

Web Development's Final Project

Author	F. Javier Gimeno Ortiz
Professor	Alicia Ansurias
Course	Web Development
Year	Fresh Year 22-23
Career	VPG (VideoGame Programming)
School	ESAT (Escuela Superior de Arte y Tecnología)



Hotline Miami Web Page Documentation

Table of Contents:

1. About...

2. Web Walkthrough

- Index Page
- Characters Page
- Contact Page
- Developers Page
- Misc. Page

3. Web Overview

- Browser Requirements

4. Navigation

- Main Menu and Submenus
- Exploring Content Categories

5. Bibliography

- Images

About Hotline Miami Wiki

This wiki aims to provide valuable information about the video games "Hotline Miami" and "Hotline Miami 2".

While it may not have the polished appearance of more professional websites, creating this wiki has been a great opportunity to learn more about CSS, HTML, and JavaScript.

But why did I choose to create a wiki?

I personally find wikis incredibly useful, especially when I encounter challenges or get stuck while playing a video game. That's one of the reasons why I embarked on the journey to recreate one myself.

To explore further details and access the webpage, please follow this link: gimenoor.github.io

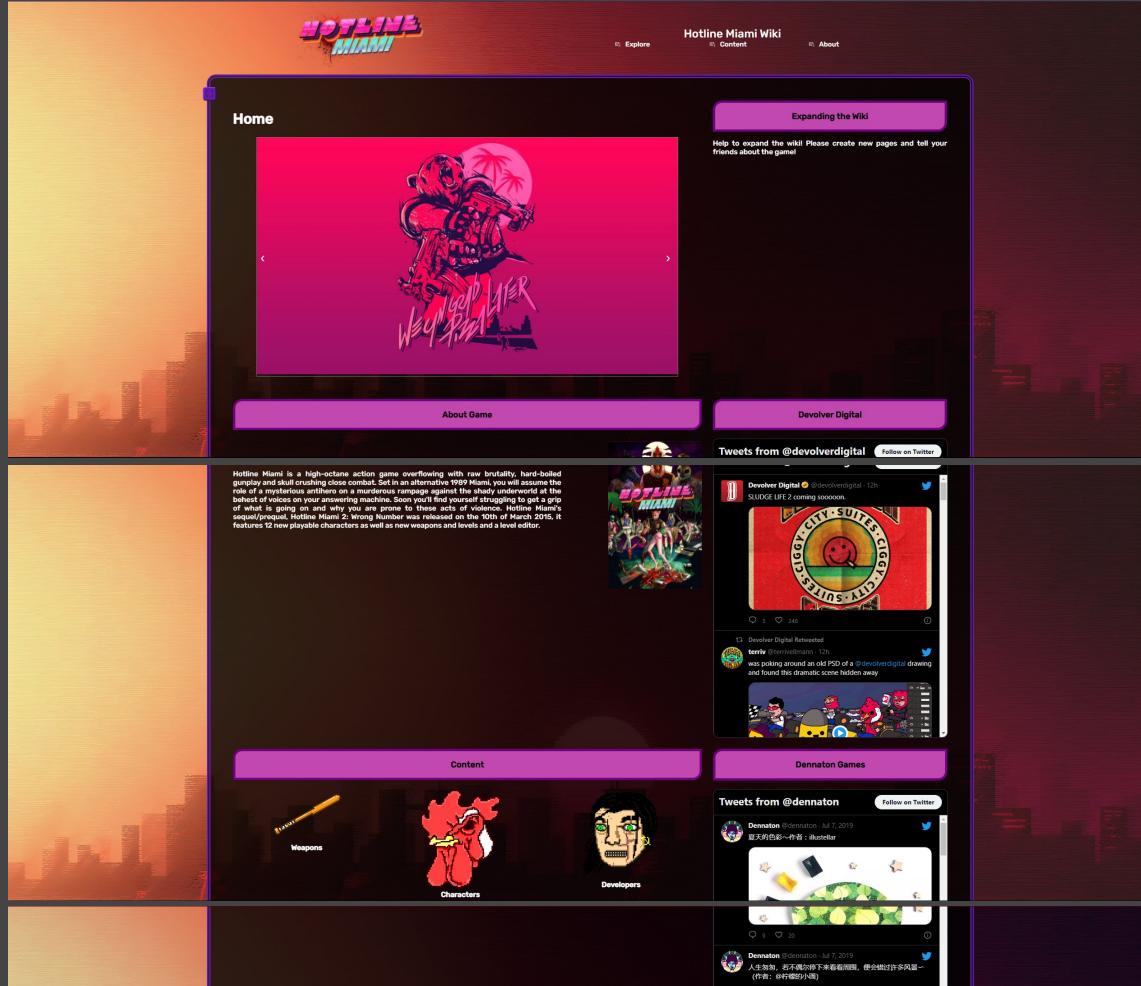
Web Walkthrough

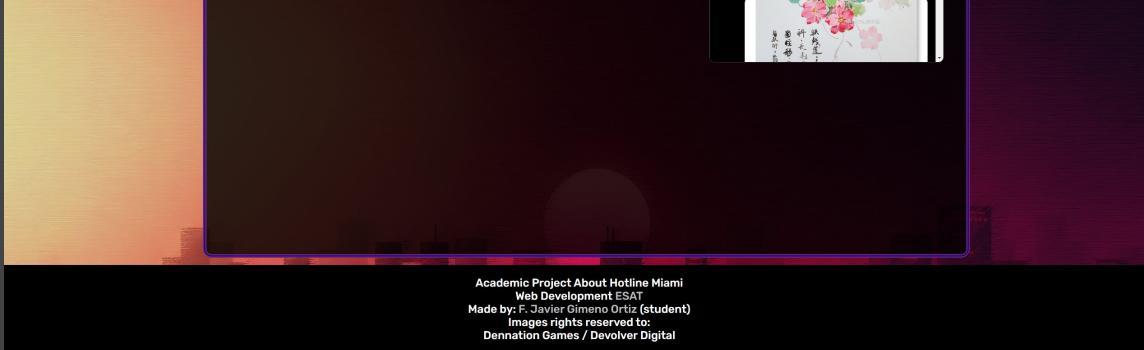
• Index Page

- Header that only shows when scrolling down the page:



- Content:





• Characters Page

Trending Pages

The Girlfriend

Richard

The Son

Tony

Biker

Beard

Corey

Manny Pardo

Devolver Digital

[Follow on Twitter](#)

Devolver Digital (@devolverdigital) · 12h
SLUDGE LIFE 2 coming soon...

Devolver Digital (@devolverdigital) · 12h
terry (@terrythebrain) · 12h
was poking around an old PSD of a @devolverdigital drawing and found this dramatic scene hidden away

Characters List

A 	B
C 	D
E 	F
G 	J
M 	R
T 	

Dennaton Games

[Follow on Twitter](#)

Dennaton (@dennaton) · Jul 7, 2019
巨大的白色一卡通插画，使用超过许多张照片 - (Dennaton) · 12h

Dennaton (@dennaton) · Jul 7, 2019
Dennaton (@dennaton) · 12h
一组儿童的插画 - 作者: LorenAweez

Academic Project About Hotline Miami
Web Development ESAT
Made by: F. Javier Gimeno Ortiz (student)
Images rights reserved to:
Dennaton Games / Devolver Digital

- When the user clicks on to any enement that depicts a character, an overlay will open showing info on said character.

Manny Pardo

Manny Pardo is a character in Hotline Miami 2: Wrong Number. He is a corrupt police officer who becomes embroiled in the violent events of the game. Pardo's character provides a unique perspective on the dark and gritty world of Hotline Miami, blurring the lines between law enforcement and criminality.

Hotline Miami 2: Wrong Number

Manny Pardo's story in Hotline Miami 2: Wrong Number revolves around his career as a police officer and his involvement in the criminal underworld. As a corrupt cop, Pardo participates in violent activities and engages in morally questionable actions, using his position of authority to further his own interests. Throughout the game, Pardo's path intersects with various characters and factions, exposing him to a web of violence, betrayal, and deception. He finds himself entangled with the masked killers and other criminal elements, blurring the boundaries between his duty as an officer and his personal desires. Pardo's story arc explores themes of corruption, moral ambiguity, and the consequences of one's actions. As players experience Pardo's perspective, they are confronted with the ethical dilemmas he faces and the internal struggle between his professional obligations and his personal motivations. Ultimately, Pardo's choices and actions shape the outcome of the game's narrative, influencing the fates of both himself and those around him. His story serves as a chilling exploration of the dark underbelly of society and the lengths some are willing to go for power, control, and self-preservation.

Manny Pardo



Position: Detective
Status: Dead (Atomic Blast)
Eye colour: Green
Hair colour: Blonde
Levels: Homicide, Dead Ahead, Caught
Cutscenes: Final Cut, First Trial, Withdrawal, Death Wish, The Bar Of...

Made by: F. Javier Gimeno Ortiz (student)
Images rights reserved to:
Dennaton Games / Devolver Digital

- Contact Page

- Due to time constraints, the form is non functional, so for now, it opens a local mail client instead.

The screenshot shows two web pages side-by-side. On the left is the 'Contact' page, which has a form with fields for First name, Last name, Company, E-mail, and a message area. A checkbox at the bottom states: 'By clicking this, you confirm that you've read the privacy policy in its entirety (Required.)'. On the right is the 'Privacy Policy' page, which contains detailed legal text about data processing, rights, and sharing. Below the policy text, there is copyright information: 'Academic Project About Hotline Miami Web Development ESAT Made by: F. Javier Gimeno Ortiz (Student) Images rights reserved to: Denimation Games / Devolver Digital'.

Web Overview

Browser Requirements

- Javascript enabled: This Webpage relies heavily on JavaScript, without it, it will not work.
- Due to previous reasons, the web may perform poorly on low end hardware, specially notable on lower end phones.
- Good internet connection: Due to the vast quantity of images that need to be downloaded, a good connection is required.
- Compatible with all the major browsers.

Menu

- Header Menu Guide



To navigate this menu, you should do as follows:

- If you want to go back to the main page, click on the text that says "Hotline Miami Wiki".
- If instead, you want to go to any other page, hover your mouse over any of the three categories, and then click on the one you're interested on.

- Header Menu Technical Info

The Menu was built as follows

- Menu itself is a 'div' element that has an id value of 'navbar' and a class value of 'header'
- Said div contains:
 - A link element that calls 'ChangeURL()' javascript function, it is used as a link to the main page.
 - A div element used as a collapsible menu, it contains the links to other webpages on the site, this element is quite similar to the parent one, the only difference being the stylesheet addressed to it.
 - A p element used to hold the current date of the system, managed by JS (more on that later).
- There are actually two headers in my page, one that stays at the top of it, and one that moves with the window x scroll value, both of them work the same way, except for the addition of a javascript function that hides or shows the other header.
- Javascript Functions used

The screenshot shows a snippet of CSS and JS code. The CSS part includes a background image with three colored dots (red, yellow, green) and a class definition for 'Header Hider'. The JS part is a variable named 'prevScollpos' set to 'window.pageYOffset'.

```

/*Header Hider*/
var prevScollpos = window.pageYOffset;

```

```

var mainContent = document.getElementsByClassName("main")[0];

document.addEventListener('scroll', function(e) {
    var currentScrollPos = window.pageYOffset;
    if (currentScrollPos <= 10) {
        document.getElementById("navbar").style.top = "-50px";
    } else {
        document.getElementById("navbar").style.top = "0";
    }
    prevScrollpos = currentScrollPos;
});

```

This is the code that hides/shows the second header when the user scrolls.

Code	Description
var prevScrollpos = window.pageYOffset;	This line declares a variable <code>prevScrollpos</code> and assigns it the initial vertical scroll position of the page.
var mainContent = document.getElementsByClassName("main")[0];	This line selects the first element with the class name "main" and assigns it to the variable <code>mainContent</code> . It assumes there is at least one element with the class name "main" in the document.
document.addEventListener('scroll', function(e) { ... });	This line adds an event listener to the document for the "scroll" event. When the user scrolls, the function defined within the curly braces { ... } will be executed.
var currentScrollPos = window.pageYOffset;	Inside the scroll event function, this line retrieves the current vertical scroll position of the page and assigns it to the variable <code>currentScrollPos</code> .
if (currentScrollPos <= 10) { ... } else { ... }	This conditional statement checks if the <code>currentScrollPos</code> is less than or equal to 10 pixels. If it is, the code block within the first set of curly braces { ... } is executed. Otherwise, the code block within the second set of curly braces { ... } is executed.
document.getElementById("navbar").style.top = "-50px";	If the scroll position is less than or equal to 10 pixels, this line sets the <code>top</code> CSS property of the element with the id "navbar" to "-50px". This effectively hides the header by moving it above the visible portion of the page.
document.getElementById("navbar").style.top = "0";	If the scroll position is greater than 10 pixels, this line sets the <code>top</code> CSS property of the element with the id "navbar" to "0". This brings the header back to its original position, making it visible again.
prevScrollpos = currentScrollPos;	Finally, this line updates the <code>prevScrollpos</code> variable with the current scroll position, so it can be used in the next scroll event to track the previous scroll position.



```

function DateHandler() {
    let dateP = document.getElementsByClassName("date")[0];
    const date = new Date();

    let months = new Array();
    months[0] = "January";
    months[1] = "February";
    months[2] = "March";
    months[3] = "April";
    months[4] = "May";
    months[5] = "June";
    months[6] = "July";
    months[7] = "August";
    months[8] = "September";
    months[9] = "October";
    months[10] = "November";
    months[11] = "December";

    let days = new Array();
    days[0] = "Sunday";
    days[1] = "Monday";
    days[2] = "Tuesday";
    days[3] = "Wednesday";
    days[4] = "Thursday";
    days[5] = "Friday";
    days[6] = "Saturday";

    dateP.innerHTML = "Today is " + days[date.getDay()] + ", " + date.getDate() + " of " +
months[date.getMonth()] + " of " + date.getFullYear() + "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;";
}

```

This is the code that updates the date on the header

Code	Description
function DateHandler() { ... }	This is the declaration of a function named DateHandler .
let dateP = document.getElementsByClassName("date") [0];	This line declares a variable dateP and assigns it the first element with the class name "date". It assumes there is at least one element with the class name "date" in the document.
const date = new Date();	This line creates a new Date object and assigns it to the constant variable date . This object represents the current date and time.
let months = new Array();	This line declares a variable months and initializes it as an empty array.
months[0] = "January";	This line assigns the string "January" to the first element of the months array.
let days = new Array();	This line declares a variable days and initializes it as an empty array.
days[0] = "Sunday";	This line assigns the string "Sunday" to the first element of the days array.
dateP.innerHTML = "Today is " + days[date.getDay()] + ", " + date.getDate() + " of " + months[date.getMonth()] + " of " + date.getFullYear() + " ";	This line sets the HTML content of the element referenced by dateP . It creates a formatted string that displays the current date, including the day of the week, day of the month, month, and year.

```
● ● ●

function ChangeURL(url, mouseBtn) {
    if(url == "random")
    {
        switch(Math.round(5 * Math.random()))
        {
            case 1:
                document.location.href = './characters.html';
                break;
            case 2:
                document.location.href = './weapons.html';
                break;
            case 3:
                document.location.href = './contact.html';
                break;
            case 4:
                document.location.href = './index.html';
                break;
            case 5:
                document.location.href = './roll.html';
                break;
        }
    }
    else if(url == './cv/index.html'/* || ( mouseBtn != 'undefined' && mouseBtn.button == 1)*/)
    {
        window.open(url, '_blank');
    }
    else
    {
        document.location.href = url;
    }
}
```

This is the code that changes the document page.

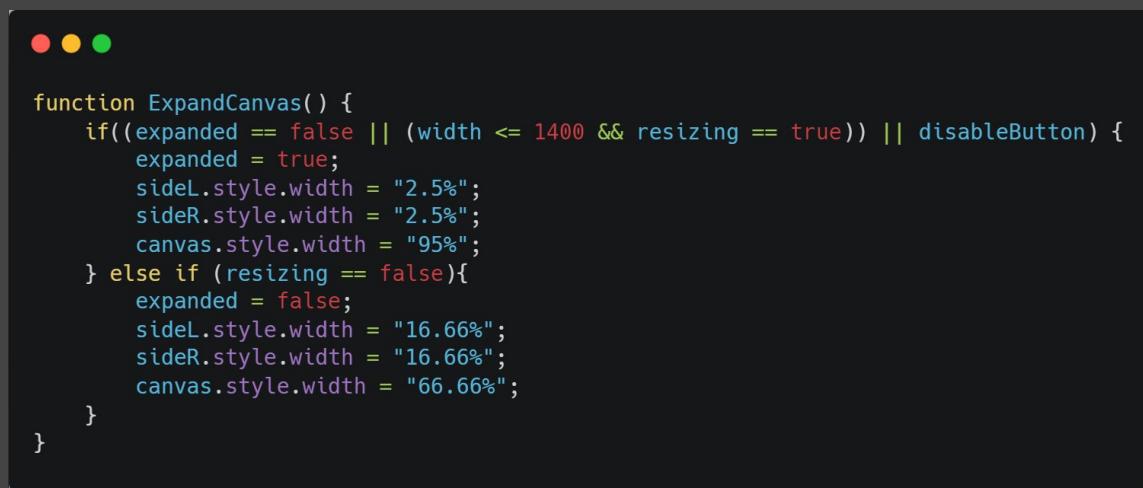
Code	Description
ChangeURL(url, mouseBtn)	This function is used to change the URL of the current page based on the provided parameters.
url == "random"	If the URL parameter is set to "random", a random page will be chosen to navigate to.
Math.round(5 * Math.random())	This expression generates a random number between 1 and 5.
case 1:	If the random number is 1, the page will be redirected to './characters.html'.
case 2:	If the random number is 2, the page will be redirected to './weapons.html'.

case 3:	If the random number is 3, the page will be redirected to './contact.html'.
case 4:	If the random number is 4, the page will be redirected to './index.html'.
case 5:	If the random number is 5, the page will be redirected to './roll.html'.
url == './cv/index.html'	If the URL parameter is './cv/index.html', a new browser window/tab will be opened with the specified URL.
window.open(url, '_blank')	This function opens the specified URL in a new browser window/tab.
else	If none of the above conditions are met, the page will be redirected to the specified URL.

Page Other Functions

- **Page expand Button**

Every page has a button on the top left corner that is used to expand the canvas of the webpage, basically giving it a zoom like effect. to deal with this, another javascript function was used, said function works as follows:



```

function ExpandCanvas() {
    if((expanded == false || (width <= 1400 && resizing == true)) || disableButton) {
        expanded = true;
        sideL.style.width = "2.5%";
        sideR.style.width = "2.5%";
        canvas.style.width = "95%";
    } else if (resizing == false){
        expanded = false;
        sideL.style.width = "16.66%";
        sideR.style.width = "16.66%";
        canvas.style.width = "66.66%";
    }
}

```

Code	Description
function ExpandCanvas()	This function is used to expand or collapse the canvas element based on certain conditions.
(expanded == false (width <= 1400 && resizing == true)) disableButton	This condition checks if either the variable <code>expanded</code> is false or if the width of the canvas is less than or equal to 1400 pixels and the <code>resizing</code> variable is true. It also checks if the <code>disableButton</code> variable is true.
expanded = true;	If the condition is met, the <code>expanded</code> variable is set to true.
sideL.style.width = "2.5%";	This line sets the <code>width</code> CSS property of the element with the id "sideL" to "2.5%".
sideR.style.width = "2.5%";	This line sets the <code>width</code> CSS property of the element with the id "sideR" to "2.5%".
canvas.style.width = "95%";	This line sets the <code>width</code> CSS property of the element with the id "canvas" to "95%".
else if (resizing == false)	If the first condition is not met, this condition checks if the <code>resizing</code> variable is false.
expanded = false;	If the second condition is met, the <code>expanded</code> variable is set to false.
sideL.style.width = "16.66%";	This line sets the <code>width</code> CSS property of the element with the id "sideL" to "16.66%".
sideR.style.width = "16.66%";	This line sets the <code>width</code> CSS property of the element with the id "sideR" to "16.66%".
canvas.style.width = "66.66%";	This line sets the <code>width</code> CSS property of the element with the id "canvas" to "66.66%".

- **Main Page Gallery**

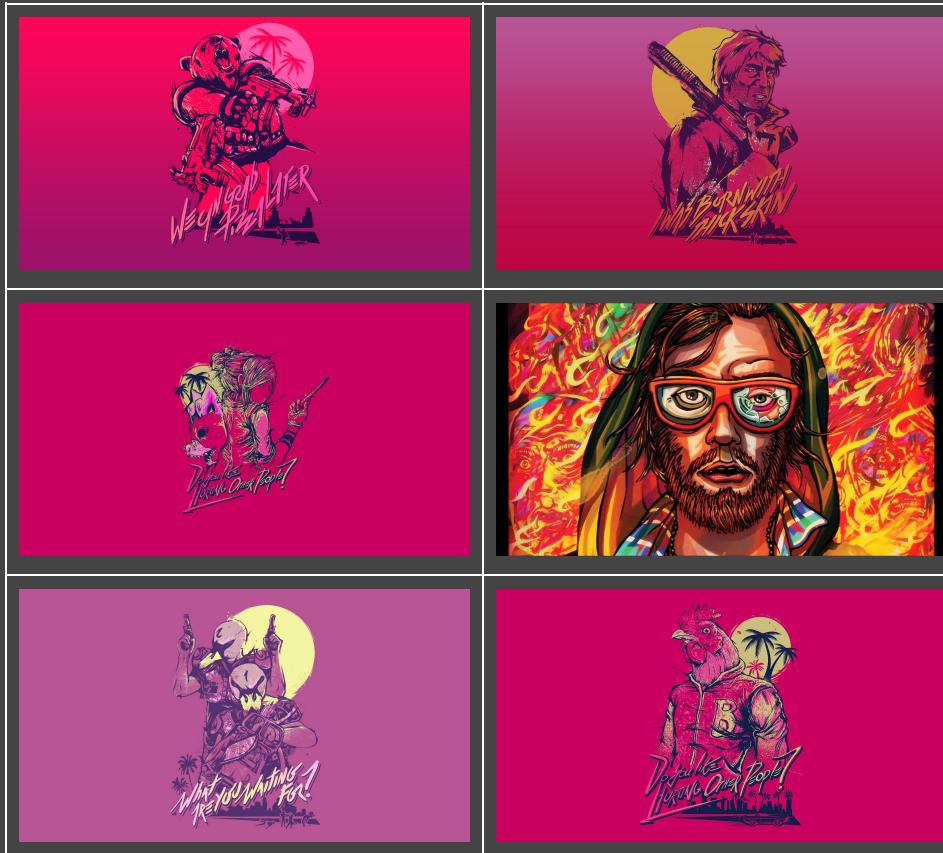
For the creation of the gallery seen on the main page, i decided to modify an existing code from W3Schools to accomodate my needs. Said code can be found [Here](#), that's why I won't get into much details, that being said, my main changes where the removal of the bar with the image previews, and the addition to randomize the starting image.

- **Twitter Feed**

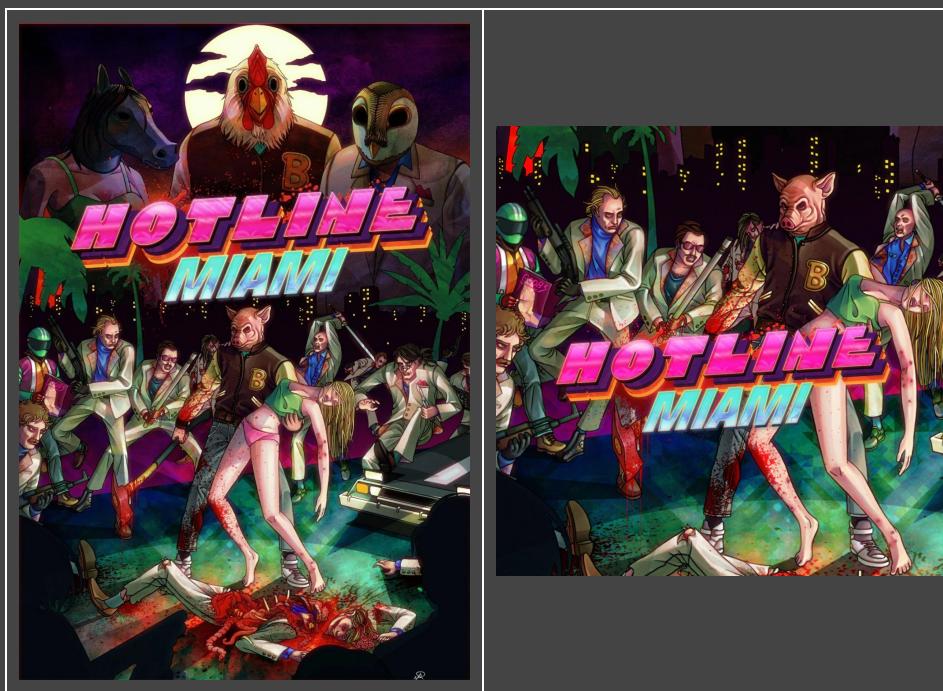
I also decided to embed the Twitter feed of the developers into the page, this was done following the instructions on [this](#) webpage.

Bibliography

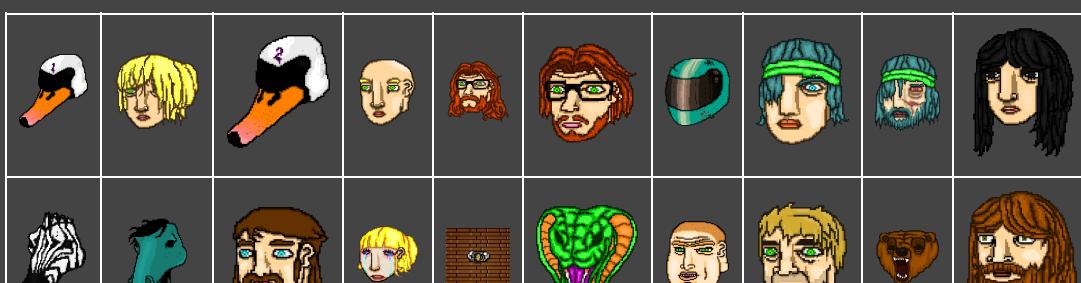
- Images



This images are all over the net, no assuming they're not original art, the artist is no where to be found, tried doing a reverse image search to try and get the original author, but had no luck.



Official Game Art





Original Game Art, extracted from hotlinemiami.fandom.com
FANDOM GAMES COMMUNITY. Hotline Miami 2 characters. Hotline Miami Wiki [en línea]. [sin fecha]
[consultado el 23 de mayo de 2023]. Disponible en:
https://hotlinemiami.fandom.com/wiki/Category:Hotline_Miami_2_characters