**《Python与人工智能》**

**实验报告**

`

**学 院： 计算机与信息科学学院**

**专业年级： 2020级计算机科学与技术**

**学生姓名： 杨昌松 学号： 2020051615325**

**授课教师： 罗凌**

**2021 - 2022 学年 第 1 学期**

**2022年 6 月**

**实验报告一**

**一、项目名称：英语单词本**

**二、实验设备和环境**

**硬件环境：window 11**

**软件环境：Visual Studio Code**

**三、项目功能：**

该程序模拟了记单词的功能，能够提供和手机客户端记单词软件相近的功能：

1. 查看单词 查看当前存储的单词元素
2. 添加单词 可以方便的添加自身不熟悉的单词，方便后续记忆
3. 记忆单词 顾名思义，记忆单词
4. 删除单词 可以删除已经能够记住的单词，或者添加错误的单词
5. **功能模块界面设计**

**图形用户界面

描述已自动生成**

图1 主界面



**图2.1 查看单词(没有单词)**

****

**图2.2 查看单词(有单词)**

**图形用户界面, 应用程序

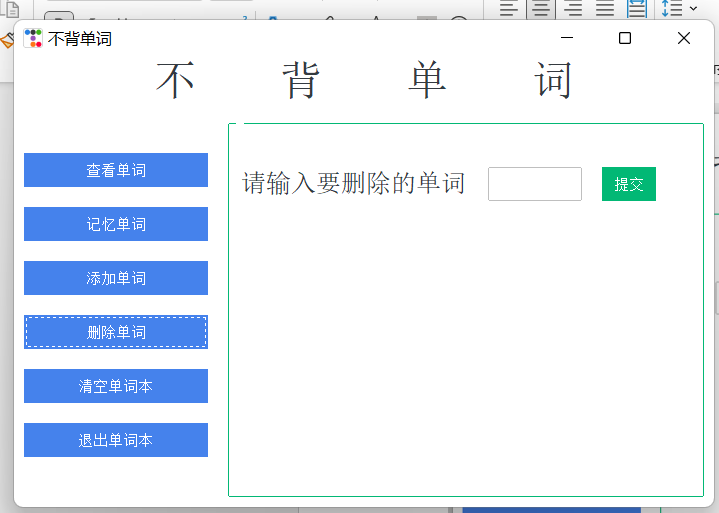
描述已自动生成**

**图3. 记忆单词**

**图形用户界面, 应用程序, 表格, Excel

描述已自动生成**

**图4. 添加单词**

****

**图5. 删除单词**

**五、程序源代码**

1. from distutils.log import info
2. from threading import get\_ident
3. from turtle import bgcolor, width
4. import ttkbootstrap as ttk
5. from ttkbootstrap.constants import \*
6. root = ttk.Window()
7. root.geometry("700x450")
8. root.title("不背单词")
9. word = {} *# 单词*
10. frame\_1 = ttk.Frame(root, bootstyle="info", width=800, height=100)
11. frame\_1.pack(side="top")
12. frame\_2 = ttk.Frame(root, width=220, height=400)
13. frame\_2.pack(side="left")
14. frame\_3 = ttk.Labelframe(root, bootstyle=SUCCESS, width=580, height=400)
15. frame\_3.pack(side='right', padx=10, pady=10)
16. frame\_3.pack\_propagate(0)
17. *# frame\_3\_1  查看单词*
18. frame\_3\_1 = ttk.Frame(frame\_3, width=580, height=100)
19. *# frame\_3\_1.pack()*
20. frame\_3\_1.pack\_propagate(0)
21. label\_3\_1 = ttk.Label(frame\_3\_1, text="Show Menu", font=("宋体", 15))
22. label\_3\_1.pack(side="top", padx=10, pady=10)
23. *# for key, value in word.items():*
24. *#     label\_3\_2 = ttk.Label(frame\_3\_1, text=f"{key}: {value}", font=("微软雅黑", 10))*
25. *#     label\_3\_2.pack(side="left", padx=10, pady=10)*
26. *# frame\_3\_2  记忆单词*
27. frame\_3\_2 = ttk.Frame(frame\_3, width=580, height=400)
28. *# frame\_3\_2.pack()*
29. frame\_3\_2.pack\_propagate(0)
30. import random
31. label\_3\_2\_word\_1 = ttk.StringVar()
32. *# if word == {}:*
33. *# label\_3\_2\_word\_1.set("")*
34. label\_3\_2\_word\_2 = ttk.StringVar()
35. label\_3\_2\_1 = ttk.Label(frame\_3\_2, text="Remember Words", font=("宋体", 15))
36. label\_3\_2\_1.place(x = 150, y = 0)
37. label\_3\_2\_2 = ttk.Label(frame\_3\_2, text="单词", font=("微软雅黑"))
38. label\_3\_2\_2.place(x = 20, y = 50 )
39. label\_3\_2\_3 = ttk.Label(frame\_3\_2, textvariable=label\_3\_2\_word\_1, font=("微软雅黑"), bootstyle="success",width=10)
40. label\_3\_2\_3.place(x = 80, y = 50)
41. label\_3\_2\_4 = ttk.Label(frame\_3\_2, text="翻译", font=("微软雅黑"))
42. label\_3\_2\_4.place(x = 220, y = 50)
43. entry\_3\_2\_1 = ttk.Entry(frame\_3\_2, width=10, textvariable=label\_3\_2\_word\_2)
44. entry\_3\_2\_1.place(x = 280, y =50)
45. frame\_3\_2\_1 = ttk.LabelFrame(frame\_3\_2, width=450, height=250, bootstyle=SUCCESS)
46. frame\_3\_2\_1.place(x= 20, y = 100)
47. def remember\_word():
48. one = label\_3\_2\_word\_1.get()  *# key*
49. two = str(label\_3\_2\_word\_2.get())
50. print(one)
51. print(two)
52. if word[one] == two:
53. label\_3\_2\_5 = ttk.Label(frame\_3\_2\_1, text=f"{one} : {two} 记忆成功", font=("微软雅黑", 10))
54. label\_3\_2\_5.pack(padx=10, pady=10)
55. a,b = random.choice(list(word.items()))
56. label\_3\_2\_word\_1.set(a)
57. label\_3\_2\_word\_2.set("")
58. else:
59. label\_3\_2\_5 = ttk.Label(frame\_3\_2\_1, text=f"{one} : {two} 记忆错误", font=("微软雅黑", 10))
60. label\_3\_2\_5.pack(padx=10, pady=10)
61. button\_3\_2\_1 = ttk.Button(frame\_3\_2, text="提交", bootstyle="success", command=remember\_word)
62. button\_3\_2\_1.place(x = 400, y = 50)
63. *# frame\_3\_2\_1 = ttk.LabelFrame(frame\_3\_2, width=450, height=250, bootstyle=SUCCESS)*
64. *# frame\_3\_2\_1.place(x= 20, y = 100)*
65. *# label\_3\_2\_1\_1 = ttk.Label(frame\_3\_2\_1, text="展示框")*
66. *# label\_3\_2\_1\_1.pack()*
67. *# frame\_3\_3  添加单词*
68. frame\_3\_3 = ttk.Frame(frame\_3, width=580, height=400)
69. *# frame\_3\_3.pack()*
70. frame\_3\_3.pack\_propagate(0)
71. label\_3\_3\_1 = ttk.Label(frame\_3\_3, text="Add Words", font=("宋体", 15))
72. label\_3\_3\_1.pack(side="top", padx=10, pady=10)
73. label\_word\_1 = ttk.StringVar()
74. label\_word\_2 = ttk.StringVar()
75. label\_3\_3\_2 = ttk.Label(frame\_3\_3, text="单词", font=("微软雅黑"))
76. label\_3\_3\_2.pack(side="left",padx=5, pady=10, anchor='nw')
77. label\_3\_3\_3 = ttk.Entry(frame\_3\_3, width=10, textvariable=label\_word\_1)
78. label\_3\_3\_3.pack(side="left", padx=10, pady=10, anchor='nw')
79. label\_3\_3\_4 = ttk.Label(frame\_3\_3, text="翻译", font=("微软雅黑"))
80. label\_3\_3\_4.pack(side="left",padx= 10, pady=10, anchor='nw')
81. entry\_3\_3\_1 = ttk.Entry(frame\_3\_3, width=10, textvariable=label\_word\_2)
82. entry\_3\_3\_1.pack(side="left", padx=10, pady=10, anchor='nw')
83. def submit():
84. if label\_word\_1.get() != "":
85. word[label\_word\_1.get()] = label\_word\_2.get()
87. label\_word\_1.set("")
88. label\_word\_2.set("")
89. *# label\_3\_3\_5 = ttk.Label(frame\_3\_3, text=f"{label\_3\_3\_3.get()} : {entry\_3\_3\_1.get()} 添加成功", font=("微软雅黑", 10))*
90. *# label\_3\_3\_5.pack(padx=10, pady=10)*
91. button\_3\_3\_1 = ttk.Button(frame\_3\_3, text="提交", bootstyle="success", command=submit)
92. button\_3\_3\_1.pack(side="right", padx=10, pady=10, anchor='nw')
93. *# frame\_3\_3\_1 = ttk.LabelFrame(frame\_3\_3, width=580, height=50)*
94. *# frame\_3\_3\_1.pack(side="bottom", padx=10, pady=10)*
95. *# frame\_3\_3\_1.pack\_propagate(0)*
96. *# label\_3\_3\_1\_1 = ttk.Label(frame\_3\_3\_1)*
97. *# label\_3\_3\_1\_1.pack()*
98. *# frame\_3\_4 清除单词*
99. frame\_3\_4 = ttk.Frame(frame\_3, width=580, height=100)
100. *# frame\_3\_4.pack()*
101. frame\_3\_4.pack\_propagate(0)
102. *## 创建label*
103. label\_3\_4\_word\_1 = ttk.StringVar()
104. label\_3\_4\_1 = ttk.Label(frame\_3\_4, text="请输入要删除的单词", font=("宋体", 15))
105. label\_3\_4\_1.pack(side="left", padx=10, pady=10)
106. entry\_3\_4\_1 = ttk.Entry(frame\_3\_4, width=10, textvariable=label\_3\_4\_word\_1)
107. entry\_3\_4\_1.pack(side="left", padx=10, pady=10)
108. def delete\_word():
109. one = label\_3\_4\_word\_1.get()
110. del word[one]
112. button\_3\_4\_1 = ttk.Button(frame\_3\_4, text="提交", bootstyle="success", command=delete\_word)
113. button\_3\_4\_1.pack(side="left", padx=10, pady=10)
114. *# frame\_3\_5 清空单词*
115. frame\_3\_5 = ttk.Frame(frame\_3, width=580, height=100)
116. frame\_3\_5.pack\_propagate(0)
117. label\_3\_5\_1 = ttk.Label(frame\_3\_5, text="单词已清空", font=("宋体", 15))
118. label\_3\_5\_1.pack(side="left", padx=10, pady=10)
119. label\_3\_2 = ttk.Label(frame\_3\_1, font=("微软雅黑", 10))
120. *# function*
121. def show():
122. frame\_3\_1.pack()
123. for key, value in word.items():
124. *# label\_3\_2 = ttk.Label(frame\_3\_1, text=f"{key}: {value}", font=("微软雅黑", 10))*
125. label\_3\_2['text'] = f"{key}: {value}"
126. label\_3\_2.pack(side="left", padx=10, pady=10)
127. *# if word == {}:*
128. *#     label\_3\_2 = ttk.Label(frame\_3\_1, text="暂无单词", font=("微软雅黑", 10))*
129. *#     label\_3\_2.pack(side="left", padx=10, pady=10)*
130. frame\_3\_2.pack\_forget()
131. frame\_3\_3.pack\_forget()
132. frame\_3\_4.pack\_forget()
133. frame\_3\_5.pack\_forget()
134. def remember():
135. frame\_3\_1.pack\_forget()
136. frame\_3\_2.pack()
137. a,b = random.choice(list(word.items()))
138. print(a, b)
139. *# remember\_word()*
140. if word != {}:
141. label\_3\_2\_word\_1.set(a)
142. else:
143. label\_3\_2\_word\_1.set("暂无单词")
144. frame\_3\_3.pack\_forget()
145. frame\_3\_4.pack\_forget()
146. frame\_3\_5.pack\_forget()
147. def add():
148. frame\_3\_1.pack\_forget()
149. frame\_3\_2.pack\_forget()
150. frame\_3\_3.pack()
151. frame\_3\_4.pack\_forget()
152. frame\_3\_5.pack\_forget()
153. def delete():
154. frame\_3\_1.pack\_forget()
155. label\_3\_2["text"] = ""
156. frame\_3\_2.pack\_forget()
157. frame\_3\_3.pack\_forget()
158. frame\_3\_4.pack()
159. frame\_3\_5.pack\_forget()
160. def clear():
161. frame\_3\_1.pack\_forget()
162. frame\_3\_2.pack\_forget()
163. frame\_3\_3.pack\_forget()
164. frame\_3\_4.pack\_forget()
165. word.clear()
166. frame\_3\_5.pack()
167. label\_3\_2['text'] = "单词已清空"
168. *# frame\_1*
169. label\_1 = ttk.Label(frame\_1, text="不    背    单    词", font=("宋体",25))
170. label\_1.pack()
171. *# frame\_2*
172. button\_1 = ttk.Button(frame\_2, text='查看单词', width=20, command=show)
173. button\_2 = ttk.Button(frame\_2, text='记忆单词', width=20, command=remember)
174. button\_3 = ttk.Button(frame\_2, text='添加单词', width=20, command=add)
175. button\_4 = ttk.Button(frame\_2, text='删除单词', width=20, command=delete)
176. button\_5 = ttk.Button(frame\_2, text='清空单词本', width=20, command=clear)
177. button\_6 = ttk.Button(frame\_2, text='退出单词本', width=20, command=quit)
178. button\_1.pack(padx=10, pady=10)
179. button\_2.pack(padx=10, pady=10)
180. button\_3.pack(padx=10, pady=10)
181. button\_4.pack(padx=10, pady=10)
182. button\_5.pack(padx=10, pady=10)
183. button\_6.pack(padx=10, pady=10)
184. *# frame\_3*
185. root.mainloop()
186. **实验中遇到了哪些难点？如何解决？**

**难点：**

**有关tkinter的布局问题，我不喜欢使用place 进行绝对布局，使用Pack布局的时候，组件的位置到处飘逸，无法达到想要的效果，**

**如何解决：**

**所以我选择了给一个窗体套上很多层面板，每个面面板里面放置不同组件，然后组合，从而达到一个比较良好的效果**

**难点：**

**组件的美化，原生的组件在我看来相当丑陋，不堪大用，**

**如何解决：**

**套用了前人写好了的框架ttkbootstrap 能够实现一个不错的效果**

1. **实验体会**

**Tkinter 的使用过程不是那么得心应手，有很大知识点都是边搜边写，写完之后就忘记了，没有真正积累下来，还有就是关于tkinter的布局，用的实在是不是很得心应手，关于布局是种感觉在实际使用的过程中比较复杂，但是老师讲的或者是网上的课程都讲得很简单，当组件多起来，复杂起来，就不能很好的控制每一个组件的位置和样式了**

**实验人： 杨昌松**

**2022年 6 月 15 日**