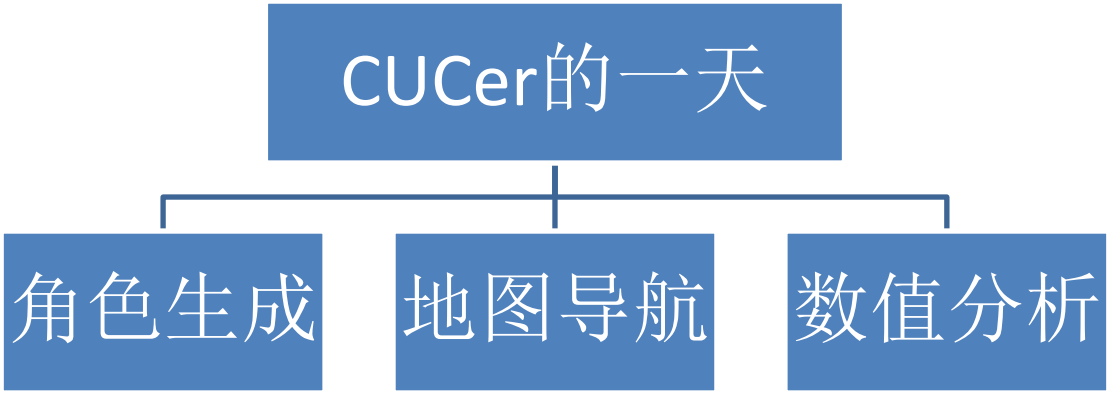
“CUCer 的一天”程序使用说明

本程序为是为了让用户体验 CUCer 的一天，通过选择不同的卡通角色形象和选项，来影响自己的属性数值，从而得到不同的游戏结果。这个程序可以趣味性地增加用户对 CUC 大学生生活的了解和兴趣，也可以为用户提供一定的娱乐性，得到休闲与放松。

1. 功能模块图



2. 模块 1——角色生成系统

2.1 本模块中涉及到的第三方模块：tk，PIL，pygame，pyautogui

2.2模块功能

本模块可以实现角色立绘的生成、角色的选择，角色数值（脑力值、体力值、幸运值）的初始化，和名字的输入、以及与后面地图的链接模块

1) 初始页面设置

创建应用程序类，生成初始页面，用户可以点击“Start!”进入下一界面

```python
# 创建一个应用程序类
class CUCerApp(tk.Tk):
    # 初始化方法
    def __init__(self):
        # 调用父类的初始化方法
        super().__init__()
        self.title("CUCer的一天")
        self.resizable(False, False)
        # 创建一个框架容器
        self.frame = tk.Frame(self)
        # 将框架容器放置在窗口中
        self.frame.pack()
        # 创建第一个界面
        self.create_first_page()
        # 获取屏幕的大小
        sw = self.winfo_screenwidth()  # 屏幕宽度
        sh = self.winfo_screenheight()  # 屏幕高度

        # 计算窗口的位置
        x = (sw - 1456) / 2  # 窗口位于x轴位置
        y = (sh - 816) / 2  # 窗口位于y轴位置

        # 设置窗口的大小和位置
        self.geometry("%dx%d+%d+%d" % (1456, 816, x, y))
```

```python
# 创建第一个界面的方法
def create_first_page(self):
    # 清空框架容器中的所有组件
    self.frame.destroy()
    # 重新创建一个框架容器
    self.frame = tk.Frame(self)
    # 将框架容器放置在窗口中
    self.frame.pack()
    # 创建一个标签组件，显示文本“欢迎来到CUCer的一天！”
    self.label = tk.Label(self.frame, text="欢迎来到CUCer的一天！", font=("SimHei", 24))
    # 将标签组件放置在框架容器中，居中对齐，垂直方向有50像素的间距
    self.label.pack(pady=100)
    # 创建一个按钮组件，显示文本“Start！”
    self.button = tk.Button(self.frame, text="Start！", font=("SimHei", 18), command=self.create_second_page)
    # 将按钮组件放置在框架容器中，居中对齐，垂直方向有50像素的间距
    self.button.pack(pady=50)
```

2) 角色立绘选择

用户可以通过点击按钮选择长发立绘与短发立绘两种角色立绘形象

```
    # 创建第二个界面的方法
    def create_second_page(self):
        # 清空框架容器中的所有组件
        self.frame.destroy()
        # 重新创建一个框架容器
        self.frame = tk.Frame(self)
        # 将框架容器放置在窗口中
        self.frame.pack()
        # 创建一个标签组件，显示文本
        self.label = tk.Label(self.frame, text="请选择你的CUCer形象吧！", font=("SimHei", 24))
        # 将标签组件放置在框架容器中，居中对齐，垂直方向有50像素的间距
        self.label.pack(pady=10)
        # 打开第一张立绘图片文件，并获取图片宽度和高度
        image1 = Image.open("character1.png")
```

```
        # 使用ImageTk.PhotoImage方法来创建图片对象（注意要传入image参数）
        self.image2 = ImageTk.PhotoImage(image=image2)

    # 创建两个按钮组件，用于显示立绘图片（以下代码是修改的）
        self.button1 = tk.Button(self.frame, image=self.image1, command=lambda: self.create_third_page(1))
        self.button2 = tk.Button(self.frame, image=self.image2, command=lambda: self.create_third_page(2))

    # 将按钮组件放置在框架容器中，左右分别对齐，水平方向有20像素的间距
        self.button1.pack(side=tk.LEFT, padx=20)
        self.button2.pack(side=tk.RIGHT, padx=20)
```

3) 角色属性数值初始化

为角色随机生成范围在（1，10）内的幸运值、脑力值、体力值三种属性数值，若用户不满意可以点击左侧立绘图像重新生成相应数值，且可以在输入框内为角色自定义角色姓名。

```
    # 创建一个按钮组件，用于显示用户点击的立绘图片
    self.button = tk.Button(self.frame, image=self.image)
    # 将按钮组件放置在框架容器中，左对齐，水平方向有20像素的间距
    self.button.pack(side=tk.LEFT, padx=20)
    # 给按钮组件绑定点击事件，触发update_attributes方法（这一行是新增的）
    self.button.bind("<Button-1>", self.update_attributes)
    # 创建一个标签组件，显示文本“角色属性”
    self.label = tk.Label(self.frame, text="角色属性", font=("华文中宋", 18))
    # 将标签组件放置在框架容器中，右对齐，垂直方向有20像素的间距
    self.label.pack(side=tk.TOP, pady=20)
      # 创建一个列表组件，用于显示角色的各种属性数值
    self.listbox = tk.Listbox(self.frame, font=("华文中宋", 16))
      # 将列表组件放置在框架容器中，右对齐，垂直方向有20像素的间距
    self.listbox.pack(side=tk.TOP, pady=20)
      # 向列表组件中插入一些示例数据
    self.listbox.insert(tk.END, "身份：CUCer")

    # 随机生成三个属性值，并插入到列表组件中
    self.physical = random.randint(1, 10)
    self.mental = random.randint(1, 10)
    self.luck = random.randint(1, 10)
    self.listbox.insert(tk.END, f"体力值：{self.physical}")
    self.listbox.insert(tk.END, f"脑力值：{self.mental}")
    self.listbox.insert(tk.END, f"幸运值：{self.luck}")
    # 创建一个标签组件，显示文本“角色姓名”
    self.label2 = tk.Label(self.frame, text="角色姓名", font=("华文中宋", 18))
    # 将标签组件放置在框架容器中，右对齐，垂直方向有20像素的间距
    self.label2.pack(side=tk.TOP, pady=20)
    # 创建一个输入框组件，用于用户自定义输入角色姓名
    self.entry = tk.Entry(self.frame, font=("华文中宋", 16))
    # 将输入框组件放置在框架容器中，右对齐，垂直方向有20像素的间距
    self.entry.pack(side=tk.TOP, pady=20)
```

```python
# 创建一个更新属性数值的方法
def update_attributes(self, event):
    # 随机生成三个属性值
    self.physical = random.randint(1, 10)
    self.intelligence = random.randint(1, 10)
    self.charm = random.randint(1, 10)

    # 删除列表组件中原来的属性数值
    self.listbox.delete(1, 4)

    self.listbox.insert(tk.END, f"体力值：{self.physical}")
    self.listbox.insert(tk.END, f"脑力值：{self.intelligence}")
    self.listbox.insert(tk.END, f"幸运值：{self.charm}")
```

4) 与地图模块连接

用户在选择立绘形象时，就确定与后面不同地图支线的链接。

```python
if image_id == 1:
    filename = "character1.png"
    self.role = "role1"
else:
    filename = "character.png"
    self.role = "role2"
```

```python
# 创建一个按钮组件，用于显示用户点击的立绘图片
self.button = tk.Button(self.frame, image=self.image)
# 将按钮组件放置在框架容器中，左对齐，水平方向有20像素的间距
self.button.pack(side=tk.LEFT, padx=20)
# 给按钮组件绑定点击事件，触发update_attributes方法（这一行是新增的）
self.button.bind("<Button-1>", self.update_attributes)
# 创建一个标签组件，显示文本“角色属性”
self.label = tk.Label(self.frame, text="角色属性", font=("华文中宋", 18))
# 将标签组件放置在框架容器中，右对齐，垂直方向有20像素的间距
self.label.pack(side=tk.TOP, pady=20)
  # 创建一个列表组件，用于显示角色的各种属性数值
self.listbox = tk.Listbox(self.frame, font=("华文中宋", 16))
  # 将列表组件放置在框架容器中，右对齐，垂直方向有20像素的间距
self.listbox.pack(side=tk.TOP, pady=20)
  # 向列表组件中插入一些示例数据
self.listbox.insert(tk.END, "身份：CUCer")
```

```python
# 创建一个跳转到地图导航部分的方法
def go_to_map(self):
    # 清空框架容器中的所有组件
    self.frame.destroy()

    # 根据role属性的值，执行不同的代码

    if self.role == "role1":
        def switch_window(cwin, twin):
            # cwin表示当前窗口，twin表示目标窗口
            cwin.withdraw()  # 当前窗口隐藏
            twin.deiconify()  # 目标窗口显示
```

3. 模块 2——地图导航系统

3.1 本模块中涉及到的第三方模块：tkinter，pyautogui，pygame

3.2 模块功能

本模块可以实现地图页面设置、人物移动、新页面设置、返回地图功能。

1) 地图页面设置

主要代码块：

```
def switch_window(cwin, twin):
    # cwin表示当前窗口，twin表示目标窗口
    cwin.withdraw()  # 当前窗口隐藏
    twin.deiconify()  # 目标窗口显示

root = self
root.title("CUCer的一天")
root.resizable(False, False)
# 获取屏幕的大小
sw = root.winfo_screenwidth()  # 屏幕宽度
sh = root.winfo_screenheight()  # 屏幕高度

# 计算窗口的位置
x = (sw - 1456) / 2  # 窗口位于x轴位置
y = (sh - 816) / 2  # 窗口位于y轴位置

# 设置窗口的大小和位置
root.geometry("%dx%d+%d+%d" % (1456, 816, x, y))

# 创建一个画布
canvas = tk.Canvas(root, width=1456, height=816)
canvas.pack()
```

```
# 定义x和y的值为窗口的中心位置（这两行是新增的）
x = 1456 / 2
y = 816 / 2

# 加载人物图片和背景图片
self.background = tk.PhotoImage(file="background.png")
character1 = tk.PhotoImage(file="character1.png")
character1 = character1.subsample(4, 4)

# 在画布上绘制背景图片
canvas.create_image(0, 0, image=self.background, anchor="nw")

# 在画布上绘制人物图片，并设置初始位置
char_id = canvas.create_image(x, y - 80, image=character1)

# 初始化pygame mixer
mixer = pgm.mixer
mixer.init(11025)
music = mixer.music
sound = pgm.mixer.Sound("sound.wav")
```

```
# 创建切换按钮
button1 = tk.Button(root, text="一教", command=lambda: (
move_character1(button1), switch_window(root, teaching1_win), recover_character1(), sound.play(
button1.place(x=700, y=100)
# 创建一个按钮，触发跳转到新页面的函数，并设置按钮位置为目标地点（可以自己调整）
button2 = tk.Button(root, text="图书馆", command=lambda: (
move_character1(button2), switch_window(root, library1_win), recover_character1(), sound.play()
button2.place(x=200, y=600)
button3 = tk.Button(root, text="手球馆", command=lambda: (
move_character1(button3), switch_window(root, gym1_win), recover_character1(), sound.play(),upda
button3.place(x=1000, y=550)
button4 = tk.Button(root, text="宿舍", command=lambda: (
move_character1(button4), switch_window(root, dormitory1_win), recover_character1(), sound.play
button4.place(x=150, y=400)
button5 = tk.Button(root, text="星光", command=lambda: (
move_character1(button5), switch_window(root, dining_hall1_win), recover_character1(), sound.pl
button5.place(x=1100, y=400)
```

2) 人物移动

可以通过获取按钮的位置，向目的地移动，主要代码如下：

```
def move_character1(btn):
    # 获取按钮的位置坐标
    x, y = btn.winfo_x(), btn.winfo_y()
    # 获取人物图片的当前坐标
    cx, cy = canvas.coords(char_id)
    # 计算人物图片需要移动的距离和方向
    dx = x - cx
    dy = y - cy
    # 计算人物图片需要移动的步长和步数，使其沿着一条直线移动
    distance = (dx ** 2 + dy ** 2) ** 0.5  # 两点间的距离公式
    step_size = 10  # 步长，可以自己调整
    steps = int(distance / step_size)  # 步数，取整数部分
    sx = dx / steps  # 每一步在x轴上的位移量
    sy = dy / steps  # 每一步在y轴上的位移量
    # 循环移动人物图片到目标地点的位置，每次移动一步
    for i in range(steps):
        canvas.move(char_id, sx, sy)
        root.update()
        pg.sleep(0.015)  # 暂停0.015秒，可以自己调整
```

3) 新页面设置

```
# 创建教学楼窗口

teaching1_win = tk.Toplevel(root)
teaching1_win.title("一教")
teaching1_win.geometry("%dx%d+%d+%d" % (1456, 816, x, y))
# 加载新的背景图片和音效文件
teaching1_background = tk.PhotoImage(file="classroom1.png")
# 创建一个新的画布
teaching1_canvas = tk.Canvas(teaching1_win, width=1456, height=816)
teaching1_canvas.pack()
# 在新的画布上绘制新的背景图片
teaching1_canvas.create_image(0, 0, image=teaching1_background, anchor="nw")
# 加载对话框图片（可以自己调整）
dialog = tk.PhotoImage(file="dialog.png")
dialog = dialog.subsample(2, 2)
# 在画布上绘制对话框图片，并设置位置（可以自己调整）
teaching1_canvas.create_image(750, 600, image=dialog)
# 在画布上绘制文字，并设置内容、字体、颜色等（可以自己调整）
teaching1_canvas.create_text(650, 565,
                             text="                在肉包子味道的攻击下，" + "\n" + "                你认真学完了一堂高数课。",
                             font=("华文中宋", 16), fill="black")
# 创建一个按钮，触发返回地图页面的函数，并设置按钮位置（可以自己调整）
button = tk.Button(teaching1_win, text="返回地图", command=(lambda: switch_window(teaching1_win, root)))
button.place(x=100, y=50)
teaching1_win.withdraw()  # 初始隐藏窗口1
```

4）返回地图

4. 模块 3——数值分析系统

4.1 本模块中涉及到的第三方模块：Numpy、Matplotlib、tkinter，pyautogui，pygame

4.2 模块功能

本模块可以实现对于三个属性点结果的输出、对数值呈现及趋势进行可视化、对数值的分析与评价。

1) 对于三个点结果的输出

输出体力值、脑力值和幸运值的结果，主要代码块：

```python
def open_new_page():
# 在新页面中添加标签，显示文字内容


    root = tk.Tk()
    root.title("DATA")
    root.geometry("1456x816")
    x = (sw - 1456) / 2  # 窗口位于x轴位置
    y = (sh - 816) / 2  # 窗口位于y轴位置

# 设置窗口的大小和位置
    root.geometry("%dx%d+%d+%d" % (1456, 816, x, y))

        # 创建绘图区域的Figure对象
    figure = plt.figure()


    final_physical = np.random.randint(0, 11)
    final_mental = np.random.randint(0, 11)
    final_luck = np.random.randint(0, 11)
```

```python
# 创建体力值、脑力值和幸运值的标签
physical_label = tk.Label(root, text="体力值：",font=("华文中宋", 24))
physical_label.pack()
physical_display = tk.Label(root, text=final_physical,font=("华文中宋", 30))
physical_display.pack()

mental_label = tk.Label(root, text="脑力值：",font=("华文中宋", 24))
mental_label.pack()
mental_display = tk.Label(root, text=final_mental,font=("华文中宋", 30))
mental_display.pack()

luck_label = tk.Label(root, text="幸运值：",font=("华文中宋", 24))
luck_label.pack()
luck_display = tk.Label(root, text=final_luck,font=("华文中宋", 30))
luck_display.pack()


    # 运行Tkinter主循环
tk.mainloop()
```

2) 对于数值趋势与可视化

对于三个值的分析以及可视化，主要代码如下：

```python
def open_new_page1():
#分析页面
    root = tk.Tk()
    root.title("day")
    root.geometry("1456x1000")



    # 初始化数值
    physical = tk.StringVar()
    mental = tk.StringVar()
    final_luck = tk.StringVar()
    middle_physical = []
    middle_mental = []
    middle_final_luck = []

    def generate_values(location_count):
        final_physical = random.randint(1, 10)
        final_mental = random.randint(1, 10)
        final_final_luck = random.randint(1, 10)
```

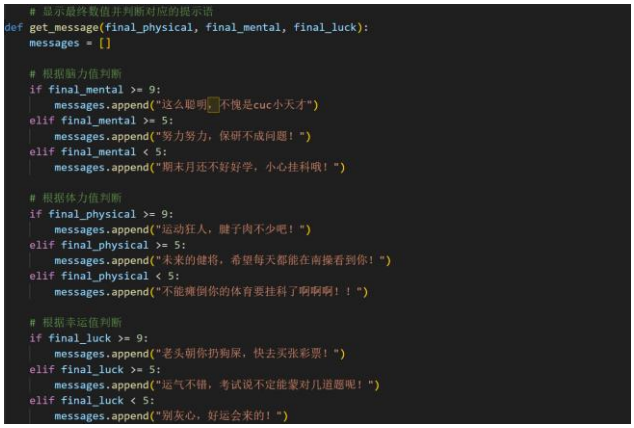
```python
# 创建折线图
x = [0]
y_physical = [int(physical.get())]
y_mental = [int(mental.get())]
y_final_luck = [int(final_luck.get())]


for i in range(location_count):
    x.append(i+1)
    middle_physical.append(random.randint(1, 10))
    middle_mental.append(random.randint(1, 10))
    middle_final_luck.append(random.randint(1, 10))

    y_physical.append(middle_physical[-1])
    y_mental.append(middle_mental[-1])
    y_final_luck.append(middle_final_luck[-1])

# 添加最后一个地方的数值到折线图数据
x.append(location_count+1)
y_physical.append(final_physical)
y_mental.append(final_mental)
y_final_luck.append(final_final_luck)
```

```python
# 绘制折线图
plt.plot(x, y_physical, label='phy')
plt.plot(x, y_mental, label='men')
plt.plot(x, y_final_luck, label='luck')

# 绘制中间数值的标记点
plt.scatter(x[1:-1], middle_physical, c='red', marker='o', label='PMD')
plt.scatter(x[1:-1], middle_mental, c='blue', marker='o', label='BMD')
plt.scatter(x[1:-1], middle_final_luck, c='green', marker='o', label='LMD')

# 设置折线图的标题、标签等
plt.title('DATAING', fontsize=14)
plt.xlabel('NUM', fontsize=20
)
plt.ylabel('DATA', fontsize=20
)
plt.legend()
```

3) 对于数值的分析以及评价

对最后的数值进行分析，并且不同数值作出不同评价。主要代码如下：

```python
def show_message():
    brainpower = int(mental_entry.get())
    physical_power = int(physical_entry.get())
    luck = int(final_luck_entry.get())

    messages = get_message(brainpower, physical_power, luck)

    # 清空显示区域
    text_output.delete('1.0', tk.END)

    # 显示消息
    for message in messages:
        text_output.insert(tk.END, message + "\n")

    # 显示图形
    plt.show()

def submit_values():
    location_count = int(location_entry.get())
    generate_values(location_count)
```



```python
    # 显示最终数值并判断对应的提示语
def get_message(final_physical, final_mental, final_luck):
    messages = []

    # 根据脑力值判断
    if final_mental >= 9:
        messages.append("这么聪明，不愧是cuc小天才")
    elif final_mental >= 5:
        messages.append("努力努力，保研不成问题！")
    elif final_mental < 5:
        messages.append("期末月还不好好学，小心挂科哦！")

    # 根据体力值判断
    if final_physical >= 9:
        messages.append("运动狂人，腱子肉不少吧！")
    elif final_physical >= 5:
        messages.append("未来的健将，希望每天都能在南操看到你！")
    elif final_physical < 5:
        messages.append("不能瘫倒你的体育要挂科了啊啊啊！！")

    # 根据幸运值判断
    if final_luck >= 9:
        messages.append("老头胡你扔狗屎，快去买张彩票！")
    elif final_luck >= 5:
        messages.append("运气不错，考试说不定能蒙对几道题呢！")
    elif final_luck < 5:
        messages.append("别灰心，好运会来的！")
```

5. 本学期学习感受

本小组成员已全部在在线问卷填写过啦！

在线填写地址如下：

https://www.wjx.cn/vm/OTokGcw.aspx#