval
100
Picture
gameover.png...
Rotates
Speed
0.0
Visible
X
80
Y
65
Z
4

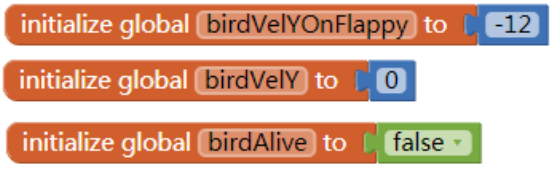

Exp3_FlappyBird2
Screen2
Add Screen ...
Remove Screen
Designer
Blocks
Palette
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size
Non-visible components
Clock1
Sound1
Components
Properties
Rename
Delete

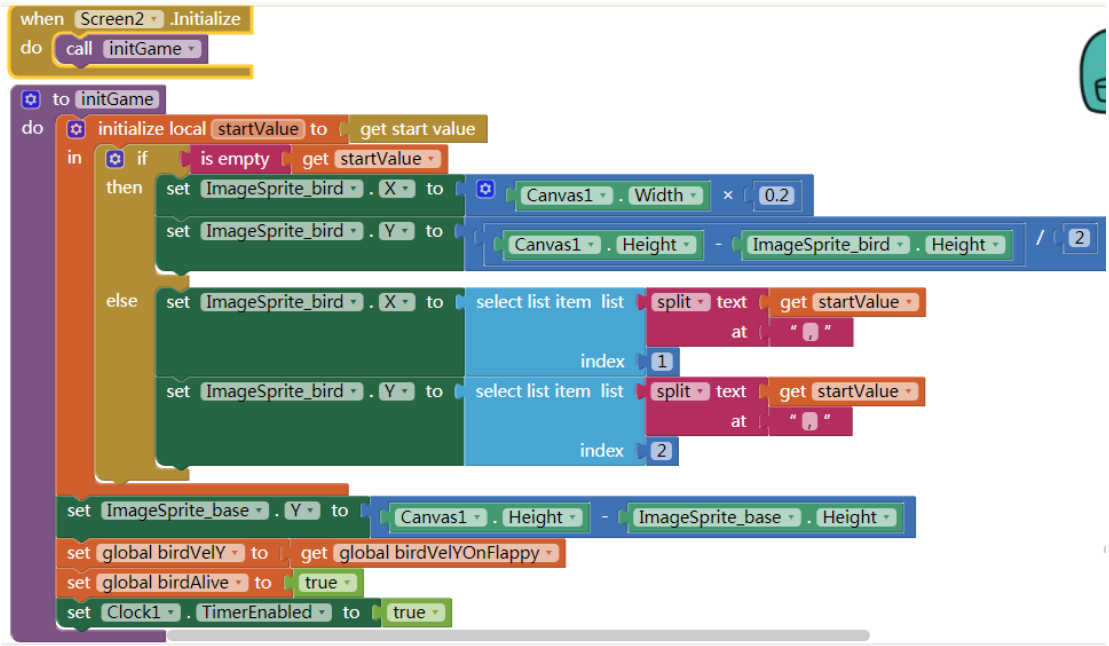
5.3 编程

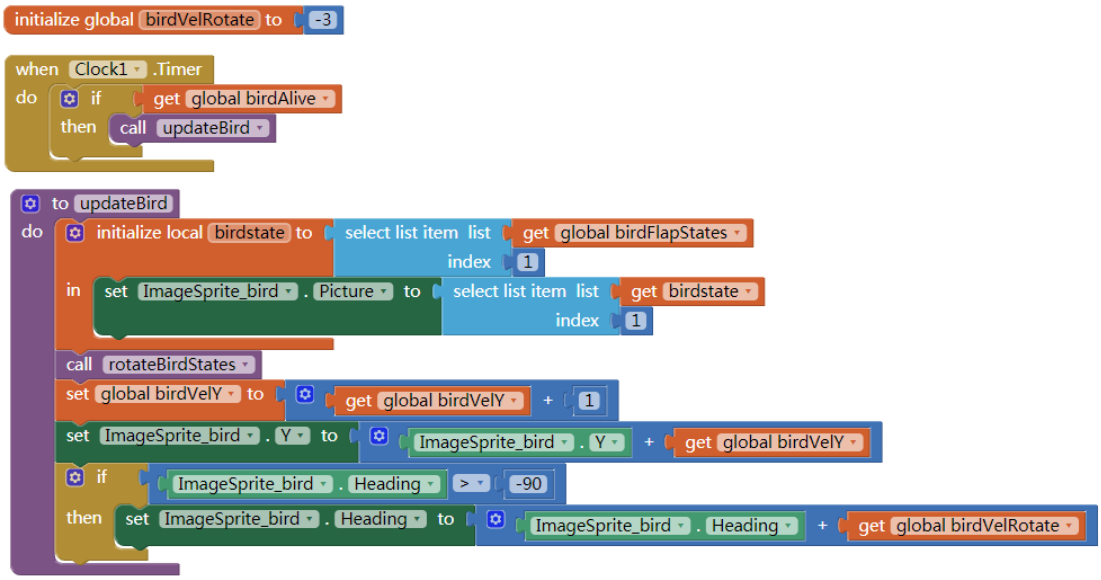

when ImageSprite_message .Touched
x y
do set Clock1 . TimerEnabled to false
open another screen with start value screenName " Screen2 "
startValue join ImageSprite_bird . X
" , "
ImageSprite_bird . Y

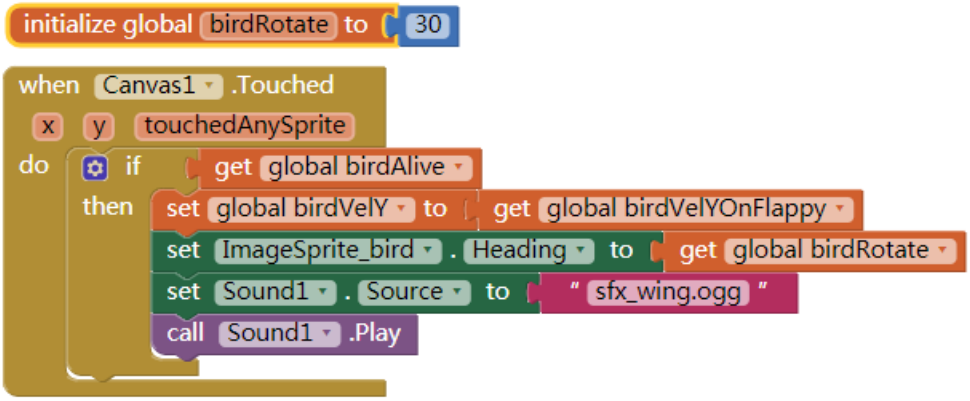

initialize global birdFlapStates to make a list
make a list " yellowbird-upflap.png "
1
make a list " yellowbird-midflap.png "
0
make a list " yellowbird-downflap.png "
-1
make a list " yellowbird-midflap.png "
0

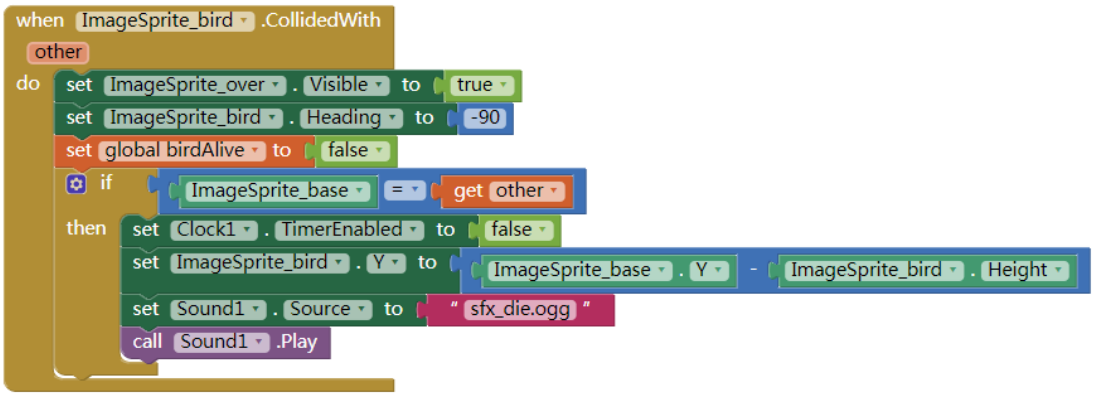

to rotateBirdStates
do add items to list list get global birdFlapStates
item select list item list get global birdFlapStates
index 1
remove list item list get global birdFlapStates
index 1


initialize global birdVelYOnFlappy to -12
initialize global birdVelY to 0
initialize global birdAlive to false


when Screen2 .Initialize
do call initGame
to initGame
do initialize local startValue to get start value
in if is empty get startValue
then set ImageSprite_bird . X to Canvas1 . Width × 0.2
set ImageSprite_bird . Y to Canvas1 . Height - ImageSprite_bird . Height / 2
else set ImageSprite_bird . X to select list item list split text get startValue
at " , "
index 1
set ImageSprite_bird . Y to select list item list split text get startValue
at " , "
index 2
set ImageSprite_base . Y to Canvas1 . Height - ImageSprite_base . Height
set global birdVelY to get global birdVelYOnFlappy
set global birdAlive to true
set Clock1 . TimerEnabled to true


initialize global birdVelRotate to -3
when Clock1 .Timer
do if get global birdAlive
then call updateBird
to updateBird
do initialize local birdstate to select list item list get global birdFlapStates
index 1
in set ImageSprite_bird . Picture to select list item list get birdstate
index 1
call rotateBirdStates
set global birdVelY to get global birdVelY + 1
set ImageSprite_bird . Y to ImageSprite_bird . Y + get global birdVelY
if ImageSprite_bird . Heading > -90
then set ImageSprite_bird . Heading to ImageSprite_bird . Heading + get global birdVelRotate


initialize global birdRotate to 30
when Canvas1 .Touched
x y touchedAnySprite
do if get global birdAlive
then set global birdVelY to get global birdVelYOnFlappy
set ImageSprite_bird . Heading to get global birdRotate
set Sound1 . Source to " sfx_wing.ogg "
call Sound1 .Play


when ImageSprite_bird .CollidedWith
other
do set ImageSprite_over . Visible to true
set ImageSprite_bird . Heading to -90
set global birdAlive to false
if ImageSprite_base = get other
then set Clock1 . TimerEnabled to false
set ImageSprite_bird . Y to ImageSprite_base . Y - ImageSprite_bird . Height
set Sound1 . Source to " sfx_die.ogg "
call Sound1 .Play

6. 实验结果与分析

7. 实验总结

实验四 Flappy Bird 3 — 避障的小鸟

1. 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

2. 实验环境

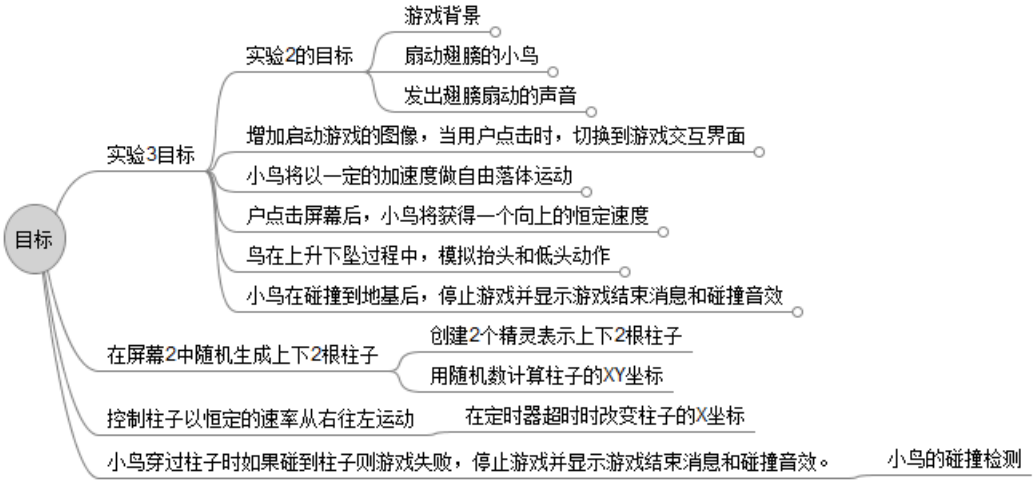
1. 硬件环境
   PC 微机
2. 软件环境
   Windows 操作系统
   App Inventor2018 离线版（AppInventor2018PersonalEdition_Win）
   桌面版 AI 伴侣（AI2Companion_Win7_32bit）
3. 如果使用 Android 手机进行测试，还需要 Android 手机和安装在手机上的 MIT AI2 CompanionApp。

3. 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。

4. 实验原理



5. 实验步骤

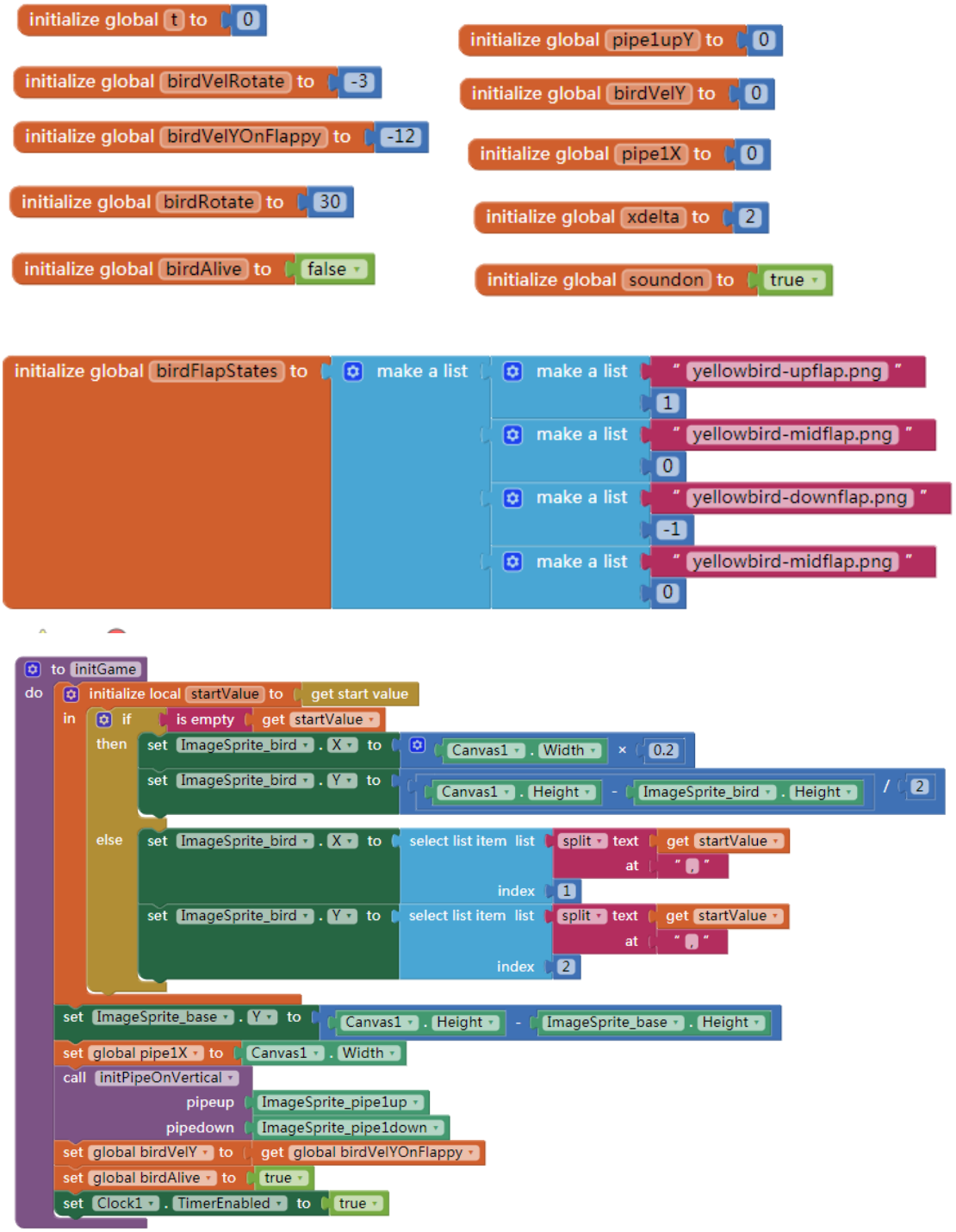
5.1 修改 Screen2 界面

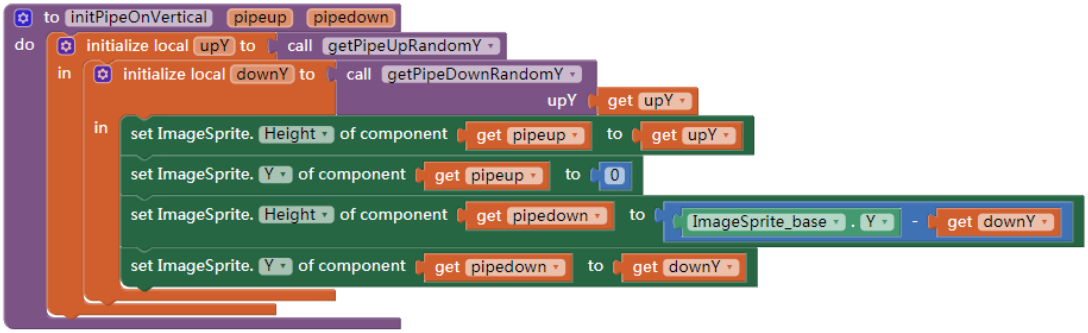
1 在 Screen2 中增加 2 根柱子。

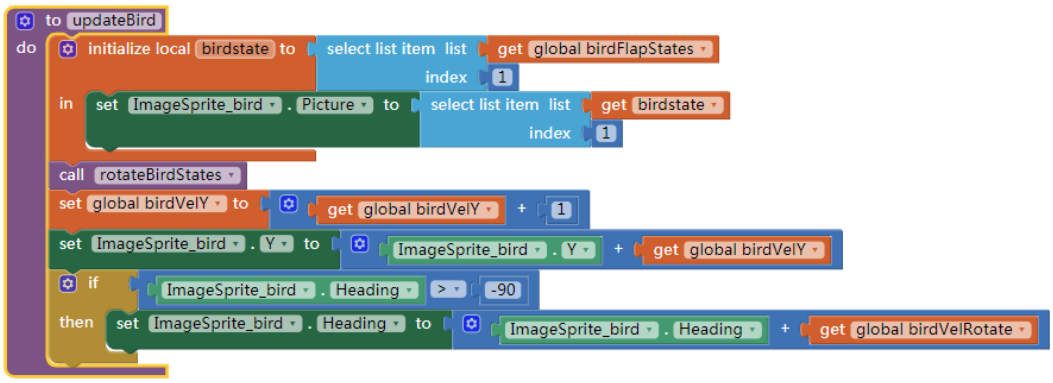
1.1 上传柱子的图像素材文件：pipe-green_down.png 和 pipe-green_up.png。

1.2 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

5.2 编程

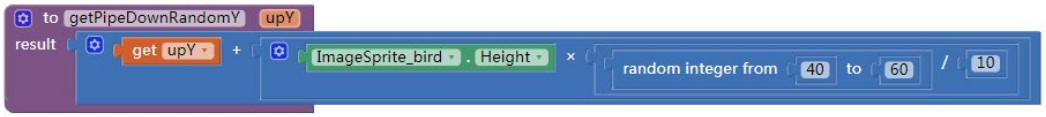

initialize global t to 0
initialize global birdVelRotate to -3
initialize global birdVelYOnFlappy to -12
initialize global birdRotate to 30
initialize global birdAlive to false
initialize global pipe1upY to 0
initialize global birdVelY to 0
initialize global pipe1X to 0
initialize global xdelta to 2
initialize global soundon to true
initialize global birdFlapStates to
make a list
make a list
" yellowbird-upflap.png "
1
make a list
" yellowbird-midflap.png "
0
make a list
" yellowbird-downflap.png "
-1
make a list
" yellowbird-midflap.png "
0
to initGame
do
initialize local startValue to get start value
in
if is empty get startValue
then
set ImageSprite_bird . X to Canvas1 . Width × 0.2
set ImageSprite_bird . Y to Canvas1 . Height - ImageSprite_bird . Height / 2
else
set ImageSprite_bird . X to select list item list split text get startValue at " , "
index 1
set ImageSprite_bird . Y to select list item list split text get startValue at " , "
index 2
set ImageSprite_base . Y to Canvas1 . Height - ImageSprite_base . Height
set global pipe1X to Canvas1 . Width
call initPipeOnVertical
pipeup ImageSprite_pipe1up
pipedown ImageSprite_pipe1down
set global birdVelY to get global birdVelYOnFlappy
set global birdAlive to true
set Clock1 . TimerEnabled to true

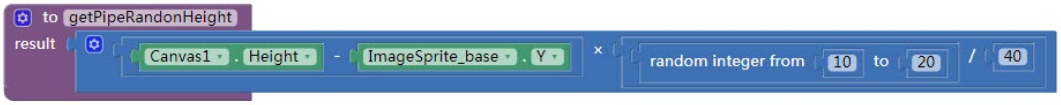

to initPipeOnVertical pipeup pipedown
do initialize local upY to call getPipeUpRandomY
in initialize local downY to call getPipeDownRandomY
upY get upY
in set ImageSprite. Height of component get pipeup to get upY
set ImageSprite. Y of component get pipeup to 0
set ImageSprite. Height of component get pipedown to ImageSprite_base . Y - get downY
set ImageSprite. Y of component get pipedown to get downY


to updateBird
do initialize local birdstate to select list item list get global birdFlapStates
index 1
in set ImageSprite_bird . Picture to select list item list get birdstate
index 1
call rotateBirdStates
set global birdVelY to get global birdVelY + 1
set ImageSprite_bird . Y to ImageSprite_bird . Y + get global birdVelY
if ImageSprite_bird . Heading > -90
then set ImageSprite_bird . Heading to ImageSprite_bird . Heading + get global birdVelRotate

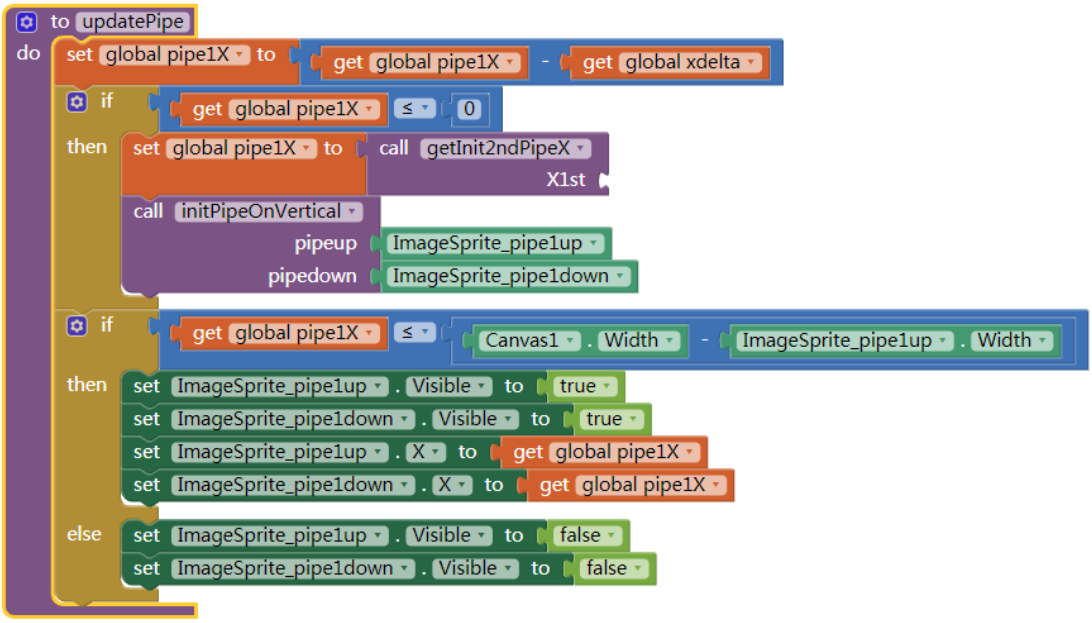

to getInit2ndPipeX X1st
result max get X1st + Canvas1 . Width × random integer from 10 to 20 / 30
Canvas1 . Width

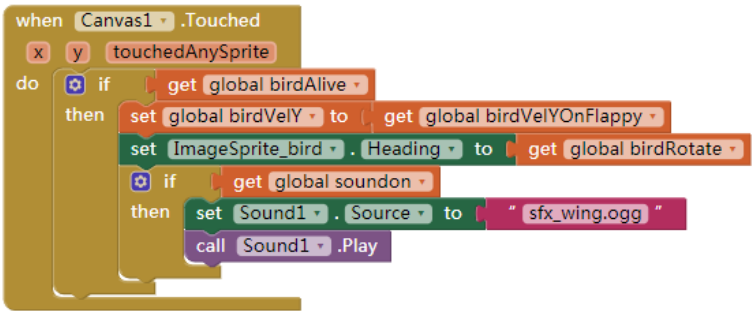

to getPipeUpRandomY
result ImageSprite_base . Y × random integer from 10 to 20 / 40

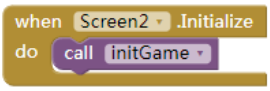

to getPipeDownRandomY upY
result get upY + ImageSprite_bird . Height × random integer from 40 to 60 / 10

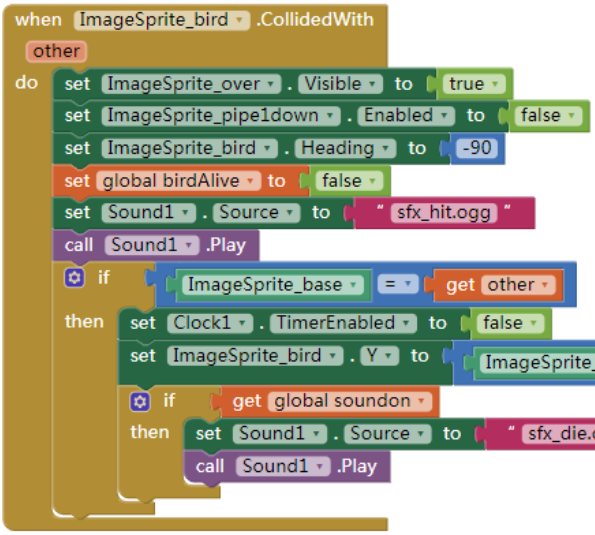

to getPipeRandonHeight
result Canvas1 . Height - ImageSprite_base . Y × random integer from 10 to 20 / 40

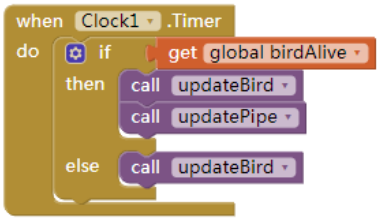

to rotateBirdStates
do add items to list list get global birdFlapStates
添加 item select list item list get global birdFlapStates
index 1
remove list item list get global birdFlapStates
index 1


to updatePipe
do
set global pipe1X to get global pipe1X - get global xdelta
if get global pipe1X ≤ 0
then set global pipe1X to call getInit2ndPipeX
X1st
call initPipeOnVertical
pipeup ImageSprite_pipe1up
pipedown ImageSprite_pipe1down
if get global pipe1X ≤ Canvas1 . Width - ImageSprite_pipe1up . Width
then set ImageSprite_pipe1up . Visible to true
set ImageSprite_pipe1down . Visible to true
set ImageSprite_pipe1up . X to get global pipe1X
set ImageSprite_pipe1down . X to get global pipe1X
else set ImageSprite_pipe1up . Visible to false
set ImageSprite_pipe1down . Visible to false

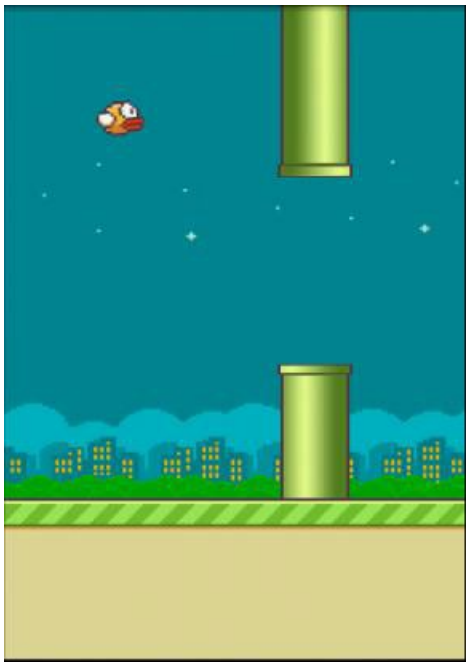

when Canvas1 .Touched
x y touchedAnySprite
do if get global birdAlive
then set global birdVelY to get global birdVelYOnFlappy
set ImageSprite_bird . Heading to get global birdRotate
if get global soundon
then set Sound1 . Source to " sfx_wing.ogg "
call Sound1 .Play


when Screen2 .Initialize
do call initGame


when ImageSprite_bird .CollidedWith
other
do set ImageSprite_over . Visible to true
set ImageSprite_pipe1down . Enabled to false
set ImageSprite_bird . Heading to -90
set global birdAlive to false
set Sound1 . Source to " sfx_hit.ogg "
call Sound1 .Play
if ImageSprite_base = get other
then set Clock1 . TimerEnabled to false
set ImageSprite_bird . Y to ImageSprite_base . Y - ImageSprite_bird . Height
if get global soundon
then set Sound1 . Source to " sfx_die.ogg "
call Sound1 .Play


when Clock1 .Timer
do if get global birdAlive
then call updateBird
call updatePipe
else call updateBird

6. 实验结果与分析



7. 实验总结