Notre projet d'Informatique et Science du Numérique (2018-2019)

par Hernandez Clara et Blanchet Alan

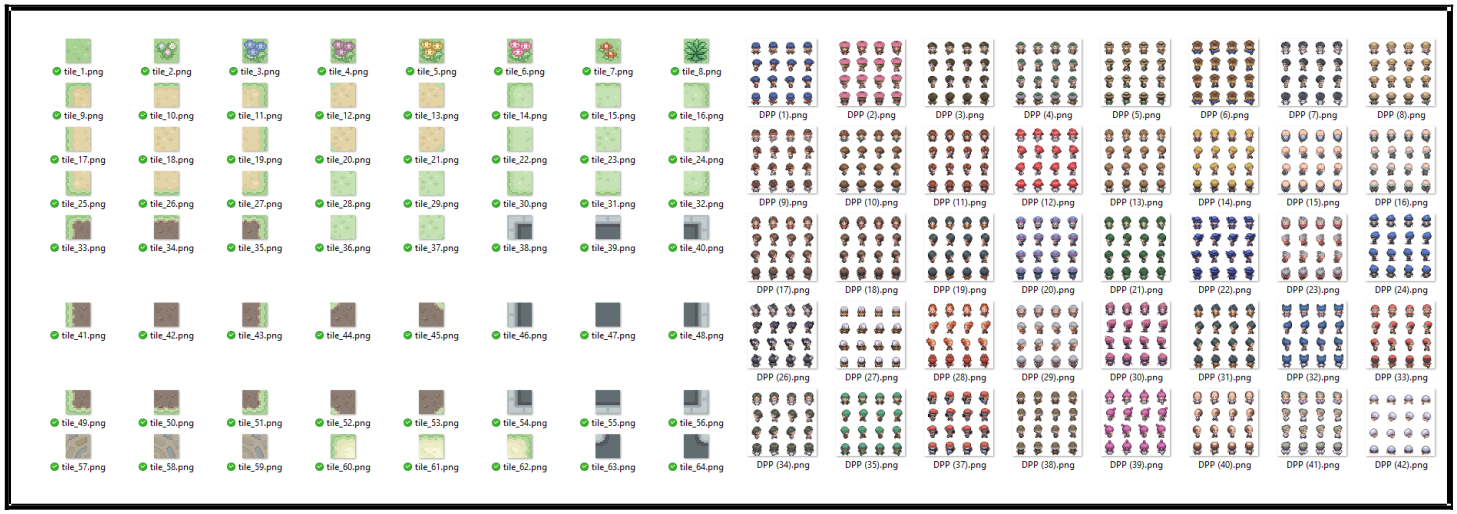
1 - Introduction :

Pourquoi avoir choisi de faire ce jeu ? Et bien pourquoi pas ? Nous sommes tous deux fans de la série pokemon (aussi bien en tant que jeux qu'en tant que dessin animé) et nous nous sommes donc appuyés sur ces connaissances afin de créer notre jeu.

) A quel(s) support(s) avons-nous eu recours :

Nous nous sommes appuyés sur la série de jeu (Nintendo DS) comprenant « Pokémon Version Diamant, Perle et Platine » mais également d'autres Pokémons encore plus vieux comme « Pokémon Version Vert Feuille » ou « Pokémon Version Rouge Feu ».

Nous avons récupéré les images permettant de créer les cartes à partir de « Pokémon Version Vert Feuille » et « Pokémon Version Rouge Feu ». Et pour ce qui est des personnages nous les avons récupérés à partir de la Série « Diamant », « Perle » et « Platine ».



Pour parler de support en termes de programmation, nous avons utilisé le module « pygame » en plus de quelques autres modules déjà disponibles en installant python. On a également eu recours à des logiciels pour faire des petites musiques (jeu et combat)

) A quoi ressemble le jeu dans son ensemble ? :

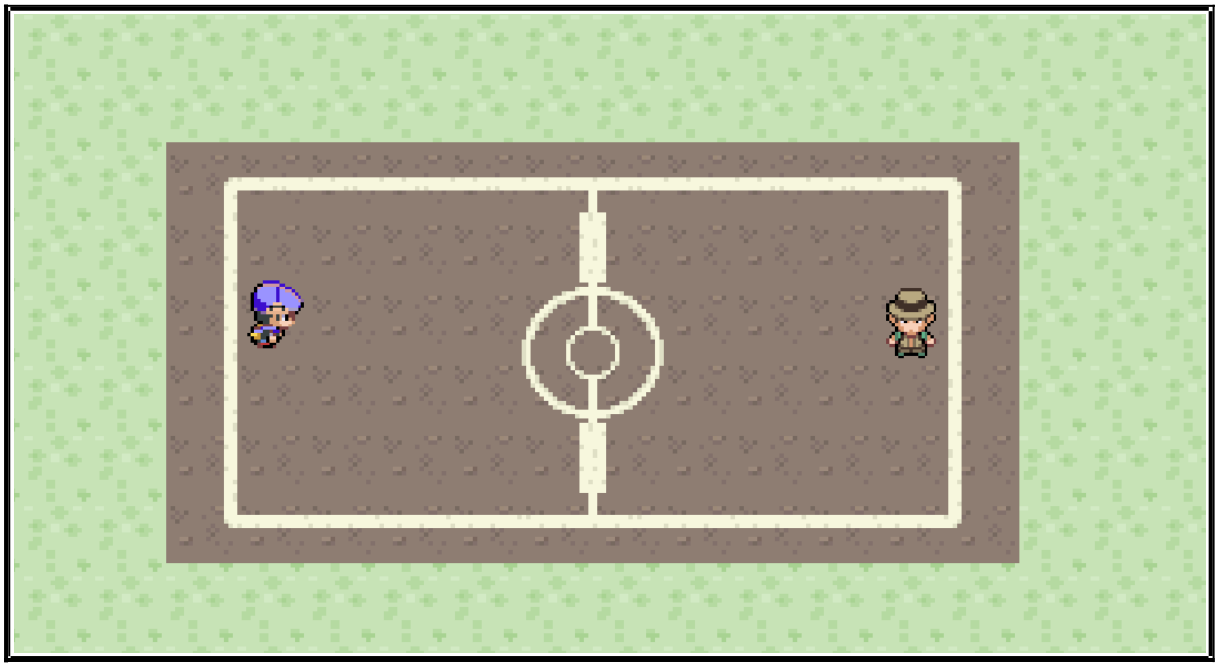
Le jeu est composé de plusieurs images insérées à des endroits bien précis, ces images forment une carte. Une image peut être caractéristique d'une simple voie à

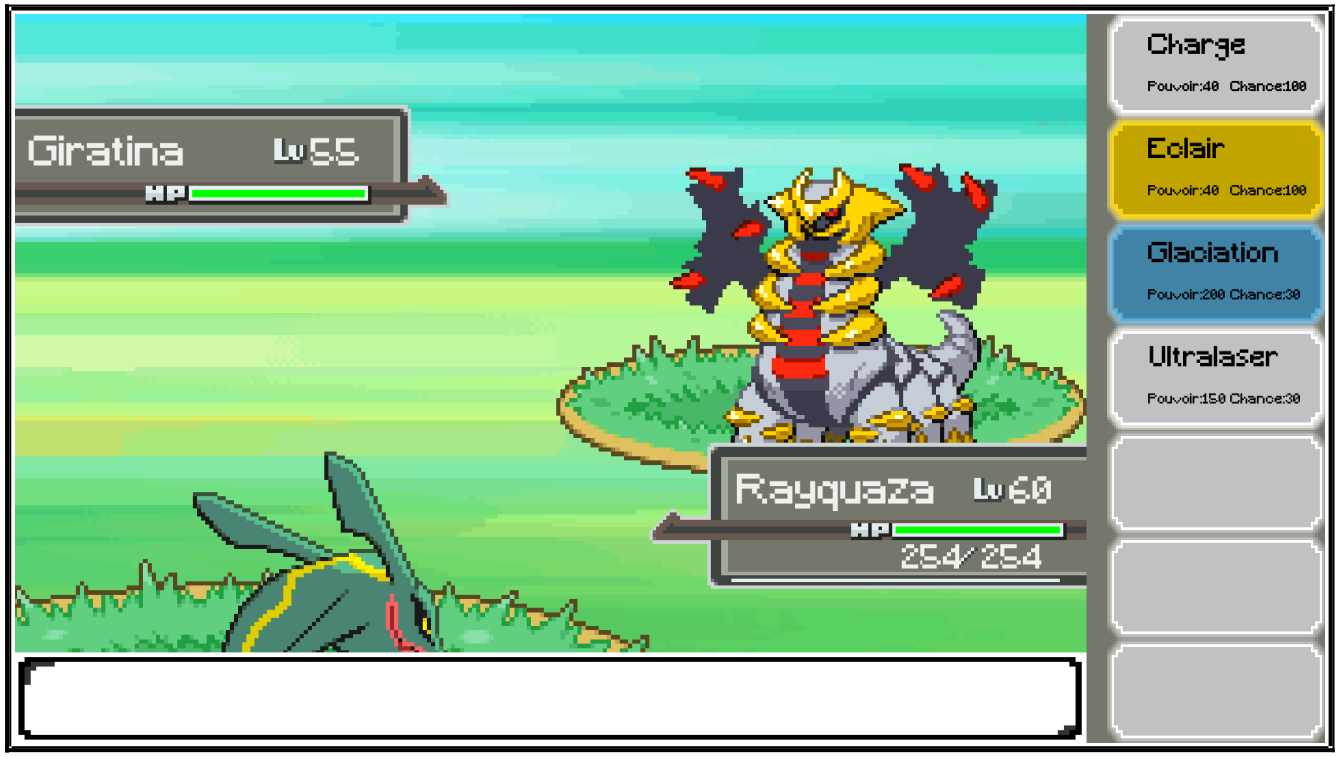
emprunter ou bien servir à décorer. Rien de mieux que quelques images pour illustrer :

Zone d'apparition

Milieu de la carte (elle n'est pas très grande)

Mais il y a également des combats que nous avons implémentés :





On a donc ici lancé un combat avec le personnage de droite

) Pourquoi avoir choisis ISN ?

Clara : J'ai choisi l'option ISN car j'ai toujours été attirée par le numérique et les programmes informatiques. J'ai voulu découvrir comment fonctionne la programmation.

Alan : C'est un domaine qui me passionne (l'informatique en général). Je trouve que l'on peut faire tellement de choses créatives et j'avais donc envie de pouvoir maîtriser d'autres langages notamment le Python.

) Quel est le but du jeu ?

Au départ, le but du jeu était très difficile à mettre en œuvre. C'est vraiment une sensation magnifique de se dire qu'au départ il n'y avait rien, et que maintenant nous en sommes là.

Ce projet a débuté non pas au début de notre 3ème trimestre de terminale mais au début de l'année de terminale. A une certaine période de l'année on avait quelques doutes quant au temps et à la difficulté que cela prendrait de faire un jeu Pokémon fonctionnel. Pendant cette même période, nous en étions à peu près à avoir inséré la fonctionnalité permettant d'afficher les images à l'écran à une position bien précise (les personnages n'existaient pas encore).

Or maintenant, nous avons un but à notre jeu ! C'est de remporter des combats pokémon ! Néanmoins le but premier de

tout Pokémons est de tous les capturer, ce qui n'est malheureusement pas encore possible dans notre jeu du fait du temps imparti pour sa création. Le but de ce pokémon sera donc juste de remporter le combat !

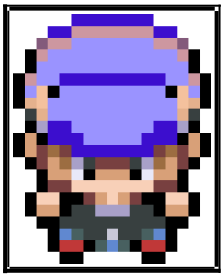
2 - La présentation:

) Un Sacrifice :

Les lignes de codes du jeu sont très longues et nécessitent parfois plusieurs minutes de réflexion incluant des fois des mathématiques (par exemple pour afficher les images). Il nous fallait donc raccourcir tout cela afin de faire une présentation à la fois du jeu mais également du programme. Il nous fallait donc faire un sacrifice et nous avons donc choisi de conserver uniquement le combat. Mais c'est une bonne chose puisque le combat est l'une des fonctionnalités indispensable de Pokémon ! En tout cas selon nous.

Nous allons donc aborder tout ce que nous avons ajouté au jeu qui permet de rendre les combats jouables et amusants !

) Personnages:



Chaque personnage (contrôlé par l'IA ou le joueur principal) est caractérisé par :

- Ses images (lorsqu'il se déplace vers la droite, gauche, haut, bas ou même lorsque qu'il reste fixe mais également son image en large, c'est à dire en style plus réaliste)
- Ses Pokémons (un joueur peut avoir maximum 6 pokémons sur lui)
- Son aptitude à pouvoir rester toujours centré sur une image (le fait qu'un joueur soit sur une image facilite tout ce qui est censé être plus complexe comme par exemple les collisions, il suffit de vérifier si l'image sur laquelle on souhaite se déplacer est libre ou pas).

*pour l'Intelligence Artificielle (IA)

- Leur capacité à pouvoir regarder uniquement dans les sens définis (c'est à dire par exemple regarder uniquement à droite et à gauche mais pas en haut et en bas).
- Leur capacité à pouvoir regarder en face d'eux pour voir dès qu'un joueur entre dans son champ de vision(ce qui permet dans les jeux officiels de lancer un combat. Dans notre jeu, cette option est possible en appuyant sur « A »).

) Pokémons :

Qui dit jeu Pokémon dit forcément qu'il y a des pokémons à l'intérieur. Nous avons donc sélectionné tous les pokémons du pokédex national que l'on peut facilement retrouver sur internet. Ainsi il y a en tout 493 pokémons que nous pouvons choisir (à partir du code uniquement).

Un pokémon est caractérisé par :

- Son image de face et de derrière.
- Son niveau (en anglais « Level » ou « Lv » pour faire court).
- Ses points de vie (en anglais « Health Points » ou encore « HP »).
- Son attaque (en anglais « Attack » ou « Atk »).
- Ses attaques (caractérisé par un type que nous verrons ultérieurement).
- Sa défense (en anglais « Defense » ou « Def »).

* Toutes les caractéristiques sont incluses dans les jeux officiels de Pokémon, mais nous ne les avons pas forcément insérées dans notre jeu (par exemple la défense n'est pas prise en compte).

* Les traductions en anglais sont essentielles pour comprendre à quoi servent les variables étant donné que la plupart d'entre elles sont écrites en anglais (avec leur diminutif parfois).

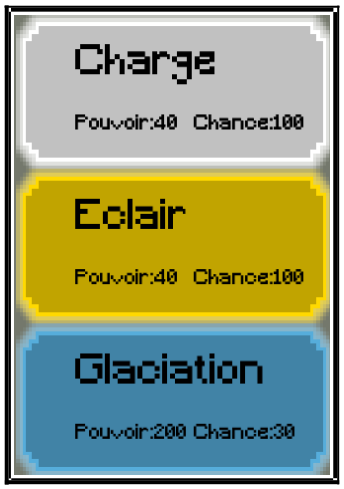
) Les Types:

Les types peuvent être appliqués à la fois aux pokémons mais également aux attaques. Il y a en tout 18 types différents mais nous n'en avons traité que quelques-uns (en vert) :

- Acier
- Eau
- Feu
- Normal
- Psy
- Spectre
- Combat
- Electrik
- Glace
- Plante
- Roche
- Ténèbres
- Dragon
- Fée
- Insecte
- Poison
- Sol
- Vol

Seul les types des attaques et les types des pokémons peuvent s'affecter entre eux (une attaque par rapport à un pokémon). Par exemple, un pokemon de type plante recevra des dégats plus importants s'il reçoit une attaque de type feu. Ceci est envisagé dans une version future. D'ailleurs, nous avons déjà anticipé les couleurs des panneaux d'attaques (car chaque type a sa propre couleur).

) Les Attaques :



Chaque pokémon du jeu possède au moins une attaque. Nous avons réussi à attribué des attaques aux pokémons mais arbitrairement pour le moment (c'est à dire qu'un pokémon de type roche peut avoir une attaque de type eau, à laquelle il est sensible…). Il existe une multitude d'attaques dont certaines même spécifiques à un pokémon, c'est pour cela que nous n'allons pas les lister.

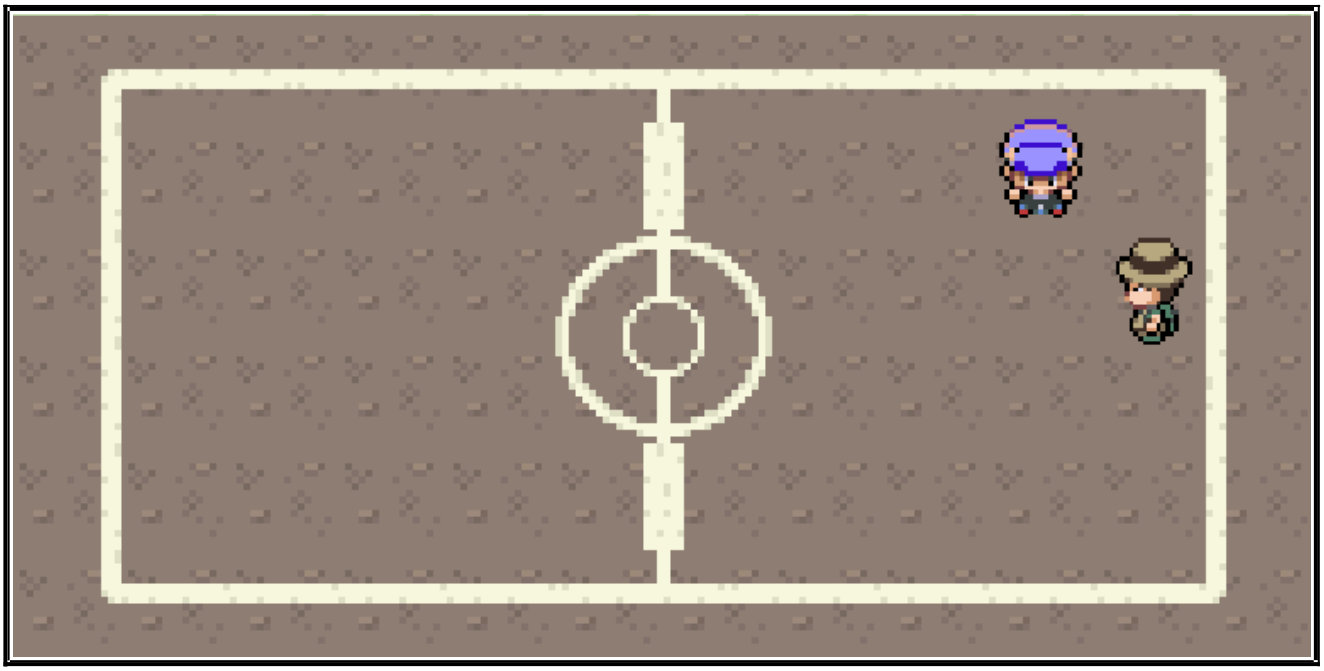
Un attaque est caractérisé par :

- Son nom
- Son type (comme vu précédemment)
- Son pouvoir (en anglais « power », c'est à dire la puissance de l'attaque)
- Sa chance de réussite (entre 0 et 100)

3 - Dans le vif du sujet:

) Lancement d'un combat :

Pour lancer un combat nous avons décidé d'attribuer une touche précise du clavier (la lettre « a ») car le code pour faire interagir le personnage avant un combat est très long et n'était donc pas envisageables pour ce « petit » projet.



Il y aura ensuite une transition rapide due à cette portion du programme.

```python
    @staticmethod
    def Transition():
        if Combat.offset[0] < width + Combat.transitionSpeed:
            pg.draw.rect(screen, black, (Combat.pos[0][0],Combat.pos[0][1],Combat.offset[0],height/4))
            pg.draw.rect(screen, black, (Combat.pos[3][0]-Combat.offset[3],Combat.pos[3][1],Combat.offset[3],height/4))
            Combat.offset[0] += Combat.transitionSpeed
            Combat.offset[3] += Combat.transitionSpeed
        elif Combat.offset[1] < width + Combat.transitionSpeed:
            pg.draw.rect(screen, black, (Combat.pos[1][0]-Combat.offset[1],Combat.pos[1][1],Combat.offset[1],height/4))
            pg.draw.rect(screen, black, (Combat.pos[2][0],Combat.pos[2][1],Combat.pos[2][0]+Combat.offset[2],height/4))
            Combat.offset[1] += Combat.transitionSpeed
            Combat.offset[2] += Combat.transitionSpeed
        elif Combat.cbg != None:
            screen.blit(Combat.cbg, (0,0))
            Combat.cbg = None
            Combat.transition = False

        pg.display.update()
        clock.tick(fpsLimit)
```

*Code Couleur :

- représente un fichier (ou module)
- représente une classe

  représente une variable
- les outils de python (if, elif, else…)
- les numéros de ligne

Cette fonction (après avoir appuyé sur « a ») va s'exécuter jusqu'à ce que la transition finisse. Ceci est dû à une boucle « while » placé à un autre endroit.
En incrémentant en fonction du temps les variables dans la liste « Combat.offset » on ajoute des valeurs au rectangle qui sera créé pour faire la transition.

) Dans un combat :

Une fois dans le combat, les attaques de chaque pokemon sont affichées à droite. Pour simplifier, tous les pokémons (même celui de l'adversaire) ont les mêmes attaques (à une exception près :)). Les points de vies de votre pokémon sont affichés à droite et ceux de l'adversaire à gauche (en haut). Il est possible de sélectionner une attaque en faisant un clic droit de la souris. Une fois appuyé, cette portion de code s'exécutera :

```python
        for event in pg.event.get():
            if event.type == pg.MOUSEBUTTONDOWN:
                for i in range(len(Combat.activePlayerPokemon.attacks)):
                    if Combat.activePlayerPokemon.attacks[i].attackButton.MouseInButton(mousePos):
                        randomNum = random.randrange(0,len(Combat.activeTrainerPokemon.attacks))

                        if Combat.activePlayerPokemon.speed > Combat.activeTrainerPokemon.speed:
                            Combat.activePlayerPokemon.AttackPokemon(i, Combat.activeTrainerPokemon)
                            Combat.trainerPokemonBar.Update()
                            Combat.activeTrainerPokemon.AttackPokemon(randomNum, Combat.activePlayerPokemon)
                        elif Combat.activePlayerPokemon.speed < Combat.activeTrainerPokemon.speed:
                            Combat.activeTrainerPokemon.AttackPokemon(randomNum, Combat.activePlayerPokemon)
                            Combat.playerPokemonBar.Update()
                            Combat.activePlayerPokemon.AttackPokemon(i, Combat.activeTrainerPokemon)
                        else:
                            r = random.randrange(0,2)
                            if r == 1:
                                Combat.activePlayerPokemon.AttackPokemon(i, Combat.activeTrainerPokemon)
                                Combat.trainerPokemonBar.Update()
                                Combat.activeTrainerPokemon.AttackPokemon(randomNum, Combat.activePlayerPokemon)
                            else:
                                Combat.activeTrainerPokemon.AttackPokemon(randomNum, Combat.activePlayerPokemon)
                                Combat.playerPokemonBar.Update()
                                Combat.activePlayerPokemon.AttackPokemon(i, Combat.activeTrainerPokemon)
```

On récupère tout d'abord l'information du clic en récupérant les évènements pygame avec une boucle « for ». On compare ensuite l'évènement pour voir si on a appuyé sur le clic gauche. Puis on cherche ensuite (ligne 1003-4) à savoir si on a cliqué sur un bouton dédié à l'attaque. On regarde donc pour chaque bouton voir si la position de la souris est à l'intérieur ou pas.

Ainsi la fonction : « Combat.activePlayerPokemon.attacks[i].attackButton.MouseInButton(mousePos) » retourne une valeur « True » si la position de la souris est dans le cadre du bouton, ou alors inversement faux si elle ne l'est pas. Voici le code exécuté dans la fonction :

```
36      def MouseInButton(self, mousePos):
37          if self.pos[0]<mousePos[0] and mousePos[0]<self.endPos[0]:
38              if self.pos[1]<mousePos[1] and mousePos[1]<self.endPos[1]:
39                  return True
40          return False
```

« mousePos » correspond dans les deux cas à la position de la souris (une liste avec une valeur pour x et une autre pour y). Mais il y a également la variable « self » dont nous n'avons pas parlé.

) Les classes:

Les classes dans python nous ont permis de réaliser environ 90 % de ce qu'on voit à l'écran. Sans elles, nous aurions dû écrire plus de 100 000 lignes de codes sûrement pour en plus ne pas pouvoir en arriver là où nous en sommes maintenant. Mais que sont les classes ?

Les classes sont issues de python. On peut définir plusieurs variables appartenant à une classe et uniquement à celle-ci. Par exemple, on sait qu'un pokémon doit avoir des attaques. Pour cela on ajoute une classe qui s'appelle « Pokemon » puis on y ajoute la variable « attacks ».

Le « self » correspond à l'objet lui-même (c'est à dire pour cibler l'instance en particulier). Voici l'exemple avec la classe « Pokemon » :

```
749 class Pokemon():
750     def __init__(self, index = 1, level = 1, name = None, type = None,type2 = None, img1=None,img2=None,hp = 10, atk = 2, defe = 2, speed = 2):
751         self.name = name
752         self.tile_index = index
753         self.type = type
754         self.type2 = None
755         self.owner = None
756         if type2 != "":
757             self.type2 = type2
758         self.frontImage = img1
759         self.backImage = img2
760         self.currentImage = None
761         if level == 0:
762             level = 1
763         self.level = level
764         self.health = hp
765         self.maxHealth = hp
766         self.attack = atk
767         self.defense = defe
768         self.speed = speed
769         self.xp = 0
770         self.max_xp = math.log(level) * 1000 + 50
771         self.ko = False
772         self.SetLevel(level)
773         self.evolution = None
774         self.attacks = []
```

Avec ces classes, on peut créer ensuite des instances qui hériteront des variables de la classe (variable qui sera propre à l'instance).Le fait de mettre ces variables dans la fonction « __init__ » permet d'attribuer des valeurs aux variables dès qu'une instance est créée.

Ainsi, chaque pokémon créé héritera de toutes les variables définies selon ce que l'on souhaite. On repère également que « self.attacks » est une liste vide au départ (on y ajoute des valeurs autre part plus tard). La fonction « init » lors de son exécution (c'est à dire dès qu'une instance est créée), donne immédiatement toute seule la variables self en paramètre, ainsi nous n'avons pas à la rentrer en appelant la fonction. Voici quelques explications :

- Créer une instance de la classe « Pokemon »
  - monPokemon = Pokemon(1, 100, « Pikachu », « Electrik »…)
- Utiliser les variables propres à l'instance
  - monPokemon.attacks = [« éclair », « coup de pied »…]
- Utiliser des fonctions
  - monPokemon.AttackPokemon(« Salamèche », « coup de pied »)
    (attaquer un pokémon avec une attaque)

Cela ne reflète pas vraiment ce qui se passe dans le programme mais on a réduit des étapes pour plus de simplicité.

De ce fait nous avons plein de classes et plein d'instances. Voici toute les classes les plus importantes (celles que qui font parties de la présentation du jeu) :

```
749   class Pokemon():
1065  class TextDisplay():
866   class Attack():
188   class PokemonBar():
  8   class Button():
879   class Combat():
```

× Il y a également les classes « Character » et « PlayerMecanics » qui correspondent aux joueurs. En effet, chague joueur possède des pokémons et donc pour accéder aux pokémons du joueur, il faut passer par la variable contenant le joueur.

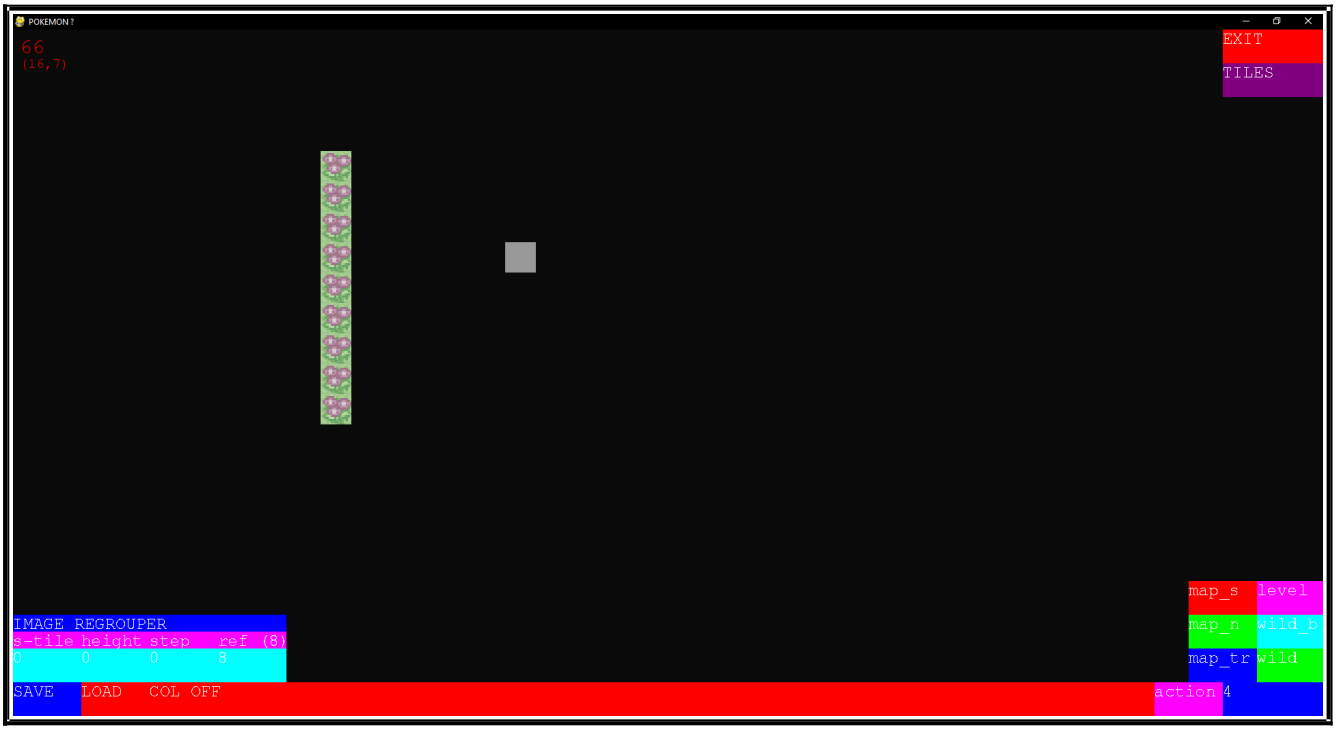
4 - Bonus :

Voici d'autres éléments du jeu que nous ne présenterons pas.

) Les cartes et le créateur de cartes :

Bien sûr nous n'allions pas rentrer toutes les valeurs des cartes à la main et nous avons donc créé un générateur de carte qui permet de placer des images à certains endroits puis d'enregistrer les images. On peut donc placer des images mais également des joueurs avec des/un pokémon(s) (possédant un niveau spécifique).

Voici à quoi le générateur de carte ressemble :



Il y a bien sûr beaucoup de choses à dire dessus mais comme le projet n'est pas centré sur ce programme nous allons donc passer très rapidement sur cette partie.

Il y a donc un espace de création et en insérant un numéro dans la case en bas à droite correspondant au numéro des images présentes dans « PokemonVBac\Assets\Graphics\Environment\Tiles » (puis appuyer sur entrer) on effectue un clic gauche de la souris pour placer les images. Il y a plusieurs autres fonctionnalités (comme le placement rapide, le placement de motifs ou le placement de joueurs). Pour cela il y a un fichier HTML (un peu vieux) qui reprend quasiment toutes ces fonctionnalités. Si tout n'y est pas il suffit de regarder un peu le programme :).

) Insertion d'éléments du jeu officiel dans notre jeu :

Il y a pleins d'images et de choses à prendre en compte dans le jeu comme par exemple les noms des pokémons, leur type, mais également toutes les images (joueurs, pokémon, carte…).

Il a donc fallu créer des petits programmes pour insérer tout à la bonne position. Par exemple pour tout ce qu'il y a dans ces dossiers (ou fichiers), les informations ont été récupérées sur « www.pokepedia.fr » en sélectionnant toute la rangée de pokémons, avec leur nom français, anglais, chinois, le(s) type(s), leur image puis en utilisant les programmes dans « PokemonVBac\Assets\Scripts\FileProcesses » afin de traiter les informations qui avaient été mises dans un fichier texte. Ceci a permis de récupérer uniquement le nom et type des pokémons) ensuite insérés par des programmes et non manuellement :

- « PokemonVBac\Assets\Graphics\Environment\Tiles » (images présentent sur les cartes)
- « PokemonVBac\Assets\Graphics\Pokemons »(graphismes des pokémons)
- « PokemonVBac\Assets\Graphics\LivingBeing\Characters » (images des personnages)
- Dans « PokemonVBac\PokemonVBac.py » ligne 212 (noms des pokémons)
- Dans « PokemonVBac\PokemonVBac.py » ligne 215 (types des pokémons)

5 - Le fichier principal (programme):

```
import pygame as pg
pg.init()

# La boucle du jeu (afficher les images de la carte)
def GameLoop():
    while not crashed:
        if menu != None:
            screen.blit(otherImages[1], (0,0))
            screen.blit(otherImages[0], (0,0))
            menu.Update()
            HandleEvents()
        else:
            screen.fill(lightBlack)
            cam.Render(deltaTime, True)
            HandleEvents()

        dt.datetime.now()
        f = clock.get_fps()
        if f != math.inf:
            f = int(f)
        label = pg.font.SysFont("monospace", 30).render(str(f), 1, (255,0,0))
        screen.blit(label, (10,10))
        pg.display.update()
        SetDeltaTime(clock.tick_busy_loop(fpsLimit))
        clock.tick(fpsLimit)

def HandleEvents():
    global menu
    keys = pg.key.get_pressed()
    mousePos = pg.mouse.get_pos()
    for event in pg.event.get():
        if event.type == pg.QUIT:
            pg.quit()
```

```
            quit()

        if event.type == pg.USEREVENT:
            pg.mixer.Channel(0).play(pg.mixer.Sound(musics[0]))
            pg.mixer.Channel(0).set_volume(0.5)

        if keys[pg.K_q]:
            # test des combats pokémons
            cam.characters[1].pokemons[0].health = cam.characters[1].pokemons[0].maxHealth
            Combat.New(player, cam.characters[1])

        if keys[pg.K_t]:
            # test de l'affichage du text
            cam.characters[1].SetText("Salut ! Moi c'est "+cam.characters[1].name+" Je te défi et tu
vas voir ce que tu vas voir. J'en suis à ma 99ème victoire d'affilé donc ne t'étonne pas de perdre ! <3
Peut être que le texte n'est pas assez long je vais donc en rajouter ! Village did removed enjoyed
explain nor ham saw calling talking. Securing as informed declared or margaret. Joy horrible moreover
man feelings own shy. Request norland neither mistake for yet. Between the for morning assured country
believe. On even feet time have an no at. Relation so in confined smallest children unpacked delicate.
Why sir end believe uncivil respect. Always get adieus nature day course for common. My little garret
repair to desire he esteem. Comfort reached gay perhaps chamber his six detract besides add. Moonlight
newspaper up he it enjoyment agreeable depending. Timed voice share led his widen noisy young. On
weddings believed laughing although material do exercise of. Up attempt offered ye civilly so sitting
to. She new course get living within elinor joy. She her rapturous suffering concealed.")
            TextDisplay.NewConversation(cam.characters[1].name, cam.characters[1].beforeBattleText)

        if keys[pg.K_RETURN]:
            player.Interact()

        if menu != None:
            if event.type == pg.MOUSEBUTTONDOWN:
                mousePos = pg.mouse.get_pos()
                if menu.play.CursorIn(mousePos) and event.button == 1:
                    LaunchGame()
                    menu = None

    if menu == None and player.target == None:
        if keys[pg.K_LEFT]:
            if keys[pg.K_RIGHT]:
                if keys[pg.K_UP]:
                    if keys[pg.K_DOWN]:
                        player.SetStatic()
                    else:
                        player.SetMovement((0,1))
                else:
                    player.SetStatic()
            elif keys[pg.K_UP]:
                if keys[pg.K_DOWN]:
                    player.SetMovement((-1,0))
                else:
                    player.SetMovement((0,1))
            elif keys[pg.K_DOWN]:
                player.SetMovement((0,-1))
            else:
                player.SetMovement((-1,0))
        elif keys[pg.K_RIGHT]:
            if keys[pg.K_UP]:
                if keys[pg.K_DOWN]:
                    player.SetMovement((1,0))
                else:
                    player.SetMovement((0,1))
            elif keys[pg.K_DOWN]:
                player.SetMovement((0,-1))
            else:
                player.SetMovement((1,0))
        elif keys[pg.K_UP]:
            if keys[pg.K_DOWN]:
                player.SetStatic()
            else:
                player.SetMovement((0,1))
        elif keys[pg.K_DOWN]:
            player.SetMovement((0,-1))
        else:
            player.SetStatic()

def LaunchGame():
    global player
    moving = False
    movementOverTime = 2
    animationTime = 1
    allowMovement = True
    pg.mixer.pre_init(44100,-16,2,5000)
    pg.mixer.init()
    pg.mixer.set_num_channels(10)
    pg.mixer.Channel(0).play(pg.mixer.Sound(musics[0]))
    pg.mixer.Channel(0).set_endevent(pg.USEREVENT)
    pg.mixer.Channel(0).set_volume(1)

    # Charger la première carte
    maps[0].LoadMap(cam)

    # Joueur principal
```

```
    player = Character("Sacha")
    player.LoadPerson(cam, cam.tiles[0][35][35], tileScale, effects)
    player.LoadAnimations(0)
    player.SetInstance()
    player.LoadPokemons([pokemons[383],pokemons[1],pokemons[30]])
    player.pokemons[0].SetLevel(60)
    player.pokemons[0].SetOwner(player)
    cam.characters.append(player)

def SetDeltaTime(time):
    global deltaTime
    deltaTime = 0
    deltaTime += time

info = pg.display.Info()
size = width, height = info.current_w, info.current_h
screen = pg.display.set_mode(size, pg.FULLSCREEN)
pg.display.set_caption('POKEMON ?')

# NOW WINDOW IS LOADED, IMPORT ALL OTHER CLASSES
from Assets.Scripts.Main import *
from Assets.Scripts.Components import *
from Assets.Scripts.Classes import *
from Assets.Scripts.tiles import *
import math

# CAMERA
cam = Camera(0,0)

menu = Menu()
combat = None

# LOADING
progressBar = ProgressBar(pg, (width/2) - 150,(height/2)-30,300, 60, darkGrey, orange)
progressText = Text([(width/2) - 45,(height/2)-25], white, "0%")
loadingFiles = Text([(width/2) - 200,(height/2)+40], white, "Loading ...")
loadingFiles.ChangeSize(15)

# MUSICS
musics.append('Assets/Sounds/Musics/Song1.ogg')
musics.append('Assets/Sounds/Musics/Song2.ogg')
effects = ['Assets/Sounds/Effects/Collision/collision.ogg']

# RANDOM IMAGES
otherImages = InitializeOthers()

# SET ATTACK SPRITES
typeButtonList.append(Button("Assets/Graphics/Combat/Normal"))
typeButtonList.append(Button("Assets/Graphics/Combat/Plant"))
typeButtonList.append(Button("Assets/Graphics/Combat/Water"))
typeButtonList.append(Button("Assets/Graphics/Combat/Ice"))
typeButtonList.append(Button("Assets/Graphics/Combat/Fire"))
typeButtonList.append(Button("Assets/Graphics/Combat/Electric"))
typeButtonList.append(Button("Assets/Graphics/Combat/Rock"))

# SET COMBAT IMAGES
combatOthers.append(MainDefinitions.CreateSpriteDim("Assets/Graphics/Combat/ChangePokemon",
int(width/6),int(height/7)))
combatOthers.append(MainDefinitions.CreateSpriteDim("Assets/Graphics/Combat/Flee",
int(width/6),int(height/7)))
combatOthers.append(MainDefinitions.CreateSpriteDim("Assets/Graphics/Combat/Bag",
int(width/6),int(height/7)))

# ALL POKEMONS
names =
["Bulbizarre","Herbizarre","Florizarre","Salamèche","Reptincel","Dracaufeu","Carapuce","Carabaffe","Tor
tank","Chenipan","Chrysacier","Papilusion","Aspicot","Coconfort","Dardargnan","Roucool","Roucoups","Rou
carnage","Rattata","Rattatac","Piafabec","Rapasdepic","Abo","Arbok","Pikachu","Raichu","Sabelette","Sab
laireau","Nidoran♀","Nidorina","Nidoqueen","Nidoran♂","Nidorino","Nidoking","Mélofée","Mélodelfe","Goup
ix","Feunard","Rondoudou","Grodoudou","Nosferapti","Nosferalto","Mystherbe","Ortide","Rafflésia","Paras
","Parasect","Mimitoss","Aéromite","Taupiqueur","Triopikeur","Miaouss","Persian","Psykokwak","Akwakwak"
,"Férosinge","Colossinge","Caninos","Arcanin","Ptitard","Têtarte","Tartard","Abra","Kadabra","Alakazam"
,"Machoc","Machopeur","Mackogneur","Chétiflor","Boustiflor","Empiflor","Tentacool","Tentacruel","Racail
lou","Gravalanch","Grolem","Ponyta","Galopa","Ramoloss","Flagadoss","Magnéti","Magnéton","Canarticho","
Doduo","Dodrio","Otaria","Lamantine","Tadmorv","Grotadmorv","Kokiyas","Crustabri","Fantominus","Spectru
m","Ectoplasma","Onix","Soporifik","Hypnomade","Krabby","Krabboss","Voltorbe","Electrode","Nœunœuf","No
adkoko","Osselait","Ossatueur","Kicklee","Tygnon","Excelangue","Smogo","Smogogo","Rhinocorne","Rhinofér
os","Leveinard","Saquedeneu","Kangourex","Hypotrempe","Hypocéan","Poissirène","Poissoroy","Stari","Star
oss","M.
Mime","Insécateur","Lippoutou","Élektek","Magmar","Scarabrute","Tauros","Magicarpe","Léviator","Lokhlas
s","Métamorph","Evoli","Aquali","Voltali","Pyroli","Porygon","Amonita","Amonistar","Kabuto","Kabutops",
"Ptéra","Ronflex","Artikodin","Électhor","Sulfura","Minidraco","Draco","Dracolosse","Mewtwo","Mew","Ger
mignon","Macronium","Méganium","Héricendre","Feurisson","Typhlosion","Kaiminus","Crocrodil","Aligatueur
","Fouinette","Fouinar","Hoothoot","Noarfang","Coxy","Coxyclaque","Mimigal","Migalos","Nostenfer","Loup
io","Lanturn","Pichu","Mélo","Toudoudou","Togepi","Togetic","Natu","Xatu","Wattouat","Lainergie","Phara
mp","Joliflor","Marill","Azumarill","Simularbre","Tarpaud","Granivol","Floravol","Cotovol","Capumain","
Tournegrin","Héliatronc","Yanma","Axoloto","Maraiste","Mentali","Noctali","Cornèbre","Roigada","Feuforê
ve","Zarbi","Qulbutoké","Girafarig","Pomdepik","Foretress","Insolourdo","Scorplane","Steelix","Snubbull
","Granbull","Qwilfish","Cizayox","Caratroc","Scarhino","Farfuret","Teddiursa","Ursaring","Limagma","Vo
lcaropod","Marcacrin","Cochignon","Corayon","Rémoraid","Octillery","Cadoizo","Démanta","Airmure","Malos
se","Démolosse","Hyporoi","Phanpy","Donphan","Porygon2","Cerfrousse","Queulorior","Debugant","Kapoera",
"Lippouti","Elekid","Magby","Écrémeuh","Leuphorie","Raikou","Entei","Suicune","Embrylex","Ymphect","Tyr
```

```
anocif","Lugia","Ho-
Oh","Celebi","Arcko","Massko","Jungko","Poussifeu","Galifeu","Braségali","Gobou","Flobio","Laggron","Me
dhyèna","Grahyèna","Zigzaton","Linéon","Chenipotte","Armulys","Charmillon","Blindalys","Papinox","Nénup
iot","Lombre","Ludicolo","Grainipiot","Pifeuil","Tengalice","Nirondelle","Hélédelle","Goélise","Bekipan
","Tarsal","Kirlia","Gardevoir","Arakdo","Maskadra","Balignon","Chapignon","Parécool","Vigoroth","Monaf
lèmit","Ningale","Ninjask","Munja","Chuchmur","Ramboum","Brouhabam","Makuhita","Hariyama","Azurill","Ta
rinor","Skitty","Delcatty","Ténéfix","Mysdibule","Galekid","Galegon","Galeking","Méditikka","Charmina",
"Dynavolt","Elecsprint","Posipi","Négapi","Muciole","Lumivole","Rosélia","Gloupti","Avaltout","Carvanha
","Sharpedo","Wailmer","Wailord","Chamallot","Camérupt","Chartor","Spoink","Groret","Spinda","Kraknoix
","Vibraninf","Libégon","Cacnea","Cacturne","Tylton","Altaria","Mangriff","Séviper","Séléroc","Solaroc",
"Barloche","Barbicha","Écrapince","Colhomard","Balbuto","Kaorine","Lilia","Vacilys","Anorith","Armaldo"
,"Barpau","Milobellus","Morphéo","Kecleon","Polichombr","Branette","Skelénox","Téraclope","Tropius","Eo
ko","Absol","Okéoké","Stalgamin","Oniglali","Obalie","Phogleur","Kaimorse","Coquiperl","Serpang","Rosab
yss","Relicanth","Lovdisc","Draby","Drackhaus","Drattak","Terhal","Métang","Métalosse","Regirock","Regi
ce","Registeel","Latias","Latios","Kyogre","Groudon","Rayquaza","Jirachi","Deoxys","Tortipouss","Boskar
a","Torterra","Ouisticram","Chimpenfeu","Simiabraz","Tiplouf","Prinplouf","Pingoléon","Étourmi","Étourv
ol","Étouraptor","Keunotor","Castorno","Crikzik","Mélokrik","Lixy","Luxio","Luxray","Rozbouton","Rosera
de","Kranidos","Charkos","Dinoclier","Bastiodon","Cheniti","Cheniselle","Papilord","Apitrini","Apireine
","Pachirisu","Mustébouée","Mustéflott","Ceribou","Ceriflor","Sancoki","Tritosor","Capidextre","Baudriv
e","Grodrive","Laporeille","Lockpin","Magirêve","Corboss","Chaglam","Chaffreux","Korillon","Moufouette"
,"Moufflair","Archéomire","Archéodong","Manzaï","Mime
Jr.","Ptiravi","Pijako","Spiritomb","Griknot","Carmache","Carchacrok","Goinfrex","Riolu","Lucario","Hip
popotas","Hippodocus","Rapion","Drascore","Cradopaud","Coatox","Vortente","Écayon","Luminéon","Babimant
a","Blizzi","Blizzaroi","Dimoret","Magnézone","Coudlangue","Rhinastoc","Bouldeneu","Élekable","Maganon"
,"Togekiss","Yanmega","Phyllali","Givrali","Scorvol","Mammochon","Porygon-
Z","Gallame","Tarinorme","Noctunoir","Momartik","Motisma","Créhelf","Créfollet","Créfadet","Dialga","Pa
lkia","Heatran","Regigigas","Giratina","Cresselia","Phione","Manaphy","Darkrai","Shaymin","Arceus"]

# ALL POKEMON TYPES (ORDERED)
types = ["Plante-Poison","Plante-Poison","Plante-Poison","Feu","Feu","Feu-
Vol","Eau","Eau","Eau","Insecte","Insecte-Acier","Vol-Insecte","Poison-Insecte","Poison-
Insecte","Poison-Insecte","Vol-Normal","Vol-Normal","Vol-Normal","Normal","Normal","Vol-Normal","Vol-
Normal","Poison","Poison","Electrik","Electrik","Sol","Eau-Sol","Poison","Poison","Poison-
Sol","Poison","Poison","Poison-Sol","Fée","Fée","Feu","Feu","Normal-Fée","Normal-Fée","Poison-
Vol","Poison-Vol","Plante-Poison","Plante-Poison","Plante-Poison","Plante-Insecte","Plante-
Insecte","Poison-Insecte","Poison-
Insecte","Sol","Sol","Normal","Normal","Eau","Eau","Combat","Combat","Feu","Feu","Eau","Eau","Eau-
Combat","Psy","Psy","Psy","Combat","Combat","Combat","Plante-Poison","Plante-Poison","Plante-
Poison","Poison-Eau","Poison-Eau","Sol-Roche","Sol-Roche","Sol-Roche","Feu","Feu","Eau-Psy","Eau-
Psy","Électrik-Acier","Électrik-Acier","Vol-Normal","Vol-Normal","Vol-Normal","Eau","Eau-
Glace","Poison","Poison","Eau","Eau-Glace","Poison-Spectre","Poison-Spectre","Poison-Spectre","Sol-
Roche","Psy","Psy","Eau","Eau","Électrik","Électrik","Plante-Psy","Plante-
Psy","Sol","Sol","Combat","Combat","Normal","Poison","Poison","Sol-Roche","Sol-
Roche","Normal","Plante","Normal","Eau","Eau","Eau","Eau","Eau","Eau-Psy","Fée-Psy","Vol-
Insecte","Glace-Psy","Électrik","Feu","Insecte","Normal","Eau","Vol-Eau","Eau-Glace","Normal","Vol-
Normal","Eau","Électrik","Feu","Normal","Eau-Roche","Eau-Roche","Eau-Roche","Eau-Roche","Vol-
Roche","Normal","Vol-Glace","Vol-Électrik","Feu-Vol","Dragon","Dragon","Vol-
Dragon","Psy","Psy","Plante","Plante","Plante","Feu","Feu","Feu","Eau","Eau","Eau","Normal","Normal","V
ol-Normal","Vol-Normal","Vol-Insecte","Vol-Insecte","Poison-Insecte","Poison-Insecte","Poison-
Vol","Eau-Électrik","Eau-Électrik","Électrik","Électrik","Fée","Normal-Fée","Fée","Vol-Fée","Vol-Psy","Vol-
Psy","Électrik","Électrik","Électrik","Plante","Eau-Fée","Eau-Fée","Roche","Eau","Plante-Vol","Plante-
Vol","Plante-Vol","Normal","Plante","Plante","Vol-Insecte","Eau-Sol","Eau-Sol","Psy","Ténèbres","Vol-
Ténèbres","Eau-Psy","Spectre","Psy","Psy","Normal-Psy","Insecte","Insecte-Acier","Normal-Sol","Vol-
Sol","Sol-Acier","Fée","Fée","Poison-Eau","Insecte-Acier","Insecte-Roche","Insecte-Combat","Glace-
Ténèbres","Normal","Normal","Feu","Feu-Roche","Sol-Glace","Sol-Glace","Eau-Sol-Roche","Eau","Eau","Vol-
Glace","Vol-Eau","Vol-Acier","Feu-Ténèbres","Feu-Ténèbres","Eau-
Dragon","Sol","Sol","Normal","Normal","Normal","Combat","Combat","Glace-
Psy","Électrik","Feu","Normal","Normal","Électrik","Feu","Eau","Sol-Roche","Sol-Roche","Roche-
Ténèbres","Vol-Psy","Feu-Vol","Plante-Psy","Plante","Plante","Plante","Feu","Feu-Combat","Feu-
Combat","Eau","Eau-Sol","Eau-Sol","Ténèbres","Ténèbres","Normal","Normal","Insecte","Insecte","Vol-Eau-
Insecte","Insecte","Poison-Insecte","Plante-Eau","Plante-Eau","Plante-Eau","Plante","Plante-Feu-
Ténèbres","Plante-Ténèbres","Vol-Normal","Vol-Normal","Vol-Eau","Vol-Eau","Fée-Psy","Fée-Psy","Fée-
Psy","Eau-Insecte","Vol-Insecte","Plante","Plante-Combat","Normal","Normal","Normal","Insecte-
Sol","Vol-Insecte","Insecte-Spectre","Normal","Normal","Normal","Combat","Combat","Normal-
Fée","Roche","Normal","Normal","Spectre-Ténèbres","Fée-Acier","Acier-Roche","Acier-Roche","Acier-
Roche","Combat-Psy","Combat-Psy","Vol-Électrik","Vol-Électrik","Électrik","Électrik","Insecte","Vol-
Insecte","Plante-Poison","Poison","Poison","Eau-Ténèbres","Eau-Ténèbres","Eau","Eau","Feu-Sol","Feu-
Sol","Feu","Psy","Psy","Normal","Sol","Sol-Dragon","Sol-Dragon","Plante","Plante-Ténèbres","Vol-
Normal","Vol-Dragon","Normal","Poison","Psy-Roche","Psy-Roche","Eau-Sol","Eau-Sol","Eau","Eau-
Ténèbres","Sol-Psy","Sol-Psy","Plante-Roche","Plante-Roche","Insecte-Roche","Insecte-
Roche","Eau","Eau","Normal","Normal","Spectre","Spectre","Spectre","Spectre","Plante-Vol","Psy","Sol-
Ténèbres","Psy","Glace","Glace","Eau-Glace","Eau-Glace","Eau-Glace","Eau","Eau","Eau","Eau-
Roche","Eau","Dragon","Dragon","Vol-Dragon","Acier-Psy","Acier-Psy","Acier-
Psy","Roche","Glace","Acier","Dragon-Psy","Dragon-Psy","Eau","Sol","Vol-Dragon","Acier-
Psy","Psy","Plante","Plante","Plante-Sol","Feu","Feu-Combat","Feu-Combat","Eau","Eau","Eau-Acier","Vol-
Normal","Vol-Normal","Vol-Normal","Normal","Eau-
Normal","Insecte","Insecte","Électrik","Électrik","Électrik","Plante-Poison","Plante-
Poison","Roche","Roche","Acier-Roche","Acier-Roche","Insecte","Plante-Insecte","Vol-Insecte","Vol-
Insecte","Vol-Insecte","Électrik","Eau","Eau","Plante","Plante","Eau","Eau-Sol","Normal","Vol-
Spectre","Vol-Spectre","Normal","Normal","Spectre","Vol-Ténèbres","Normal","Normal","Psy","Poison-
Ténèbres","Poison-Ténèbres","Acier-Psy","Acier-Psy","Roche","Fée-Psy","Normal","Vol-Normal","Spectre-
Ténèbres","Sol-Dragon","Sol-Dragon","Sol-Dragon","Normal","Combat","Acier-Combat","Sol","Sol","Poison-
Insecte","Poison-Ténèbres","Poison-Combat","Poison-Combat","Plante","Eau","Eau","Vol-Eau","Plante-
Glace","Plante-Glace","Glace-Ténèbres","Électrik-Acier","Normal","Sol-Roche","Plante","Vol-
Électrik","Feu","Vol-Fée","Vol-Insecte","Plante","Glace","Vol-Sol","Sol-Glace","Normal","Combat-
Psy","Acier-Roche","Spectre","Glace-Spectre","Électrik-Spectre","Psy","Psy","Psy","Acier-Dragon","Eau-
Dragon","Feu-Acier","Normal","Dragon-Spectre","Psy","Eau","Eau","Ténèbres","Plante","Normal",]

sTime = 0
for i in range(len(types)):
    front = MainDefinitions.CreateSpriteDim('Assets/Graphics/Pokemons/'+names[i]+"/1",
pokemonScale,pokemonScale)
```

```
    front2 = MainDefinitions.CreateSpriteDim('Assets/Graphics/Pokemons/'+names[i]+"/2",
pokemonScale,pokemonScale)
    back = MainDefinitions.CreateSpriteDim('Assets/Graphics/Pokemons/'+names[i]+"/
3",pokemonScale,pokemonScale)
    back2 = MainDefinitions.CreateSpriteDim('Assets/Graphics/Pokemons/'+names[i]+"/
4",pokemonScale,pokemonScale)
    pokemonTypes = types[i].split('-')

    if len(pokemonTypes) == 2:
        pokemons.append(Pokemon(i,0,names[i],pokemonTypes[0], pokemonTypes[1],front, back))
    else:
        pokemons.append(Pokemon(i,0,names[i],pokemonTypes[0], None,front, back))

    sTime += clock.tick_busy_loop(1000)/500
    if sTime > 0.1:
        screen.fill(Components.black)
        progress=i/len(types)
        progressBar.Update(progress)
        progressText.ChangeText((str(int(progress * 100)) + "%"))
        progressText.Update()
        loadingFiles.ChangeText("Loading : Assets/Graphics/Pokemons/"+names[i]+"/1.png")
        loadingFiles.Update()
        pg.display.update()
        m.clock.tick(10000)
        sTime = 0

pokemons[0].SetEvolution(1)
pokemons[1].SetEvolution(2)

# ALL ATTACKS
attacks.append(Attack("Charge", Attack.TYPES[0], copy.copy(typeButtonList[0]), 40, 100))
attacks.append(Attack("Eclair", Attack.TYPES[5], copy.copy(typeButtonList[5]), 40, 100))
attacks.append(Attack("Glaciation", Attack.TYPES[3], copy.copy(typeButtonList[3]), 200, 30))
attacks.append(Attack("Ultralaser", Attack.TYPES[0], copy.copy(typeButtonList[0]), 150, 30))
attacks.append(Attack("Devoir Maison", Attack.TYPES[0], copy.copy(typeButtonList[0]), 70, 80))
attacks.append(Attack("Intégrale", Attack.TYPES[0], copy.copy(typeButtonList[0]), 50, 90))
attacks.append(Attack("KWYK", Attack.TYPES[0], copy.copy(typeButtonList[0]), 60, 100))
attacks.append(Attack("Proba continue", Attack.TYPES[0], copy.copy(typeButtonList[0]), 75, 80))

for i in range(len(pokemons)):
    for j in range(len(attacks)):
        if i != 486:
            if j < 4:
                pokemons[i].attacks.append(attacks[j])
        else:
            if j > 3:
                pokemons[i].attacks.append(attacks[j])

# <== MAPS ==>

Map.allTiles = Initialize([progressBar, progressText, loadingFiles])
for i in range(1,13):
    Map.transitionTiles.append(MainDefinitions.CreateSpriteDim("Assets/Graphics/Map/Transitions/
Def/"+str(i), pg.display.Info().current_w,pg.display.Info().current_h))

maps = []
maps.append(Map(tileScale, "Assets/Maps/Villages/Mtp2.map"))
maps.append(Map(tileScale, "Assets/Maps/Villages/Montpellier.map"))
Camera.maps = maps

# <== END MAPS ==>

GameLoop()
```

6 - Informatique et Science du Numérique:

Ça a été un plaisir pour nous deux de suivre les cours d'ISN et d'avoir ainsi pu en apprendre plus sur la programmation et la façon dont un projet doit être structuré. Merci beaucoup à *Mr Mischler* et *Mr Zabban* pour cette année.

Hernandez Clara et Blanchet Alan