

Alexander Wohlfahrt

Farid Abdalla

Celien donze

Projet Django

03.04.2019

He-arc

David Grunenwald



Sommaire

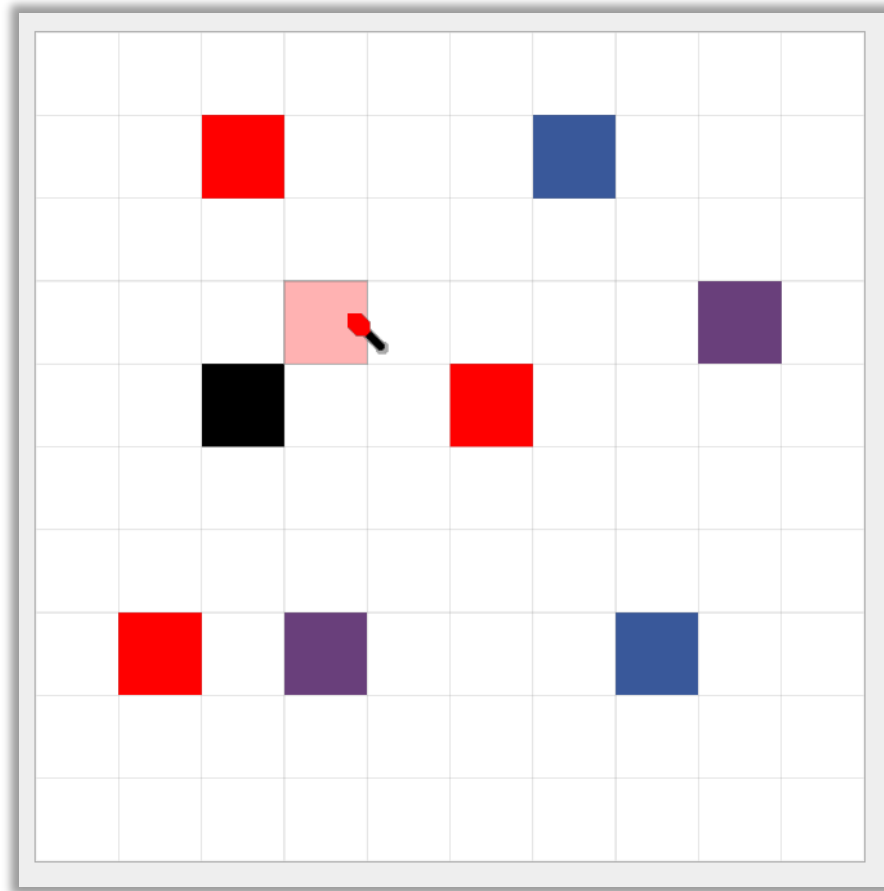
- Introduction
- Démonstration
- Canvas
- Crispy
- Boutique
- Conclusion
- Stripe

INTERFACED



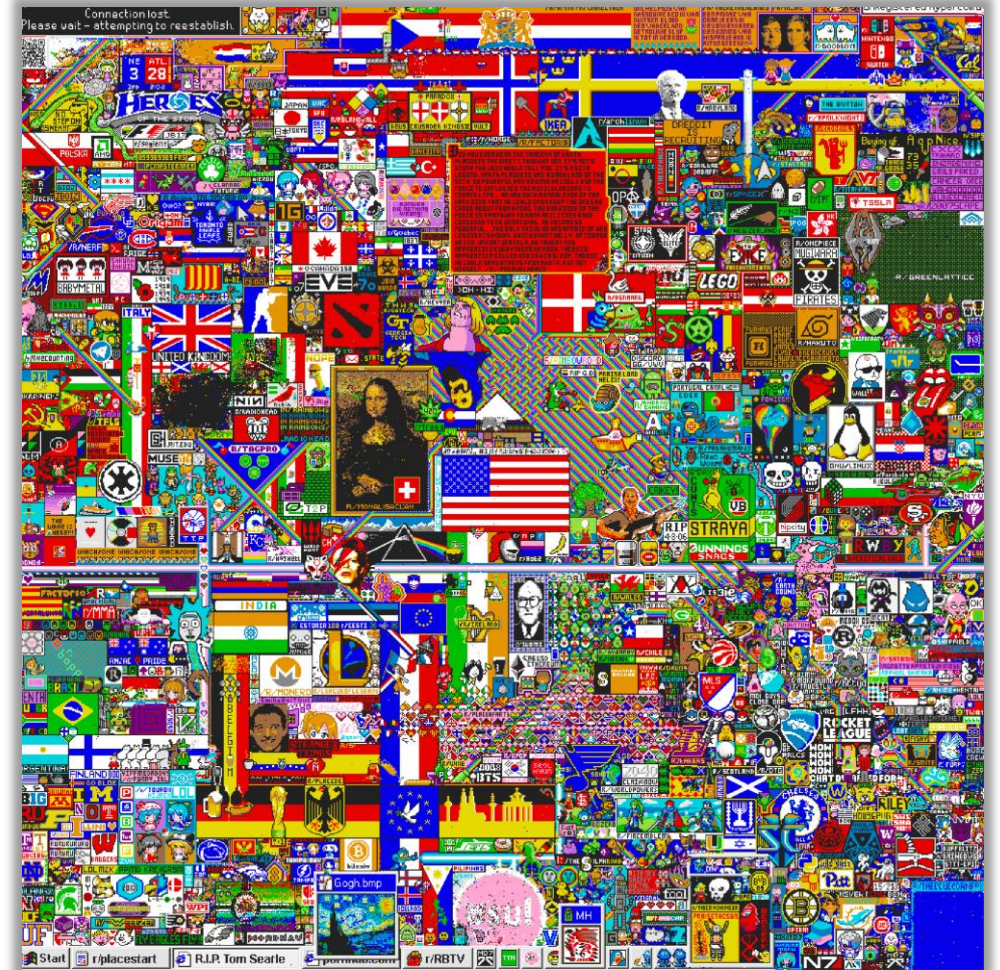
resume

- Canvas
- Communauté
- Paiement (Stripe)



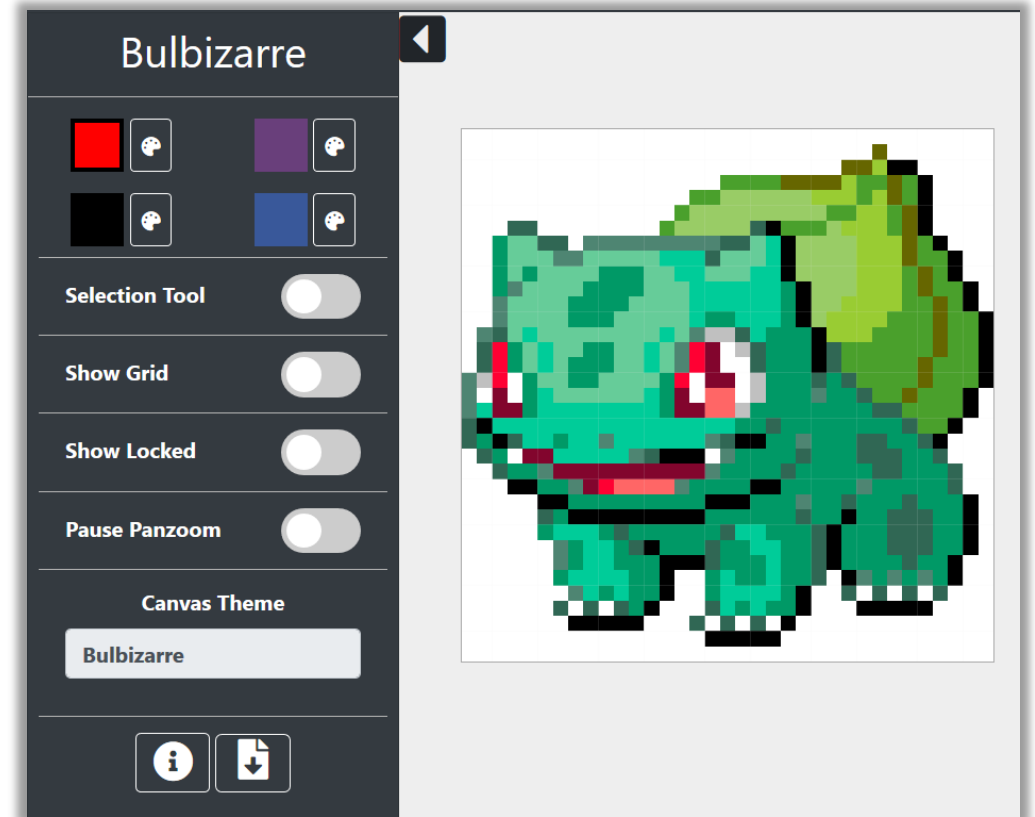
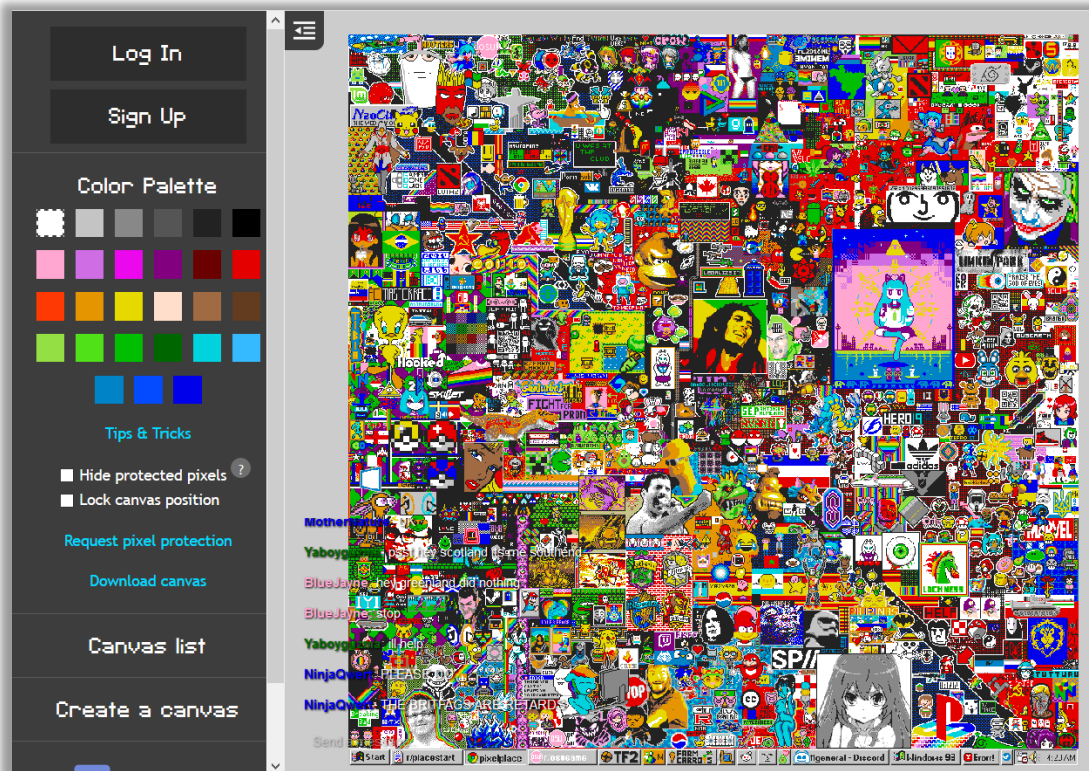
inspirations

- <https://www.reddit.com/r/place/>
1 avril 2017 – 72 heures
- Beaucoup d'autres clones existent



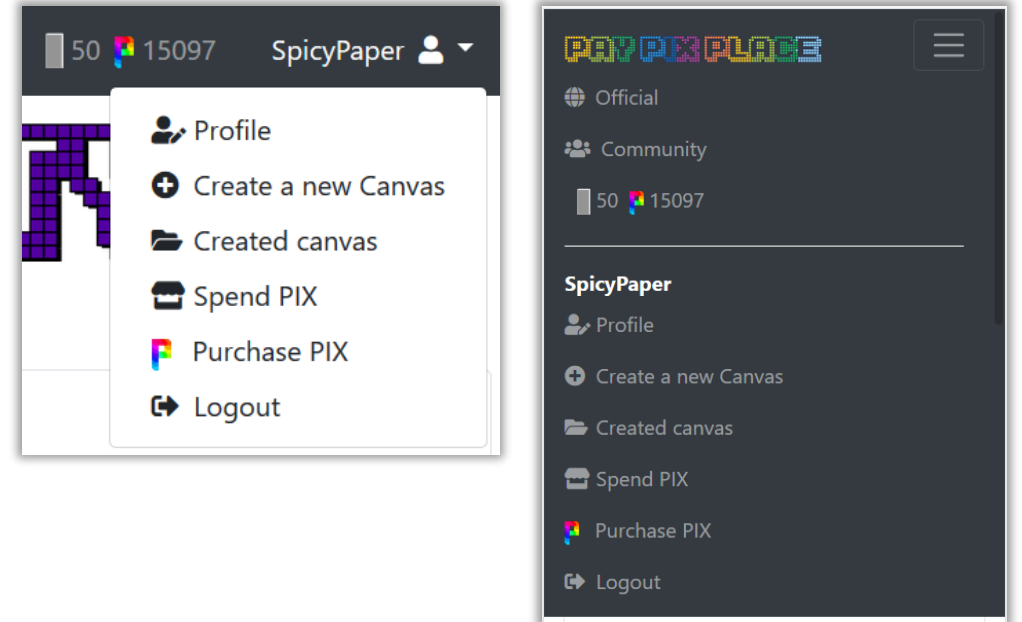
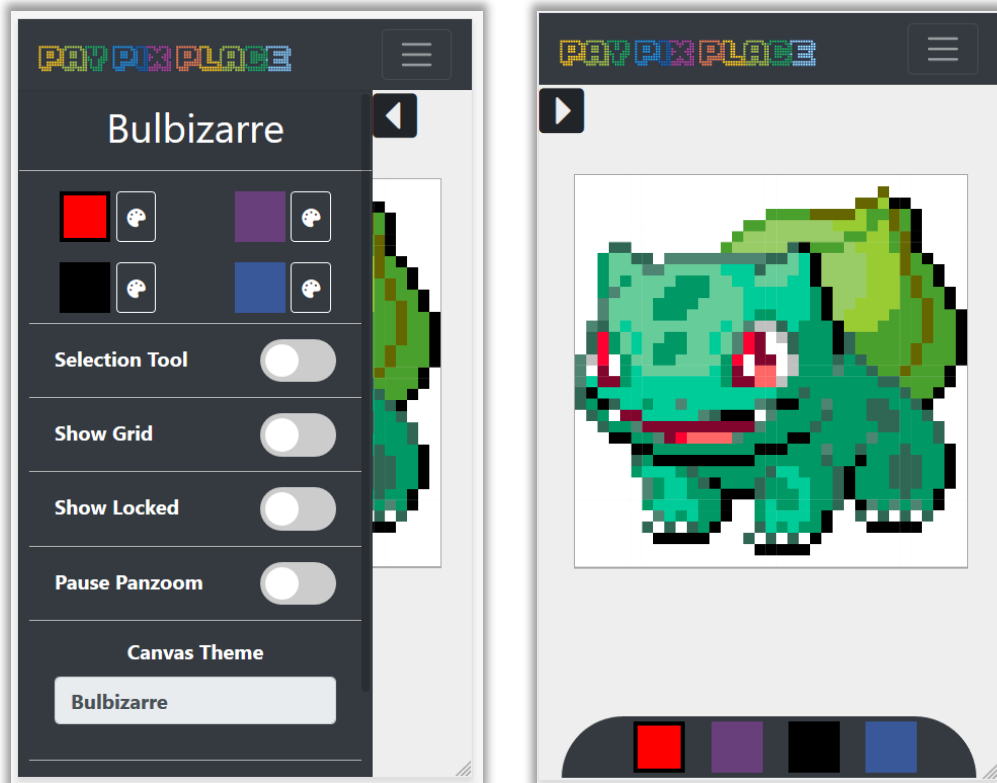
inspirations

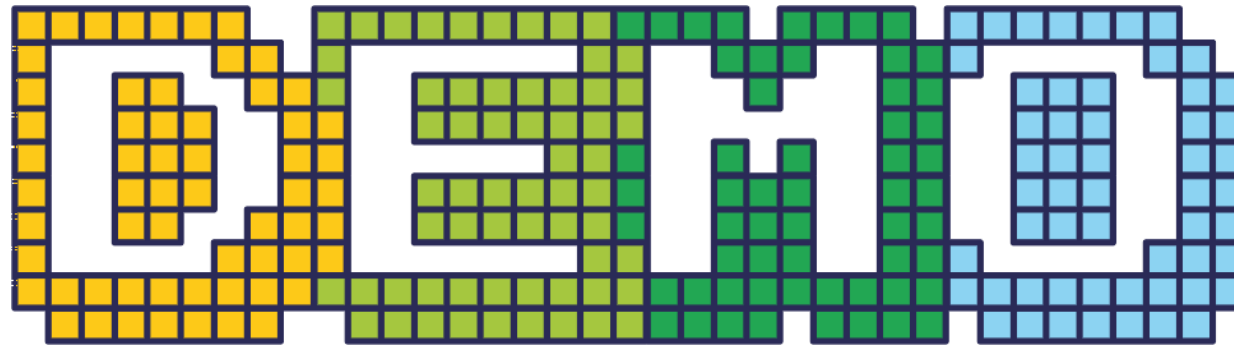
- <https://pixelplace.io/1>
inspiration principale



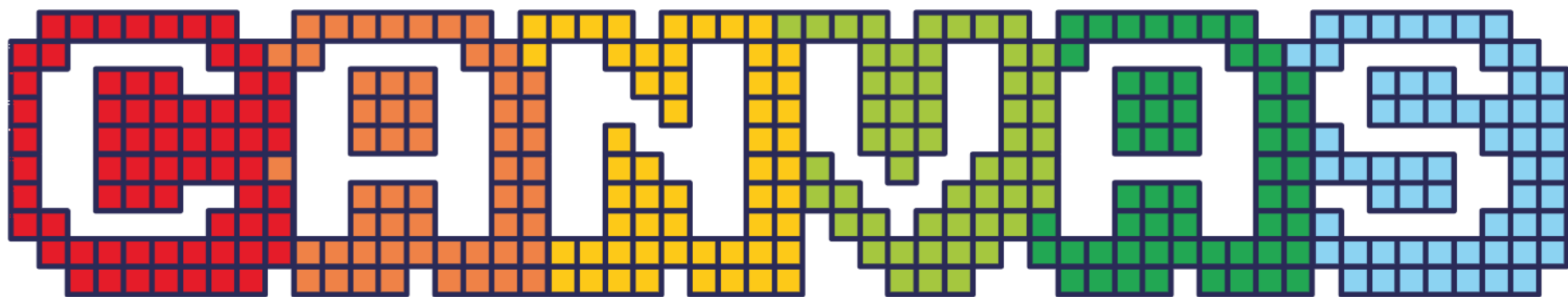
Mobile first

- Responsive design





<https://paypixplace.srvz-webapp.he-arc.ch/>





Actualisation des donnees

- Toutes les 4 secondes
- AJAX
- Actualise tous les pixels du canvas
- 3 URI par canvas
 - /canvas/<int:id>/ → interface de dessin
 - /canvas/<int:id>/json/ → données du canvas
 - /canvas/<int:id>/img/ → image PNG

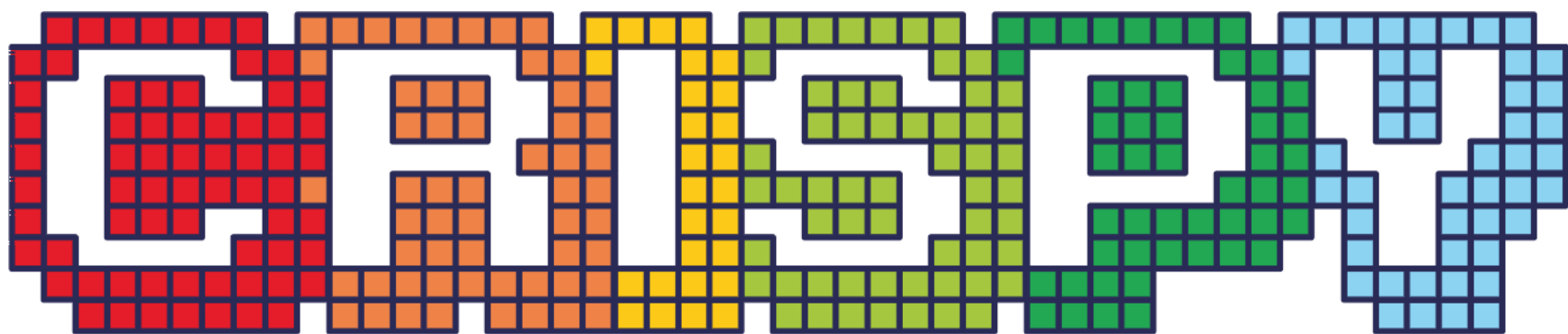


Panzoom

- Très simple à mettre en place
- Desktop + Mobile
- Zoom
- Déplacement du canvas

```
panzoomInstance = panzoom(  
  canvas,  
  {  
    maxZoom: 10,  
    minZoom: 0.05,  
    smoothScroll: false,  
    zoomDoubleClickSpeed: 1  
  }  
);  
  
panzoomInstance.zoomAbs(  
  0,    // initial x position  
  0,    // initial y position  
  0.1   // initial zoom  
);
```

- Source : <https://github.com/anvaka/panzoom>





Resume

```
pip install --upgrade django-crispy-forms
```

```
INSTALLED_APPS = (  
    ...  
    'crispy_forms',  
)
```

```
CRISPY_TEMPLATE_PACK = 'bootstrap4'
```

```
{% load crispy_forms_tags %}
```

```
<form method="post">  
    {{ my_formset|crispy }}  
</form>
```

- Source : <https://django-crispy-forms.readthedocs.io/en/latest/>



Exemple

- {{ form }}

Register

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. First name:

Last name:

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.



Exemple

- `{% load crispy_forms_tags %}`
- `{{ form|crispy }}`

Register

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

Last name

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.




Exemple

- {% load crispy_forms_tags %}
- {% crispy form %}
- Personnalisation


Register

Username*




Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.


First name




Last name



Email*




Password*



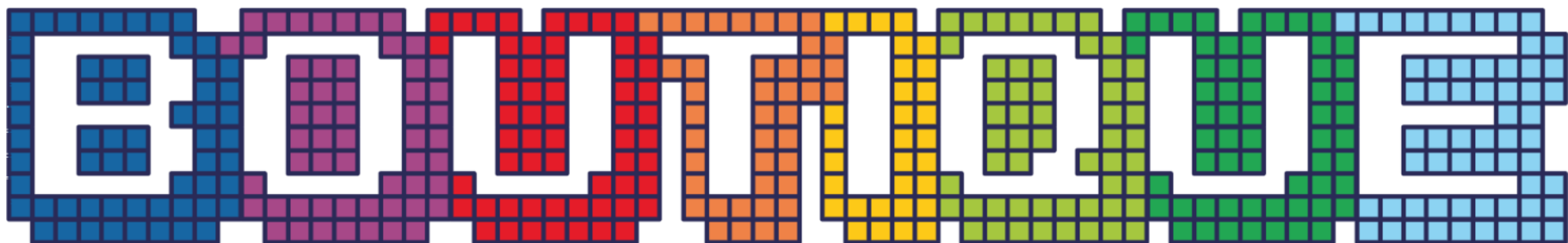
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*



Enter the same password as before, for verification.

Register



Boutique

- Accessible partout sur le site
- Modal Bootstrap
- Bootstrap-color-picker





securite

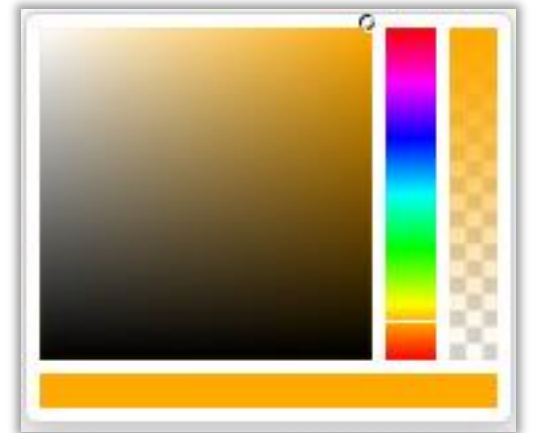
- Vérification du solde
- Fonctionne avec des ID
→ Impossible de tromper le serveur
- Impossible d'acheter deux fois la même couleur
- URI pour acheter
`/buy/<int:id>/`

✓ I'm over here !

plugins

- Bootstrap Colorpicker
- Officiel : <https://github.com/farbelous/bootstrap-colorpicker>
- Exemples : <https://farbelous.io/bootstrap-colorpicker/v2/>
- CDN : <https://cdnjs.com/libraries/bootstrap-colorpicker>
- Notify.js
- Source : <https://notifyjs.jpillora.com/>

```
$.notify("I'm over here !");
```



CODES



Problemes rencontres

- Mouvement et zoom du canvas
→ Panzoom



Améliorations possibles

- Optimiser les échanges pour le dessin
 - Websocket
 - Canvas beaucoup plus grand
- Sélection de pixels par zone
 - Dessin
 - Verrouillage
- Chat
 - Par canvas
 - Publique



Mot de la fin

- Projet fonctionnel et aucun bug connu
- Fonctionnalités minimums et visées réalisées
→ sauf canvas privés
- Fonctionnalités supplémentaires réalisées
→ sauf pack couleurs par canvas

WERNER

QUESTIONS

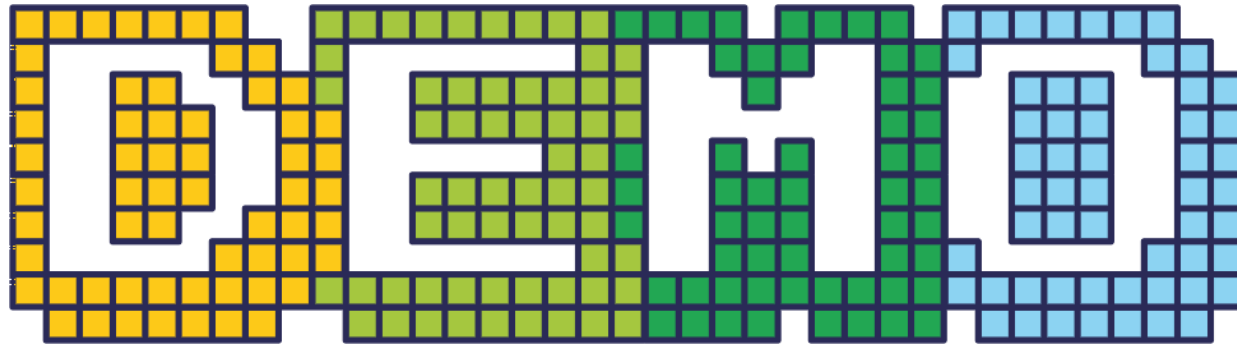
REMARKS

STAYSAFE



resume

- API de paiement par carte bancaire
- Disponible en
 - Ruby
 - Python
 - Php
 - Java
 - ...
- Utilisé par Amazon, Wish, Twitch...
- Tarifs 2.9% + 0.30 Fr



<https://github.com/HE-Arc/TutoStripe>



Installation

- Ruby : `gem install stripe`
- Python : `pip install stripe`
- Php : `composer require stripe/stripe-php`
- Java (maven) :
 `<dependency>`
 `<groupId>com.stripe</groupId>`
 `<artifactId>stripe-java</artifactId>`
 `<version>8.1.0</version>`
 `</dependency>`
- Java (gradle) : `compile "com.stripe:stripe-java:8.1.0"`



Cote client - de base

```
1 <form action="your-server-side-code" method="POST">
2   <script
3     src="https://checkout.stripe.com/checkout.js" class="stripe-button"
4     data-key="pk_test_TYooMQauvdEDq54NiTphI7jx"
5     data-amount="999"
6     data-name="Stripe.com"
7     data-description="Example charge"
8     data-image="https://stripe.com/img/documentation/checkout/marketplace.png"
9     data-locale="auto"
10    data-zip-code="true">
11  </script>
12 </form>
```





Cote client - personnalise

```
1 <script src="https://checkout.stripe.com/checkout.js"></script>
2
3 <button id="customButton">Purchase</button>
4
5 <script>
6 var handler = StripeCheckout.configure({
7   key: 'pk_test_TYooMQauvdEDq54NiTphI7jx',
8   image: 'https://stripe.com/img/documentat
9   locale: 'auto',
10  token: function(token) {
11    // You can access the token ID with `to
12    // Get the token ID to your server-side
13  }
14 });
15
```

```
16 document.getElementById('customButton').addEventListener('click', function(e) {
17   // Open Checkout with further options:
18   handler.open({
19     name: 'Stripe.com',
20     description: '2 widgets',
21     zipCode: true,
22     amount: 2000
23   });
24   e.preventDefault();
25 });
26
27 // Close Checkout on page navigation:
28 window.addEventListener('popstate', function() {
29   handler.close();
30 });
31 </script>
```



Token

THE TOKEN OBJECT

```
{
  "id": "tok_1EKQBF2eZvKYlo2CQ8iOzLVk",
  "object": "token",
  "card": {
    "id": "card_1EKQBF2eZvKYlo2CGgg1nNtL",
    "object": "card",
    "address_city": null,
    "address_country": null,
    "address_line1": null,
    "address_line1_check": null,
    "address_line2": null,
    "address_state": null,
    "address_zip": null,
    "address_zip_check": null,
    "brand": "Visa",
```

```
    "country": "US",
    "cvc_check": null,
    "dynamic_last4": null,
    "exp_month": 8,
    "exp_year": 2020,
    "fingerprint": "Xt5EWLLDS7FJjR1c",
    "funding": "credit",
    "last4": "4242",
    "metadata": {},
    "name": null,
    "tokenization_method": null
  },
  "client_ip": null,
  "created": 1554125285,
  "livemode": false,
  "type": "card",
  "used": false
}
```



Cote serveur

```
1  # Set your secret key: remember to change this to your live secret key in production
2  # See your keys here: https://dashboard.stripe.com/account/apikeys
3  stripe.api_key = "sk_test_4eC39HqLyjWDarjtT1zdp7dc"
4
5  # Token is created using Checkout or Elements!
6  # Get the payment token ID submitted by the form:
7  token = request.POST['stripeToken'] # Using Django
8
9  charge = stripe.Charge.create(
10     amount=999,
11     currency='usd',
12     description='Example charge',
13     source=token,
14 )
```



Autres options

- Coupons
- Remboursement
- Abonnement
- Plans



Notre implementation

- Gestion du formulaire en Ajax
- Routes pix/purchase/<int:id>/
 - 1 -> 1 CHF
 - 2 -> 2 CHF
 - 3 -> 5 CHF
 - 4 -> 20 CHF

The screenshot shows a mobile application interface for a payment process. At the top, there's a 'Community' header with a user icon and a circular profile picture. Below this, the title 'Pay Pix Place' is displayed, followed by a promotional message '6000 PIX + 2000 PIX bonus !'. The main form area contains several input fields: an email address field with an envelope icon, a card number field with a card icon, and two smaller fields for the expiration date (MM / AA) and CVV (with a lock icon). Below these fields is a checkbox labeled 'Se souvenir de moi'. At the bottom of the form is a large blue button labeled 'Payer 20,00 CHF'.



Retour experience

- Très simple
- Bien documenté
- Très bonne tarification par rapport à l'offre
- Beaucoup d'options
- Vaste proposition de service de paiement

WERNER

QUESTIONS

REMARKS