A textual adventure game

Gaëtan Blaise-Cazalet - Head of programming section

Textual adventure?

Basics of a video game

```
window {
    load();

    loop {
        inputs();
        update();
        draw();
    }
}
```

Vertical slice

- Pressing a keyboard key will send us to an other choice

Write text on multiple lines, try 1

- On peut essayer d'écrire le texte en concaténant des chaînes de caractères :

```
# Load
    pygame.init()
    path = "c:\ArtFx\Cours\Python\python-training\01.adventure_text\code\00.start\"
    screen = pygame.display.set_mode((800, 600))
    font = pygame.font.Font(path + "arial.ttf", 24)
    text = font.render("Alors tu veux créer des jeux videos ?\n G pour devenir Game Designer\n P pour devenir Programmeur\n", False, (0, 0, 0))
    quit = False
```

- But :

Write text on multiple lines, try 2

- We will use an array. One declares and fills an array like this :

```
# Load
    ...
    text = []
    text.append('Alors du veux créer des jeux vidéo ?')
    text.append('G pour devenir Game Designer')
    text.append('A pour devenir Artiste')
    text.append('P pour devenir Programmeur')
    text.append('Z pour devenir Producteur')
```

- And one displays it like this :

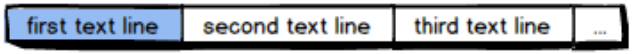
```
        # Draw
        screen.fill((0, 0, 0))

        for i in range(0, text.__len__()):
            text_surface = font.render(text[i], False, (255, 255, 255))
            screen.blit(text_surface, (10, i * 30 + 20))

        pygame.display.update()
```

What is an array?

- An array is a data structure that contains ordered elements

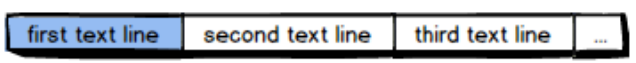


- One accesses the array's elements by calling them through their index (here called i) between brackets

```
text[i]
```

- The for loop allows i to vary between the first array index (index 0) and the last, excluded (array length)

```
for i in range(0, text.__len__()):
    text_surface = font.render(text[i], False, (255, 255, 255))
    screen.blit(text_surface, (10, i * 30 + 20))
```



Alors du veux créer des jeux vidéo ?

G pour devenir Game Designer

A pour devenir Artiste

P pour devenir Programmeur

Z pour devenir Producteur

Implement options

- Create different texts for options
- Create booleans that are flagged when a kay is pressed. Don't forget pygame uses a QWERTY keyboard:
  - A : Z
  - G : G
  - P : P
  - Z : W
- Ne pas oublier de remettre le texte à 0 avec :

```
text = []
```

Implement options - Code

```
# Load
...
text = []
text.append('Alors du veux créer des jeux vidéo ?')
text.append('G pour devenir Game Designer')
text.append('A pour devenir Artiste')
text.append('P pour devenir Programmeur')
text.append('Z pour devenir Producteur')

quit = False
key_a = False
key_g = False
key_p = False
key_z = False

while not(quit):
    # Inputs
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            quit = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                quit = True
            if event.key == pygame.K_q:
                key_a = True
            if event.key == pygame.K_g:
                key_g = True
            if event.key == pygame.K_p:
                key_p = True
            if event.key == pygame.K_w:
                key_z = True

    # Update
    if key_g:
        text = []
        text.append('Tu préfères les petits jeux Smartphones ou les Grosses productions ?')
        text.append('S pour Smartphone')
        text.append('G pour Grosses production')
    if key_p:
        text = []
        text.append('Plutôt programmeur Gameplay ou Moteur ?')
        text.append('G pour Gameplay')
        text.append('M pour Moteur')
    ...
```

Plutôt programmeur Gameplay ou Moteur ?

G pour Gameplay

M pour Moteur

Two problems

- First problem : if one presses A while we are on Programer page, we go to Artist page
- Second problem : G key cannot be used for Game designer, Gameplay programer and "Grosse production"

Two problems, one solution : state machine

- A state machine is a (logical) machine which behaves some way when it is in a certain state...
- ...and which behaves in an other way when it is in an other state.
- To implement a state machine, we create a "state" variable that holds the current state
- We will test key presses in function of the current state
- One can check a test (for instance 'start' state) with the condition :

```
if state == 'start':
    ...
```

Two problems, one solution : state machine

- main.js

```
# Update
        if state == 'start':
            if key_a:
                text = []
                text.append('Tu te spécialises en graphisme 2D ou en 3D ?')
                text.append('D pour 2D')
                text.append('T pour 3D')
                state = 'artiste'

            if key_g:
                text = []
                text.append('Tu préfères les petits jeux Smartphones ou les Grosses productions ?')
                text.append('S pour Smartphone')
                text.append('G pour Grosses production')
                state = 'gd'

            if key_p:
                text = []
                text.append('Plutôt programmeur Gameplay ou Moteur ?')
                text.append('G pour Gameplay')
                text.append('M pour Moteur')
                state = 'programmeur'

        if key_z:
            text = []
            text.append('Tu travailles en Freelance ou dans une Entreprise ?')
            text.append('F pour Freelance')
            text.append('E pour Entreprise')
            state = 'producteur'

    if state == 'gd':
        if key_g:
            text = []
            text.append('Il te faudra te spécialiser au sein d\'une équipe.')
            state = 'gd grosses productions'
        if key_s:
            text = []
            text.append('Tu vas devoir travailler seul sur plusieurs compétences.')
            state = 'gd smartphone'

    if state == 'programmeur':
        if key_g:
            text = []
            text.append('Un boulot répandu et facile à trouver.')
            state = 'programmeur gameplay'
        if key_m:
            text = []
            text.append('Un boulot assez spécifique, mais très apprécié.')
            state = 'programmeur moteur'
```

Caractère d'échappement pour l'apostrophe

Problem

- If we press G (game designer), we go directly to state Game designer grosses productions
- It is as if we would have pressed G twice
- Why this bug ?

Reset key press : design

- The problem comes from the key_g variable, which stays True, so Grosses production choice is selected immediatly
- The solution is set variables key_XXX False after each time we change current state
- Create a reset_keys() function with this purpose

Reset key press : code

- main.js

```
def main():

    # Load
    ...

    def reset_keys():
        nonlocal  key_a, key_g, key_p, key_z, key_s, key_m, key_space
        key_a = key_g = key_p = key_z = key_s = key_m = key_space = False


    ...

    # Update
        if state == 'start':
            if key_a:
                text = []
                text.append('Tu te spécialises en graphisme 2D ou en 3D ?')
                text.append('D pour 2D')
                text.append('T pour 3D')
                state = 'artiste'
                reset_keys()
            ...
```

Improve code : avoid copy & paste

- In our code, we often write:

```
state = 'gd grosses productions';
reset_keys();
```

- We could replace this with a function that would take the new state as an argument
- This function would have this signature :

```
def change_state(new_state)
```

We pass an argument
which will be used in the function

- And would be called with :

```
change_state('gd grosses production');
```

Improve code : avoid copy & paste

- main.js

```
# Load
...
def change_state(new_state):
    nonlocal state
    reset_keys()
    state = new_state


...
    # Update
    if state == 'start':
        if key_a:
            text = []
            text.append('Tu te spécialises en graphisme 2D ou en 3D ?')
            text.append('D pour 2D')
            text.append('T pour 3D')
            change_state('artiste')

        if key_g:
            text = []
            text.append('Tu préfères les petits jeux Smartphones ou les Grosses productions ?')
            text.append('S pour Smartphone')
            text.append('G pour Grosses production')
            change_state('gd')
        ...
```

nonlocal is used to modify a variable declared in the above block

Improve code: separate data and gameplay

- In current code, we mix text variable changes (data) and state changes (gameplay):

```
        # Update
        if state == 'gd':
            if key_g:
                text = []
                text.append('Il te faudra te spécialiser au sein d\'une équipe.')
                change_state('gd grosses productions')
...
```

- Adding a state or changing a state text would not be easy
- We will separate data and gameplay by creating a change_text(state) method: it will take a state as a parameter and change text

Improve code: separate data and gameplay

- main.js

```
# Load
    ...
    def change_state(new_state):
        nonlocal state
        state = new_state
        reset_keys()
        change_text(new_state)

    def change_text(state):
        nonlocal text
        if state == 'start':
            text = []
            text.append('Alors du veux créer des jeux vidéo ?')
            text.append('G pour devenir Game Designer')
            text.append('A pour devenir Artiste')
            text.append('P pour devenir Programmeur')
            text.append('Z pour devenir Producteur')


     ...

        # Update
        if state == 'start':
            if key_a:
                change_state('artiste')
            if key_g:
                change_state('gd')
            if key_p:
                change_state('programmeur')
            if key_z:
                change_state('producteur')

        if state == 'gd':
            if key_g:
                change_state('gd grosses productions')
            if key_s:
                change_state('gd smartphone')
```

Now, everything is tidy, and you are an happy developer, so...

Create your own adventure!

(And allow our player to start over with the space bar.)