Trò chơi cá cược đua xe Schneller

**TÀI LIỆU CODING**

Tài liệu coding trò chơi cá cược đua xe Schneller

Phiên bản 1.0

Ngày tạo: 24/12/2022

Contents

[**1.** **Coding Workflow** 2](#_Toc124168853)

[**2.** **Library** 2](#_Toc124168854)

[**3.** **Các Toolboxs được sử dụng** 2](#_Toc124168855)

[**4.** **File Organization** 2](#_Toc124168856)

[**5.** **Naming Conventions** 2](#_Toc124168857)

[**6.** **Indentation, white space** 2](#_Toc124168858)

[**7.** **Comments, declarations** 2](#_Toc124168859)

[**8.** **Tab and Indent** 2](#_Toc124168860)

[**9.** **Space** 2](#_Toc124168861)

# **Coding Workflow**

**Plan**

**Code Library**

**Modules**

**Code Functional**

**Modules**

**Integrate**

**Components**

**Create System**

**Description**

**Code Functional**

**Modules**

# **Library**

Các thư viện ở đoạn trên phục vụ cho mô hình AI, các thư viện ở đoạn dưới phục vụ cho interface và source code của game.

from keras.preprocessing.text import Tokenizer

from string import digits

from tensorflow import keras

from tensorflow.keras.preprocessing.sequence import pad\_sequences

import numpy as np

import pandas as pd

import tensorflow as tf

from minigame1 import main as mn1

from minigame2 import main as mn2

from PIL import ImageTk, Image

from pygame import mixer

from pygame.constants import KEYDOWN, MOUSEBUTTONDOWN, QUIT

from pygame.constants import MOUSEBUTTONDOWN

from pygame.locals import \*

from testAi import responseChat

from tkinter import \*

from turtle import width

import function, ioexcel, try\_nam, maingameplay

import pygame, sys

import random, click

import user\_account, newdraw, math

# **Các Toolboxs được sử dụng**

* Button
* GridView (self-made)
* Panel
* Picture Box
* Textbox

# **File Organization**

* data
* docs
* fonts
* minigame1
* minigame2
* sound effects
* AI\_final.h5
* click.py
* credits.txt
* function.py
* gitignore
* ioexcel.py
* lichsuchoi.py
* main\_minigame1.py
* main\_minigame2.py
* maingameplay.py
* newdraw.py
* source.py
* testAI.py
* try\_nam.py
* user\_account.py

# **Naming Conventions**

* Các file Python đặt tên dưới dạng ‘ten\_file.py’:

Ex: try\_nam.py

* Tên biến 1 từ đặt theo dạng ‘name’:

curTime = 0

* Tên biến nhiều hơn 1 từ đặt theo dạng ‘longName:

iCountDown = 0

# **Indentation, white space**

* Function:

def doSomething(*target*):

    # Return Something

* If:

if (0 > 1):

    # Do something

* For:

for i in range (0,9):

    # Do something

# **Comments, declarations**

* Comment: # Some Comments

# Do something

* Declarations: <varible name> = <value>

char\_name = ''

# **Tab and Indent**

* 4 dấu cách = 1 indent (qui tắc default của Visual Studio Code)
* Độ dài mỗi dòng: không quá dài, < 150 kí tự
* Lượng code trên mỗi file tối đa 1000 dòng.

# **Space**

* Sử dụng dòng trống để tăng khả năng đọc code
* Sử dụng 2 dòng trống:

+ Giữa các class với nhau

+ Giữa hai mục lớn trên cùng một file

* Sử dụng 1 dòng trống:

+ Giữa các mục lớn trong một function

+ Giữa hai functions

+ Tăng tính dễ đọc cho các code bị rối

# **Code Planning**

* Lập kế hoạch chuẩn bị cho quá trình phát triển phần mềm
* Các bước:
  + Lên check list các mục trong tài liệu thiết kế, bao gồm: Đăng nhập, Trang chủ, Chọn đường đua, Chọn nhân vật, Chọn cá cược, Màn chơi chính, Màn hình ăn mừng, Chọn màn chơi phụ, Màn chơi phụ #1, Màn chơi phụ #2, Cửa hàng, Tùy chỉnh, Lịch sử, Hướng dẫn và Nguồn tham khảo.
  + Xác định nguồn lực cần dùng:

+ 1 PM

+ 1 BA

+ 1 Designer

+ 2 Testers

+ 3 Developers

* + Mục tiêu:

+ Tạo ra một sản phẩm hoàn thiện, không gặp lỗi khi chạy

+ Đáng ứng hết tất cả các yêu cầu của khách hàng

+ Có nhiều hiệu ứng và interface dễ sử dụng

* + Nhiệm vụ:

+ Mỗi tuần họp 1 lần, chia sẻ tiến độ công việc

+ Đảm bảo mỗi thành viên hoàn thành phần công việc được giao

# **Classes & Functions**

* Vì các bạn coder vẫn còn đang trong quá trình học, chưa đến phần lập trình hướng đối tượng và chỉ vừa học xong struct trong cpp, nên team hạn chế dùng class cho các file chính mà tạo ra nhiều functions giao tiếp với nhau. (Trừ phần user interface hoặc các phần riêng có template code theo nên dùng class khá nhiều)
* Cách làm này đòi hỏi người dùng trình bày và comments rõ ràng các biến để tránh lỗi.
* Các functions làm nhiều việc nhỏ cho mỗi file, vd: try\_nam.py có resize, run, runAnimation, …