# 30 ans du Tamagotchi

Curiosité Numérique Digeste du 27 octobre 2026 par emoc




Quelques liens de sites présentés

**1996 ?**

- The information superhighway : reportage de 1994 (BBC) sur le début du web : https://www.bbc.com/videos/czv20818q2no
- https://56k.rip/
- https://3615co.de/

**Tamagotchi / visuel**

- https://logos.fandom.com/wiki/Tamagotchi
- Une feuille de sprites complète P1/P2 : https://www.spriters-resource.com/lcd_handhelds/tamagotchioriginalp1p2/asset/144400/
- Les magnifiques sprites en noir et blanc des premières générations : https://www.spriters-resource.com/lcd_handhelds/tamagotchiconnectionversion2/
- Bandaï : https://fr.wikipedia.org/wiki/Bandai

**Y jouer en ligne :** : https://autosam.github.io/Tamaweb/

**Quelques projets libres**

- TamaFi : https://github.com/cifertech/TamaFi (esp32, se nourrit du wifi!)
- TamaPoke : https://github.com/socquique/TamaPoke
- Termgotchi : https://github.com/pfurpass/Termgotchi
- Tama96 : https://github.com/siegerts/tama96
- ArduinoGotchi : https://github.com/anabolyc/Tamagotchi

**Utiliser Termgotchi**

```
sudo apt install python3-venv python3-pip git
git clone https://github.com/pfurpass/Termgotchi.git
cd Termgotchi
python3 -m venv .venv
source .venv/bin/activate
pip install -e .

serverpet start --name Byte --species cat --personality cheerful
serverpet watch
```

**Jeux «virtual pets» / simulation de vie**

Virtual Pet Homepage : https://www.virtualpet.com/vp/

- Little Computer People (1985) :
  - https://www.mobygames.com/game/9241/little-computer-people/
  - Sur la partie générative : https://archive.wikiwix.com/cache/?url=http%3A%2F%2Fwww.softpres.org%2F%3Fid%3Darticle%3Agame%3Alittle_computer_people

- Puppy Love (1986)
    - https://www.mobygames.com/game/84568/puppy-love/
    - https://macfan.book.mynavi.jp/article/85817/ (.jp)
    - https://www.macintoshrepository.org/4994-puppy-love : download
- Furby (1998) Tiger Electronics / Hasbro
    - https://tigerelectronics.miraheze.org/wiki/Furby
- Les Sims (2000) / Will Wright / Maxis
    - https://fr.wikipedia.org/wiki/Les_Sims_(jeu_vid%C3%A9o)
    - https://tcrf.net/Prerelease:The_Sims_(Windows) (sur The Cutting Room Floor)
    - Corpus Simsi (Chloé Delaume) : https://chloedelaume.net/?page_id=294
- Nintendogs (2005)
    - https://nintendogs.fandom.com/wiki/Main_Page
- Tomodachi Life (2013, 2026)
    - https://www.mobygames.com/game/84860/tomodachi-life/
    - https://www.mobygames.com/game/259393/tomodachi-life-living-the-dream/

# Diaporama Karaoke

Un diaporama qui change de slides quand on fait assez de bruits réalisé avec Processing...

En zip avec les images :

diaporama_karaoke.zip

### diaporama_karaoke.pde (cliquer pour afficher le code)

diaporama_karaoke.pde

```
/*  Diaporama - Karaoke
    Récupérer le son du micro pour faire défiler un diaporama
      chaque slide est associé à un court texte
    Utilisé pour la CND «30 ans du Tamagotchi» / 27 octobre 2026
    processing 4.0b2 @ Tenko / pierre@lesporteslogiques.net

    2 paramètres :
      - durée minimum (ms) : une slide reste affichée au moins ce temps là
      - seuil : seuil sonore de déclenchement (0 à 1)
    Modifiables au clavier
      - haut/bas : augmenter/réduire la durée (tranches de 0.1 entre 0 et 1)
      - gauche/droite : baisser/augmenter le seuil (tranches de 500ms, min 0, max 10000ms)
      - p : affichage ou pas des valeurs de paramètres et du niveau sonore capté
      - t : affichage ou pas du texte de légende
    Un clic souris fait aussi passer à la slide suivante

    Notes : le réaffichage de l'image ne se fait que si nécessaire, cad :
      - nouvelle image
      - affichage des paramètres affichés
 */


import processing.sound.*;          // Importer la bibliothèque dédiée au son

boolean DISPLAY_PARAMS  = true;     // affichage des paramètres
boolean REFRESH_DISPLAY = false;    // faut-il rafraîchir l'affichage ?
boolean DISPLAY_TEXT    = true;     // faut-il afficher le texte de légende ?

AudioIn micro;                      // Créer un objet nommé micro de type AudioIn
Amplitude volume;                   // Créer un objet nommé volume de type Amplitude

float niveau_sonore = 0;            // niveau sonore lissé
float seuil = 0.1;                  // seuil sonore à atteindre
long changed_last = 0;              // instant du dernier changement de slide
long duree_mini = 3000;             // durée minimale d'affichage d'une slide

PImage[] images;                    // Tableau d'images
int slideCourante = 0;              // index de l'image en cours du tableau
boolean suivant = false;            // état utilisé pour le changement d'image


// ***************** Images et légende *********

// Tableau de données : { nom de fichier, texte à afficher }
String[][] slides = {
  { "debatchi.png", "debatchi" },
  { "dorotchi.png", "dorotchi" },
  { "hanatchi.png", "hanatchi" },
```

```
  { "kabutchi.png", "kabutchi" },
  { "kiwitchi.png", "kiwitchi" },
  { "kusatchi.png", "kusatchi" },
  { "mimitchi.png", "mimitchi" },
  { "oniontchi.png", "oniontchi" },
  { "propellertchi.png", "propellertchi" },
  { "takotchi.png", "takotchi" },
  { "tarakotchi.png", "tarakotchi" },
  { "marumimitchi.png", "marumimitchi" }
};


void setup() {
  fullScreen(); //
  //size(1200, 675);
  background(255);
  frameRate(20);
  noSmooth();                   // Pas d'interpolation pour agrandir les images
  Sound.list();                 // Debug : liste des devices sonores

  micro = new AudioIn(this, 0); // Créer une entrée micro
  micro.start();                // Démarrer l'écoute du micro
  volume = new Amplitude(this); // Démarrer l'analyseur de volume
  volume.input(micro);          // Brancher l'entrée micro sur l'analyseur de volume

  // Chargement de toutes les images une seule fois
  images = new PImage[slides.length];
  for (int i = 0; i < slides.length; i++) {
    images[i] = loadImage(slides[i][0]);
  }
}


void draw() {

  // Mesurer le volume sonore *************************
  float niveau_actuel = volume.analyze(); // renvoie une valeur entre 0 et 1
  // lissage des valeurs pour «gommer» les aberrations
  niveau_sonore = (niveau_sonore * 0.95) + (niveau_actuel * 0.05);


  // Tester si on change de slide *********************
  if (niveau_sonore > seuil && (millis() - changed_last > duree_mini)) {
    suivant = true;
    niveau_sonore = 0;
  }
  if (suivant) {
    changed_last = millis();
    slideCourante = (slideCourante + 1) % slides.length;
    REFRESH_DISPLAY = true;
    suivant = false;
  }

  // Faut il rafraîchir l'écran ? ***********************
  if (DISPLAY_PARAMS) {
    REFRESH_DISPLAY = true;
  }
  if (REFRESH_DISPLAY) {
    afficherSlide(slideCourante, DISPLAY_TEXT);
    REFRESH_DISPLAY = false;
  }

  // Indications des valeurs de paramètres ***************
  if (DISPLAY_PARAMS) {
    float diametre = map(niveau_sonore, 0, 1, 1, 500);
    noStroke();
    fill(255, 0, 150);
    ellipse(55, 100, diametre, diametre);
    textSize(14);
    textAlign(LEFT, CENTER);
    text(niveau_sonore, 20, 20);
    text(seuil, 60, 20);
    text(duree_mini, 20, 40);
    text(millis() - changed_last, 20, 60);
  }

}

void afficherSlide(int n, boolean display_text ) {
  background(255);

  // Image centrée, redimensionnée pour tenir dans la fenêtre
  PImage img = images[n];
  if (img != null && suivant == false) {
    imageMode(CENTER);
    float ratio = min((float) width / img.width, (float) height / img.height);
    image(img, width/2, height/2, img.width * ratio, img.height * ratio);
  }

  // Afficher le texte
  if (display_text) {

    /* // Bandeau semi-transparent pour la lisibilité du texte
    noStroke();
```

```
    fill(0, 150);
    rectMode(CORNER);
    rect(0, height - 80, width, 80);
    */

    fill(255, 0, 150);
    textSize(64);
    textAlign(CENTER, CENTER);
    text(slides[n][1], width/2, height - 100);
  }
}

void keyPressed() {
  if (key == 'p') {
    DISPLAY_PARAMS = !DISPLAY_PARAMS;
  }
  if (key == 't') {
    DISPLAY_TEXT = !DISPLAY_TEXT;
  }
  if (key == CODED) {
    if (keyCode == RIGHT) {
      seuil += 0.1;
      if (seuil > 0.9) seuil = 0.9;
    } else if (keyCode == LEFT) {
      seuil -= 0.1;
      if (seuil < 0.1) seuil = 0.1;
    } else if (keyCode == UP) {
      duree_mini += 500;
      if (duree_mini < 0) duree_mini = 0;
    } else if (keyCode == DOWN) {
      duree_mini -= 500;
      if (duree_mini > 10000) duree_mini = 10000;
    }
  }
}

void mousePressed() {
  suivant = true;
}
```

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/cnd/30_ans_du_tamagotchi?rev=1793614162**
Article mis à jour: **2026/11/02 11:09**