شرح لعبة PIKA GO

settings.py

اول ملف في اللعبة وكان غرضه انه يكون فيه الثوابت بتاعت اللعبة و استيراد المكتبات
فيها طول وعرض شاشة اللعبة و ال tile size دا اللي محددينه في برنامج Tiled و دا بيحسب كل بلوكاية فيها قد ايه , و معدل الفريم دا علشان نثبت عدد الفريمات اللي بتاخدها اللعبة في الثانية لان بيختلف الفريمات من سرعة جهاز للتاني لو محددناش ف بالتالي اننا نحدد بيخلي اللعبة ثابت على اي جهاز و كمان علشان نقدر نربط المتغيرات بالوقت, و join استوردناها علشان لو بنتعامل ب operating system مختلف و فيه بيستخدم لل path "/" أو "\"
فعلشان نبعد عن استخدامها ويحصل مشكلة في التشغيل بنستخدم join , و walk بيقدر يمشي بين ال folders و دا هنحتاجه علشان يمشي في ملفات اللاعب , و load_pygame دا علشان نحمل الملف .tmx للبرنامج

support.py

دا ملف بيساعد على اننا نحمل ال folders , files , audio

import_image

لو هيحمل صورة بتديله اسم كل subfolder until the image و في حالة ان الصورة محتاجة يتعملها تغيير في المقاس بنستخدم transform.scale
و convert_alpha دي بتستخدم علشان لو الصورة مش بخلفية ميحطهاش (image transparent)

import_folder

بيعامل هنا ان ال frames عبارة عن ليست و بنمشي بيها باسمها و بيكون متسلسل من رقم 0, "walk"
بترجع ال parent folder, subfolder, file name و بيرجع ال file name ك list , بنلف نشوف المسار بالكامل باسم ال parent folder , file name و بنحمل الصورة بيها و نضيف للفريمات الصور اللي بيلاقيها في الفولدر

audio_importer

نفس فكرة ال فولدر بس هو بيشوف ال audio و يحفظها ك dict , و ال key هو اسم ال audio

settime.py

ال class Timer معمول علشان يحسب الوقت للكائنات
activation بيحسب المدة من وقت ما يحصله -> duration
بداية العداد -> start time
اللي بيشوف بيها اذا كانت متفعلة ام لا -> active
دي لو هنفعل فانكشن معين مع الوقت -> function
لو هيفضل يتكرر -> repeat
دي بتفعل الوقت او ما بنكريت الكائن -> auto start

والفانكشنس اللي موجودة :

بيخلينا نتشيك بالكائن نفسه اذا كان اكتف ام لا -> bool
بيفعل التايمر و يخلي العداد يبدأ يعد -> activate
بيرجع البداية بصفر و يوقف العد -> deactivate
دا اللي بيتحسب في اللعبة و يشوف الحالة طول ما اللعبة شغالة -> update

groups.py

دا بينفذ فكرة الكاميرا ان لما اللاعب بيتحرك ال tiled تيجيله بمعنى ان لما اللاعب يتحرك يمين ال tiled يتحرك شمال بحيث اللاعب يفضل متوسط الشاشة و يقدر يتحرك
فدا بنفذه على كل ال objects اللي عاويزنها تبان له و تجيله
بياخد ال surface بتاعت الحاجات اللي هتتحرك وال offset دا علشان نقول هيحركها بمقدار قد ايه
دالة ال draw هي override للدالة اللي موجود في ال group علشان نعدل عليها في الرسم و دي اللي هتقولنا هيبان فين في الشاشة , هيبان استخدامها اكتر في ال main.py

sprites.py

دا الاساس لكل حاجة حرفيا في اللعبة, الجزء المسؤول عن خصائص كل حاجة كل من الاعداء و والورد واللاعب و حتى خصائص ضرب النار, اي كائن هنعمله في اللعبة لازم يكون مر من هنا
هنقسم كلامنا عن كل كلاس واهم الخصائص اللي فيه :

Sprite

دا كلاس هنعمله علشان بدل ما في كل مرة نعرف ال image, rect بتاع اي كائن فهنعمله مرة ونورث منه الخصائص دي طبعا ال image بيقولنا الصورة وال rect بيدينا موضعها
و هو بيورث من pygame.sprite.Sprite و دا علشان فيها خصائص محتاجنها واكييد بينظم المحتوى فبيوفر دوال زي update, self.kill , collision

Bullet

كل خصائصها ان بنغير اتجاها بناء على اتجاه اللاعب (هنشوف بعدين ازاي اتجاه اللاعب بيغيره) و بنحدد سرعتها و اتجاه حركتها , دالة update اللي هتخليها تغير من مكانها قد ايه خلال ثانية (dt)

> ملحوظات على الهامش: افتكر ان update بتتحدث في قلب اللعبة كل ثانية.

Fire

زودنا لها خصائص زي مكانها بالظبط من اللاعب و الوقت اللي هتشتغل فيه جزء من الثانية لانه بيشتغل بس لحظة ضرب النار , بس ضيفنا حاجة في الاخر كدا في دالة ال update, ان لو كان اتجاها مش نفس اتجاه اللاب يتمسح self.kill ودا علشان لما نقلب مرة واحدة بسرعة واحنا بنضرب نار ميكونش النار في ناحية واللاعب في ناحية لحظتها (enhancement) مش اكتر

AnimatedSprite

الجزء المسؤول عن ال frames فبياخد ال index للرسمة و يحطها كصورة و تتغير خلال dt في سرعتها و دا هنحتاجه للاعب والاعداء , اللاعب هنحط له حالات سواء الحركة يمين او شمال و اللاعداء طول ما هم موجودين

Enemy

ضيفنا فيها خصائص للدود والخفاش من حيث الحركة والموت و لها فانكشن لحظة التدمير بحيث انه يوقف عن الحركة ويغير لونه و يموت

> pygame.mask
>
> دي بنستخدمها علشان نتاكد من ملامسة الكائن مش حواليه بمعنى لو اتعاملنا ب rect فهو حرفيا هياخد المستطيل بتاعه ويتعامل معاه انما mask بيحدد حواف الكائن نفسه

ال constraint دا بنحدد بيه الحدود اللازمة لكل عدو في الدودة هيكون في ليميت مستطيل متتحدد في tiled و الخفاش لحد ما يوصل لنقطة برا فريم اللعبة

Bat

دا اول عدو هيورث من كلاس ال Enemy , استخدمنا فيها sin علشان يتحرك بشكل موجي ف ثبتنا اتجاه x

يفضل ماشي ناحية الشمال و الحركة الافيقة عشوائية موجية
ال amplitude دي سعة الموجة بحيث نقول هيطلع و ينزل لحد فين و ال frequency لتذبذب الموجة
Worm
دي ضفنا لها حدود للمسطيل اللي يتحرك فيه و لما يوصل لحد اخره بيقلب اتجاهه و حالته من حالات ال enemy عموما
Player
دا اكبر كلاس موجود في اللعبة لان خصائصه كتير فهو عنده خصائص لضرب النار و خسارة صحة و الحركة بالانيميشن و الاصطدام فممكن نعرف الدوال الخاصة بيه الاول علشان نعرف محتاجين فيه ايه
ال input , دا عندنا اربعة اتنين بحالات واتنين لا
الاتنين اللي تقدر تستخدمهم بطلاقة right, left ال jump له حالة انه لازم يكون على الارض علشان ميفضلش ينط و ننسى مفهوم الجاذبية اولا و علشان هيوسع حدوده في الحركة خارج المطلوب ثانيا
فبكدا احنا محتاجين يكون عندنا اتجاهه direction علشان نشوف حركته ونخليه يروح مكانها ف الاتجاه عبارة عن متجه و بنحدده -1, 1 لاتجاه الحركة و علشان نحدد اذا كان بيتحرك بانهي انيميشن , يمين ام شمال فبالتالي هنحاج برضو نعرف faceing_right: bool, هيحدد لنا الانيميشن و برضو نعرف منه اتجاه النار اللي كنا حاطينه و نرجع لل jump, هنديله gravity علشان لما ينط يهبط تاني على الارض فلازم علشان يعرف ينط يتشيك اذا كان على الارض on_floor, و حددنا وقت لاطلاق النار علشان ميفضلش يضرب نار كتير جدا فمفس اللحظة ودا مش منطقي فحددنا يكون نص ثانية و اتجاه الرصاصة حسب اتجاه اللاعب, و نفعل ضرب النار
و ضيفنا ال collision مع البلوكات في ال tiled ان لو حصل collision يخلي اللاعب مكانه بالاضافة لشرطية انه على الارض تتحقق لو كان ملامس ارضية اللاعب
ال animate بيتشك برضو اتجاهه ويفضل يعيد الفريمات بشكل loopy
بعد كدا دالتين علشان يقلل الصحة و يتاكد انه حصله ضرر فيقلل من الصحة و يخليه ميتأذاش لمدة 0.6 ثانية
ولازم نعمل ابديت لكل حاجة بتحصل جوا اللعبة
Heart
الكلاس دا بيرسم القلوب و بيغير من الفريم على حسب الحالة بتاعتها ودا هنشوفه اكتر في فايل ال heartmanage
Flower
دا بيخاد الورد و يكبره مريتن لانه صغير عن المطلوب و بيحطها في مكانها بتحديد لها
Pika
البنوتة بتاعتنا اللي هنجيبلها الورد بيتاكد من حالتها وعلى اساها بيدينا شكلها و بيحسب الفترة اللي هي حزينة فيها و سعيدة

heartmanage.py

دا هياخد فريمات القلوب و يحطهم جمب بعض ويتاكد من حالة اللاعب وعلى اساسها هيحدث شكل القلب

main.py

دا قلب اللعبة اللي هيبدأ من عندها التشغيل وهنعمل كلاس للعبة ننفذ فيها كل حاجة
بنعرض الشاشة و نحدد متغير للوب بتاع اللعبة علشان نعرف هنقفلها ازاي و علشان لما اللعبة تنتهي سواء بالمكسب او الخسارة و بعد كدا نعمل لكل مجموعة اي حاجة مش هنتعامل بيها لوحدها sprite.Group, و نعمل فانكشن تستورد كل حاجة عملناها من ملف assets و ال score مربوط بالورد اللي جمعه واللي هنستخدمه بعدين علشان نعرف حالة البيكوكة البنت و ال setup متعلق باللي هنجيبه من ال tmx_map و

هنربطه بالاماكن اللي فيه و ممكن اكبر فانكشن عندنا هي ال collision علشان عندنا اربع انواع غير اللي اللي اتذكرت مع البلوكات, الاول الرصاصة مع الاعداء ودا اللي هيخلي العدو يموت و بعدين اللاعب مع العدو ودا هيقلل من صحة اللاعب (و برضو ضيفنا ان لو وقع يقلل من الصحة) و اللاعب مع الورد فبيجمعه و بيزود الاسكور و في الاخر البيكوكة و على حسب الاسكور زي ما قولنا يحدد الحالة و في حالة ان الالصحة وصلت لصفر بيجمد الشاشة و يقوله انه خسر و تنتهي اللعبة , في داالة ال run هو اللي بنفع كل حاجة عملناها و بنعرضها فيها.