第四組分組專題報告

主題：吃豆小精靈

小組名單：

組長	**40723244**	劉冠志	負責：撰寫程式
組員	**40723211**	李昌隆	負責：無
組員	**40723233**	陳謙諭	負責：投影片製作
組員	**40723234**	曾信誠	負責：小組網頁更新
組員	**40723237**	黃俊翔	負責：小組網頁更新
組員	**40723241**	葉承叡	負責：報告撰寫

參考資料：

程式參考

https://mdecp2018.github.io/finalproject-bgx/content/Spacegame.html

第三組之部分程式

Kmolgame 內部各範例

圖片取材

https://www.google.com.tw/

自評成績：

組長	**40723244**	劉冠志	**90** 分
組員	**40723211**	李昌隆	**0** 分
組員	**40723233**	陳謙諭	**75** 分
組員	**40723234**	曾信誠	**75** 分
組員	**40723237**	黃俊翔	**75** 分
組員	**40723241**	葉承叡	**75** 分

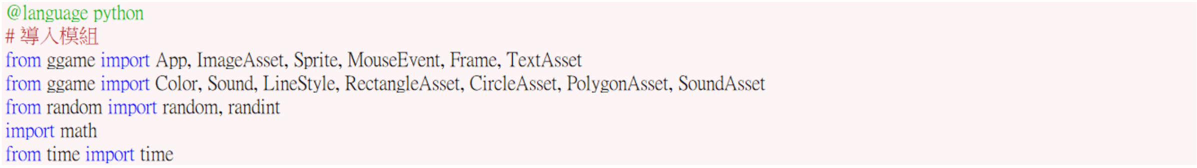
製作動機：

因範例中的程式皆與圖像移動有關，因此我們想利用這個點來作為出發點，製作出與圖像移動相關的專題，經討論後決定以吃豆小精靈為題目製作該專題。

程式介紹：

導入模組

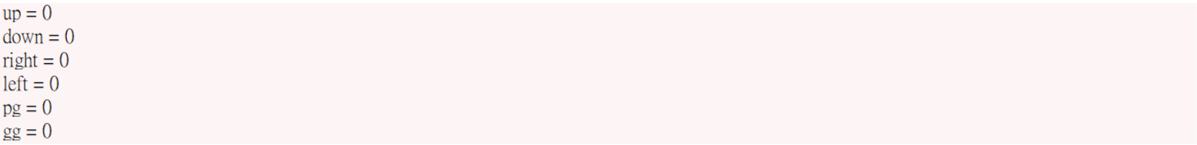
從 ggame 中導入下列的模組，讓程式在運行時不會顯得太過龐大且簡潔方便。



```
@language python
# 導入模組
from ggame import App, ImageAsset, Sprite, MouseEvent, Frame, TextAsset
from ggame import Color, Sound, LineStyle, RectangleAsset, CircleAsset, PolygonAsset, SoundAsset
from random import random, randint
import math
from time import time
```

設定變數

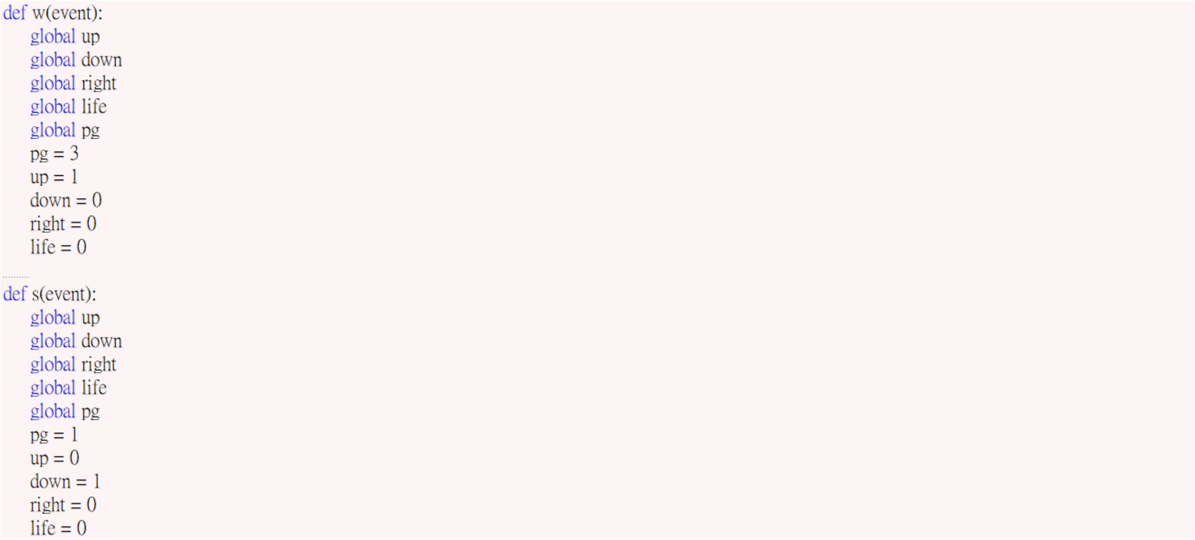
設定以下的全區域變數提供後續程式判斷與 if 事件時使用。



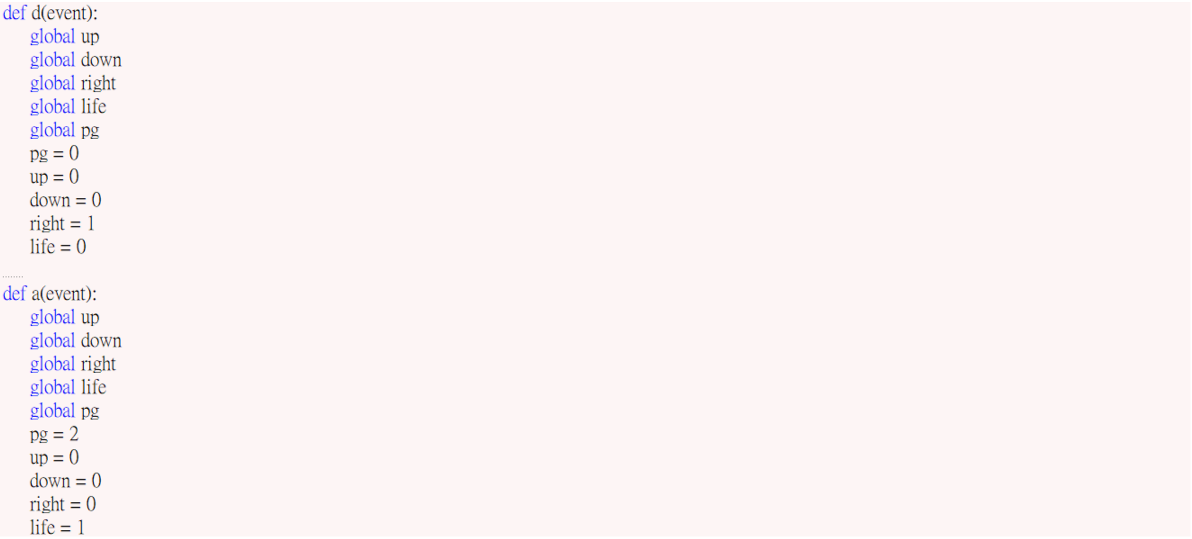
```
up = 0
down = 0
right = 0
left = 0
pg = 0
gg = 0
```

設定函數

設定以下的方向函數提供後續程式觸發時使用。



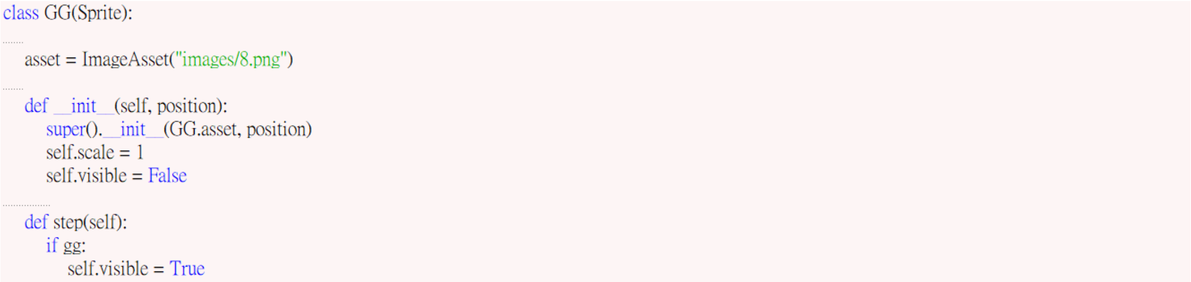
```
def w(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 3
    up = 1
    down = 0
    right = 0
    life = 0

def s(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 1
    up = 0
    down = 1
    right = 0
    life = 0
```



```
def d(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 0
    up = 0
    down = 0
    right = 1
    life = 0

def a(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 2
    up = 0
    down = 0
    right = 0
    life = 1
```

定義物件

設定以下的物件(GAME OVER、鬼魂、小精靈、豆子、背景)的圖片、移動參數、大小屬性以及隱藏屬性等，讓程式啟動後有物件的產生。



```
class GG(Sprite):

  asset = ImageAsset("images/8.png")

  def __init__(self, position):
    super().__init__(GG.asset, position)
    self.scale = 1
    self.visible = False

  def step(self):
    if gg:
      self.visible = True
```

```
class G(Sprite):

    asset = ImageAsset("images/5.png")

    def __init__(self, position):
        super().__init__(G.asset, position)
        self.scale = 0.5

    def step(self):
        if random() < 0.1:
            self.x += randint(-50,50)
            self.y += randint(-50,50)
```

```
class Bunny(Sprite):

    asset = ImageAsset("images/444.png",Frame(0,0,230,208), 4)

    def __init__(self, position):
        super().__init__(Bunny.asset, position)
        App.listenKeyEvent('keydown', 'w', w)
        App.listenKeyEvent('keydown', 's', s)
        App.listenKeyEvent('keydown', 'd', d)
        App.listenKeyEvent('keydown', 'a', a)
        self.scale = 0.4
```

```
    def step(self):
        global up
        global down
        global right
        global life
        global pg
        global gg
        self.G = app.G
        if up and self.y > 0:
            self.setImage(pg)
            self.y -= 10
        if down and self.y < 770:
            self.setImage(pg)
            self.y += 10
        if right and self.x < 1710:
            self.setImage(pg)
            self.x += 10
        if life and self.x > 0:
            self.setImage(pg)
            self.x -= 10
        if self.G.x + self.G.width >= self.x >=self.G.x and self.G.y + self.G.height >= self.y >= self.G.y:
            self.visible = False
            gg = 1
```

```
class Bg(Sprite):

    asset = ImageAsset("images/bg.png")

    def __init__(self, position):
        super().__init__(Bg.asset, position)

    def step(self):
            self.x = 0
            self.y = 0
```

```
class A(Sprite):

    asset = ImageAsset("images/a.png")

    def __init__(self, position):
        super().__init__(A.asset, position)
        self.scale = 0.1

    def step(self):
        self.Bunny = app.Bunny
        if self.Bunny.x + self.Bunny.width >= self.x >=self.Bunny.x and self.Bunny.y + self.Bunny.height >= self.y >= self.Bunny.y:
            self.visible = False
```

```
class AA(Sprite):

    asset = ImageAsset("images/a.png")

    def __init__(self, position):
        super().__init__(AA.asset, position)
        self.scale = 0.15

    def step(self):
        self.Bunny = app.Bunny
        if self.Bunny.x + self.Bunny.width >= self.x >=self.Bunny.x and self.Bunny.y + self.Bunny.height >= self.y >= self.Bunny.y:
            self.visible = False
            self.Bunny.scale =0.7
```

定義啟動參數

設定程式的啟動參數以及物件的初始位置，讓程式可以順利啟動。

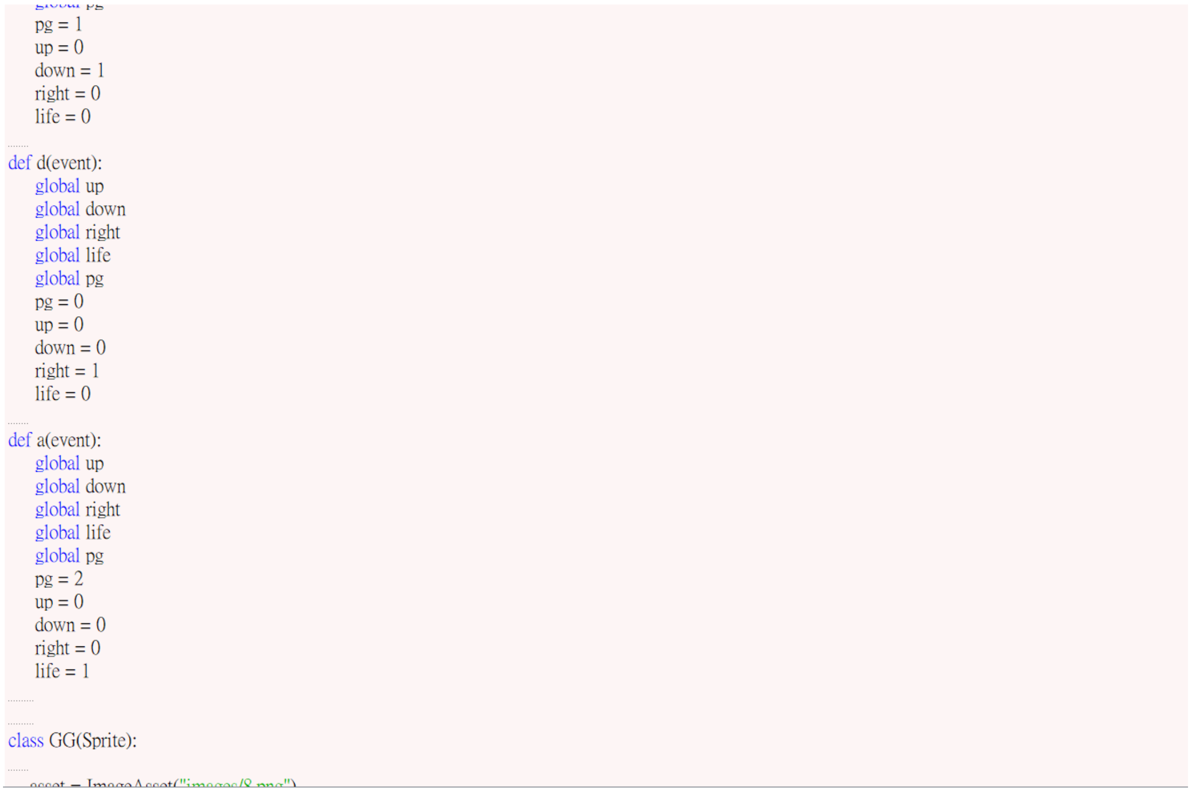
```
class DemoApp(App):


    def __init__(self):
        super().__init__()
        Bg((self.width/2, self.height/2))
        for i in range(20):
            A((randint(50,1680),randint(50,750)))
        for c in range(1):
            AA((randint(50,1680),randint(50,750)))
        self.Bunny = Bunny((0,0))
        self.G = G((840, 250))
        self.GG = GG((740, 250))

    def step(self):
        for bunny in self.spritelist:
            bunny.step()
```

```
app = DemoApp()
app.run()
```

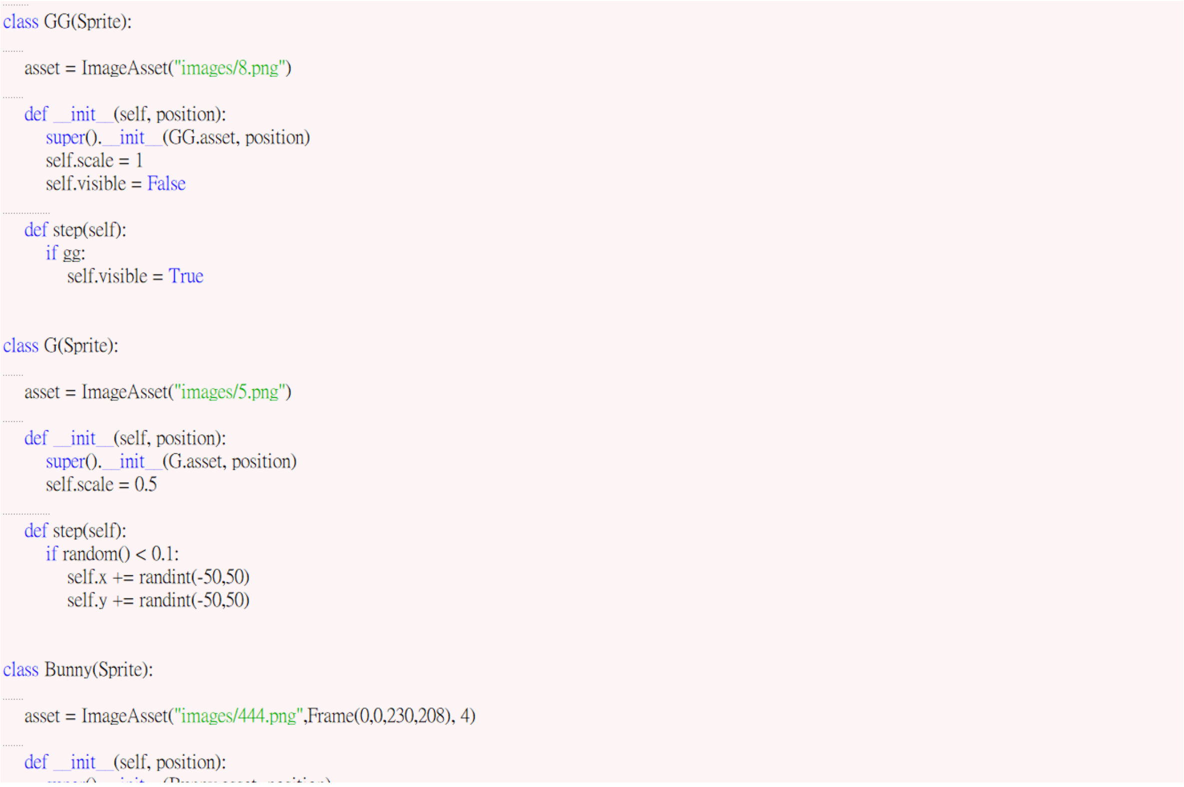
程式整體

整體程式畫面。

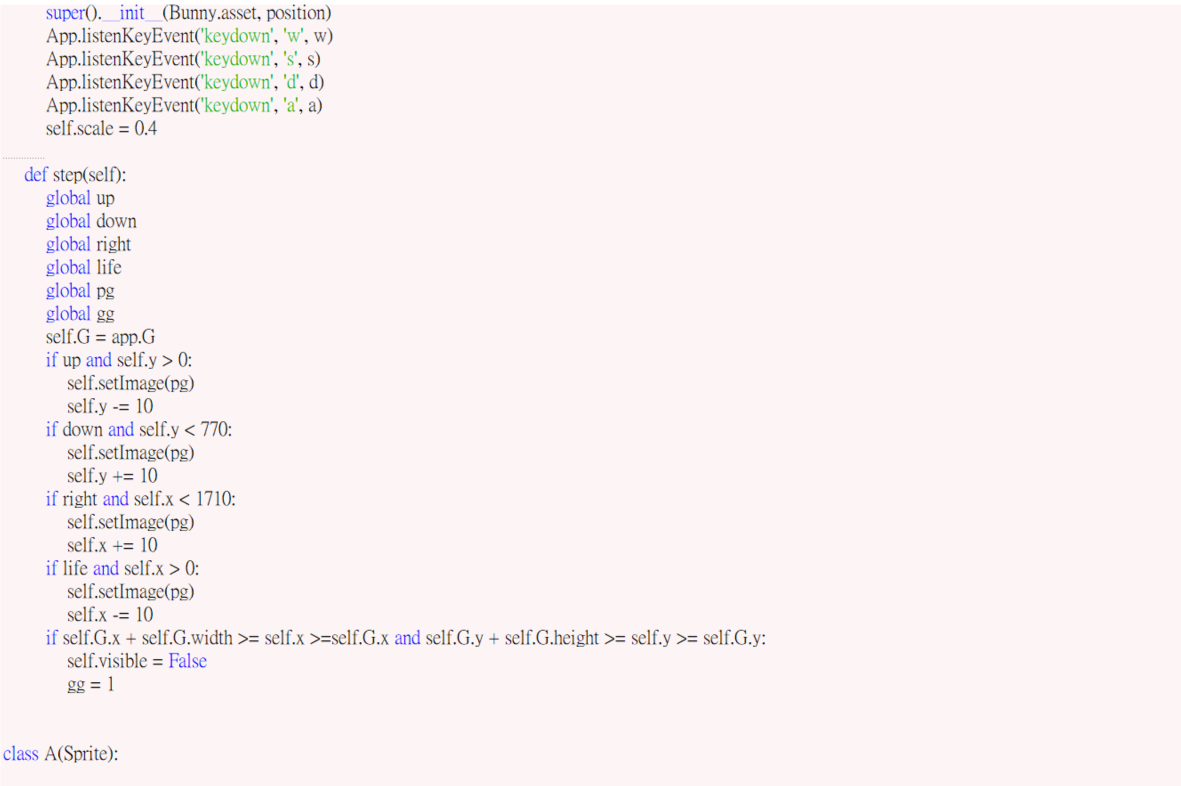
```
@language python
# 導入模組
from ggame import App, ImageAsset, Sprite, MouseEvent, Frame, TextAsset
from ggame import Color, Sound, LineStyle, RectangleAsset, CircleAsset, PolygonAsset, SoundAsset
from random import random, randint
import math
from time import time

up = 0
down = 0
right = 0
left = 0
pg = 0
gg = 0

def w(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 3
    up = 1
    down = 0
    right = 0
    life = 0

def s(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 1
```



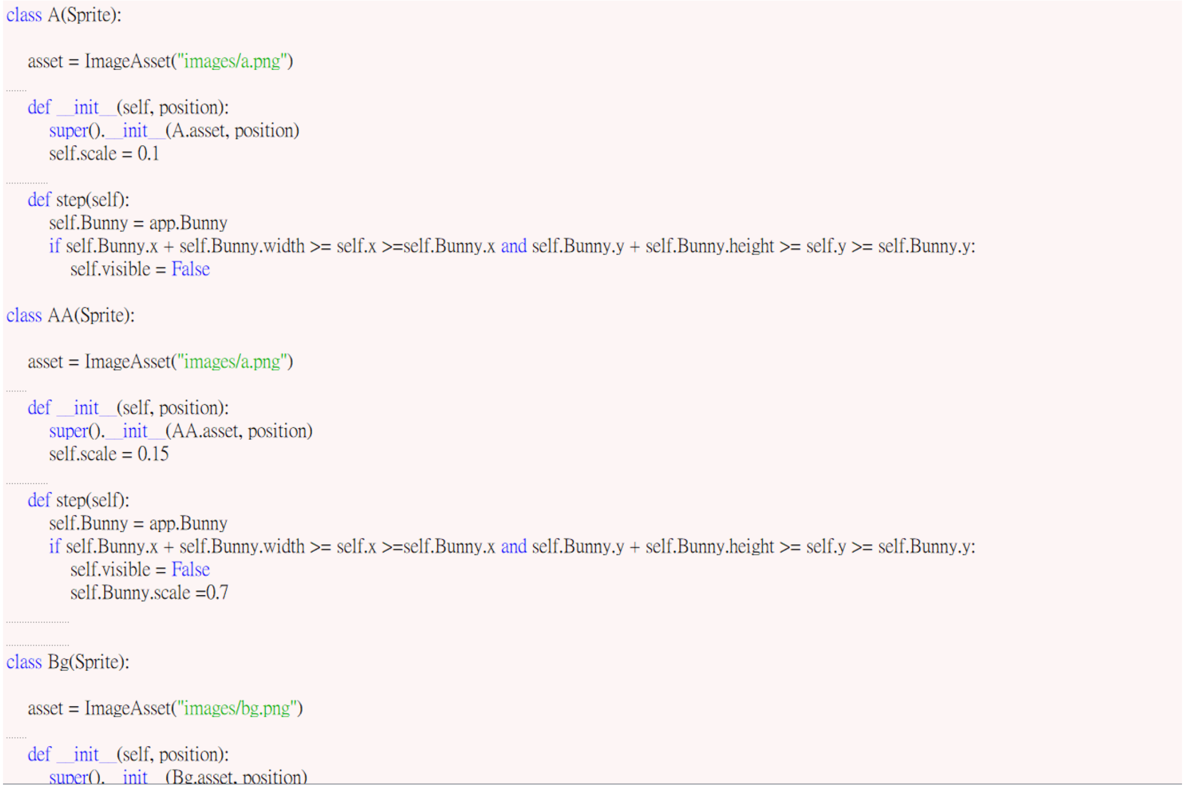
```
    global pg
    pg = 1
    up = 0
    down = 1
    right = 0
    life = 0

def d(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 0
    up = 0
    down = 0
    right = 1
    life = 0

def a(event):
    global up
    global down
    global right
    global life
    global pg
    pg = 2
    up = 0
    down = 0
    right = 0
    life = 1


class GG(Sprite):

    asset = ImageAsset("images/8.png")
```



```
class GG(Sprite):

    asset = ImageAsset("images/8.png")

    def __init__(self, position):
        super().__init__(GG.asset, position)
        self.scale = 1
        self.visible = False

    def step(self):
        if gg:
            self.visible = True


class G(Sprite):

    asset = ImageAsset("images/5.png")

    def __init__(self, position):
        super().__init__(G.asset, position)
        self.scale = 0.5

    def step(self):
        if random() < 0.1:
            self.x += randint(-50,50)
            self.y += randint(-50,50)


class Bunny(Sprite):

    asset = ImageAsset("images/444.png",Frame(0,0,230,208), 4)

    def __init__(self, position):
        super().__init__(Bunny.asset, position)
```



```
        super().__init__(Bunny.asset, position)
        App.listenKeyEvent('keydown', 'w', w)
        App.listenKeyEvent('keydown', 's', s)
        App.listenKeyEvent('keydown', 'd', d)
        App.listenKeyEvent('keydown', 'a', a)
        self.scale = 0.4

    def step(self):
        global up
        global down
        global right
        global life
        global pg
        global gg
        self.G = app.G
        if up and self.y > 0:
            self.setImage(pg)
            self.y -= 10
        if down and self.y < 770:
            self.setImage(pg)
            self.y += 10
        if right and self.x < 1710:
            self.setImage(pg)
            self.x += 10
        if life and self.x > 0:
            self.setImage(pg)
            self.x -= 10
        if self.G.x + self.G.width >= self.x >=self.G.x and self.G.y + self.G.height >= self.y >= self.G.y:
            self.visible = False
            gg = 1


class A(Sprite):
```



```
class A(Sprite):

    asset = ImageAsset("images/a.png")

    def __init__(self, position):
        super().__init__(A.asset, position)
        self.scale = 0.1

    def step(self):
        self.Bunny = app.Bunny
        if self.Bunny.x + self.Bunny.width >= self.x >=self.Bunny.x and self.Bunny.y + self.Bunny.height >= self.y >= self.Bunny.y:
            self.visible = False

class AA(Sprite):

    asset = ImageAsset("images/a.png")

    def __init__(self, position):
        super().__init__(AA.asset, position)
        self.scale = 0.15

    def step(self):
        self.Bunny = app.Bunny
        if self.Bunny.x + self.Bunny.width >= self.x >=self.Bunny.x and self.Bunny.y + self.Bunny.height >= self.y >= self.Bunny.y:
            self.visible = False
            self.Bunny.scale =0.7


class Bg(Sprite):

    asset = ImageAsset("images/bg.png")

    def __init__(self, position):
        super().__init__(Bg.asset, position)
```

```
class Bg(Sprite):

    asset = ImageAsset("images/bg.png")

    def __init__(self, position):
        super().__init__(Bg.asset, position)

    def step(self):
            self.x = 0
            self.y = 0


class DemoApp(App):

    def __init__(self):
        super().__init__()
        Bg((self.width/2, self.height/2))
        for i in range(20):
            A((randint(50,1680),randint(50,750)))
        for c in range(1):
            AA((randint(50,1680),randint(50,750)))
        self.Bunny = Bunny((0,0))
        self.G = G((840, 250))
        self.GG = GG((740, 250))

    def step(self):
        for bunny in self.spritelist:
            bunny.step()


app = DemoApp()
app.run()
```