

The Dino Game

Le jeu le plus délirant jamais crée

Yassine "Y4ss" Chouiti

WEB DEVELOPER

Jules "T00rtue" Holl

SOFTWARE DEVELOPER

Sommaire

- Idée du jeu
- Langage utilisé
- Différents composants
- Design
- Difficultés rencontrées

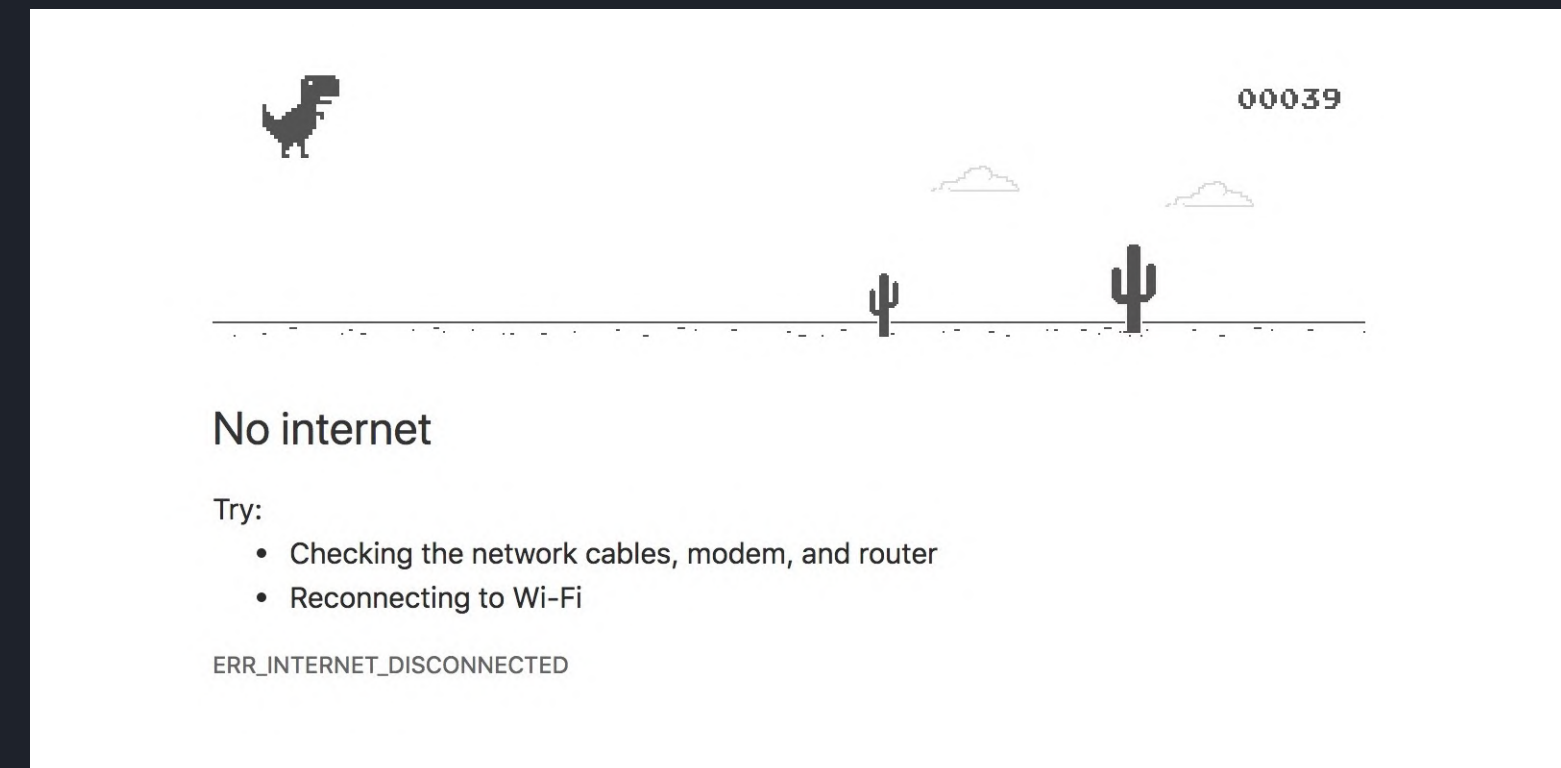
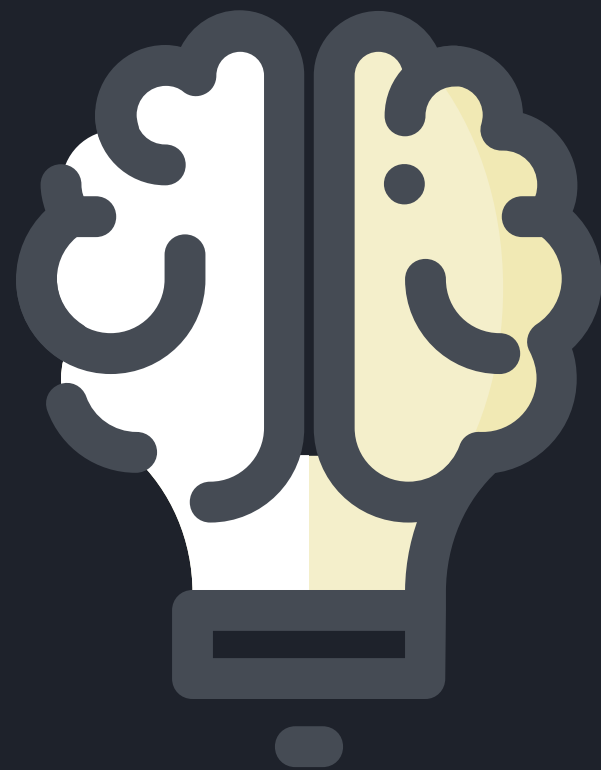


Comment nous est venue l'idée du jeu

LA SUITE VA VOUS SURPRENDRE

Situation

Idée initiale puis changement en cours de route





Langage et bibliothèques utilisés

FINI LA RIGOLADE

LANGAGE ET BIBLIOTHÈQUE



Python 3

Le langage informatique utilisé



PyGame

La principale librairie utilisée



Différents composants du jeu

RENTRONS DANS LE CODE DU SUJET

Avoir plusieurs composants

ET POURQUOI CELA EST IMPORTANT

- Meilleure lecture des fichiers
- Facilité à trouver les erreurs
- Permet le travail de groupe

```
def reset_keys(self):
    """
    This function is for reset all keys events.
    :return: none
    """
    self.UP_KEY, self.DOWN_KEY, self.LEFT_KEY, self.RIGHT_KEY = False, False, False, False

    self.ENTER_KEY, self.ESCAPE_KEY = False, False

def text(self, text, pos, font, color1, color2):
    """
    This function is to draw automatically a text on screen.
    :param text: the text of the text
    :param pos: the position (top_left) of the text
    :param font: the font of the text
    :param color1: the color of the text
    :param color2: the color of text's background
    :return: none
    """
    if font == "MENU":
        var_text = MENU_FONT.render(text, False, color1, color2)
        self.screen.blit(var_text, pos)
    elif font == "STAT":
        var_text = STAT_FONT.render(text, False, color1, color2)
        self.screen.blit(var_text, pos)
    elif font == "TITLE":
        var_text = TITLE_FONT.render(text, False, color1, color2)
        self.screen.blit(var_text, pos)
    elif font == "TEXT":
        var_text = TEXT_FONT.render(text, False, color1, color2)
        self.screen.blit(var_text, pos)

def text_fps(self, text, pos, font, color1, color2):
    """
    This function is to draw automatically a text who is actualised for each fps on screen.

    :param text: the text of the text
    :param pos: the position (top_left) of the text
    :param font: the font of the text
    :param color1: the color of the text
    :param color2: the color of text's background
    :return: none
    """
    if font == "MENU":
        var_text = MENU_FONT.render(text, False, color1, color2)
    elif font == "STAT":
        var_text = STAT_FONT.render(text, False, color1, color2)
    elif font == "TITLE":
        var_text = TITLE_FONT.render(text, False, color1, color2)
    elif font == "TEXT":
        var_text = TEXT_FONT.render(text, False, color1, color2)
    self.screen.blit(var_text, pos)
```


Un fichier final

```
from [composants].[nom_du_composant] import *
```

- Un seul fichier final qui appelle tous les composants

Mais il ne faut pas oublier que :

- Les bibliothèques doivent être compatibles
- Le fichier peut vite devenir lourd

Chaque composant du jeu remplit sa propre fonction

L'exécution

Noyau du jeu

Les événements

Collisions, retour au menu

L'affichage

Fenêtre, responsive

Le menu

New game, options...



Le choix du design

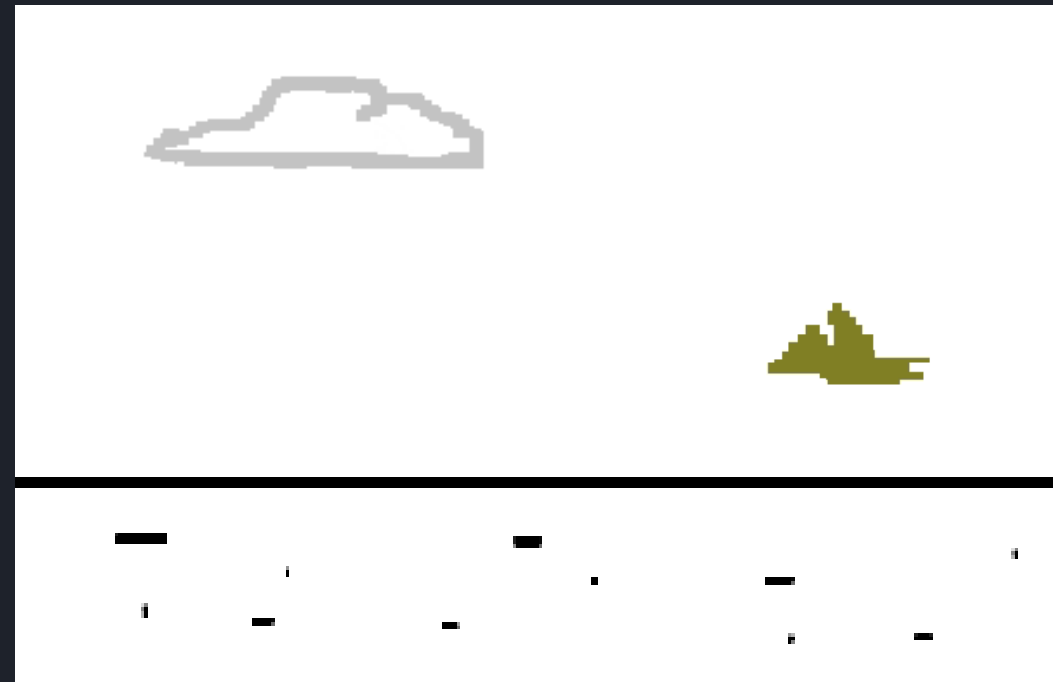
PAS DE PLAGIAT !

LE CHOIX DU DESIGN



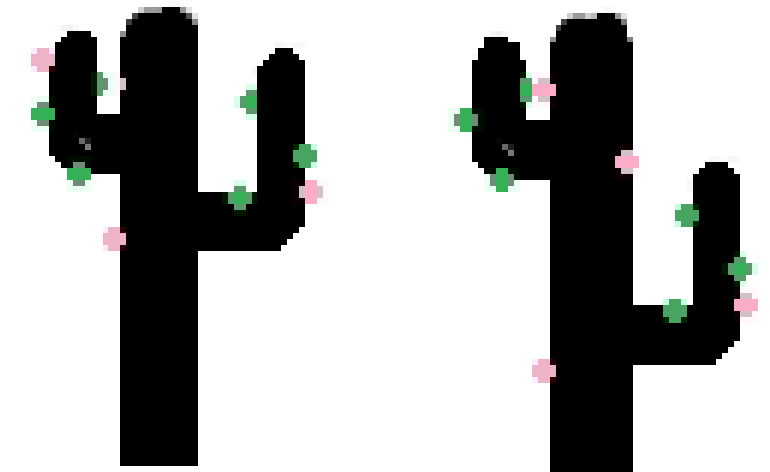
Le Dinosaur

Tel l'est sur le jeu original,
mais refait de A à Z



Une route

Très ressemblante, mais refaite



Les cactus

Refait eux aussi



Les difficultés rencontrées

ON A DU PAIN SUR LA PLANCHE

DIFFICULTÉS RECONTRÉES



Ne pas savoir par où commencer

Aussi appelé :
"Syndrome de la page blanche"

Ecran qui clignote

Problème au niveau de
l'affichage des textes

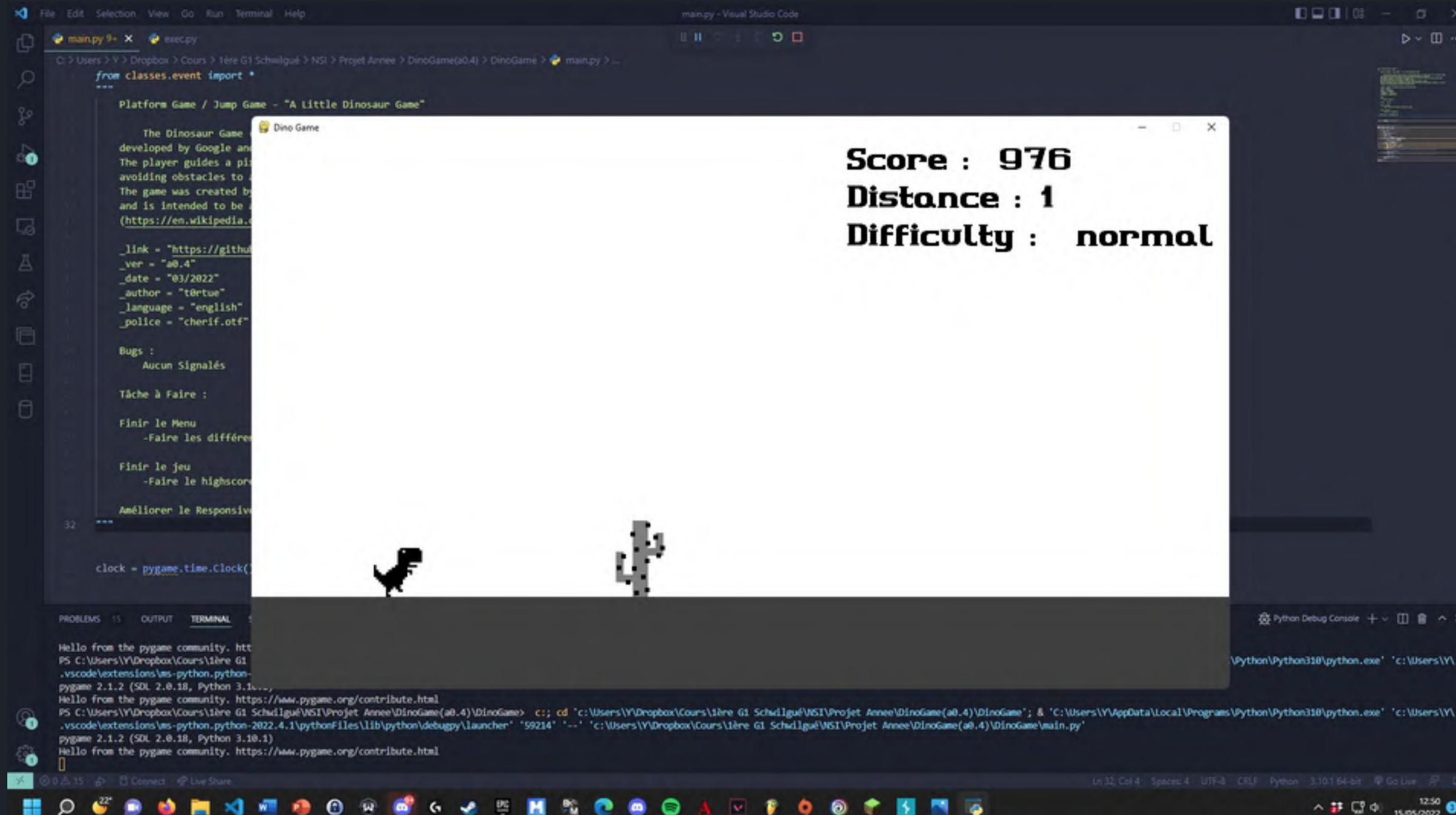
Responsive

Faire un responsive fixe et non
resizable "librement"

Vitesse du jeu

Sur les PC plus puissants que
ceux du lycée (non résolu)

THE END



Le fruit de notre travail