
Laïka

Que fait Laïka ?

Que fait Laïka ?

Prenez en photo le ciel étoilé présent sous vos yeux

Que fait Laïka ?

Prenez en photo le ciel étoilé présent sous vos yeux

Laïka

Que fait Laïka ?

Prenez en photo le ciel étoilé présent sous vos yeux

Laïka

Et Laïka se charge d'y tracer les constellations présentent

Domaine d'étude de Laïka

Domaine d'étude de Laïka

Ciel Original*

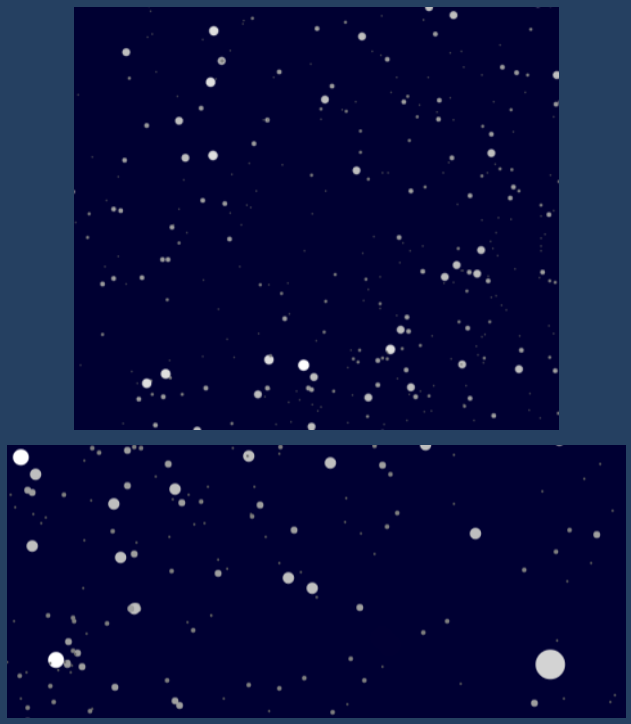
Domaine d'étude de Laïka

Ciel Original*

*Fixé à une heure et journée précise (23h solstice d'hiver) car la position relative des étoiles change au cour du temps.

Domaine d'étude de Laïka

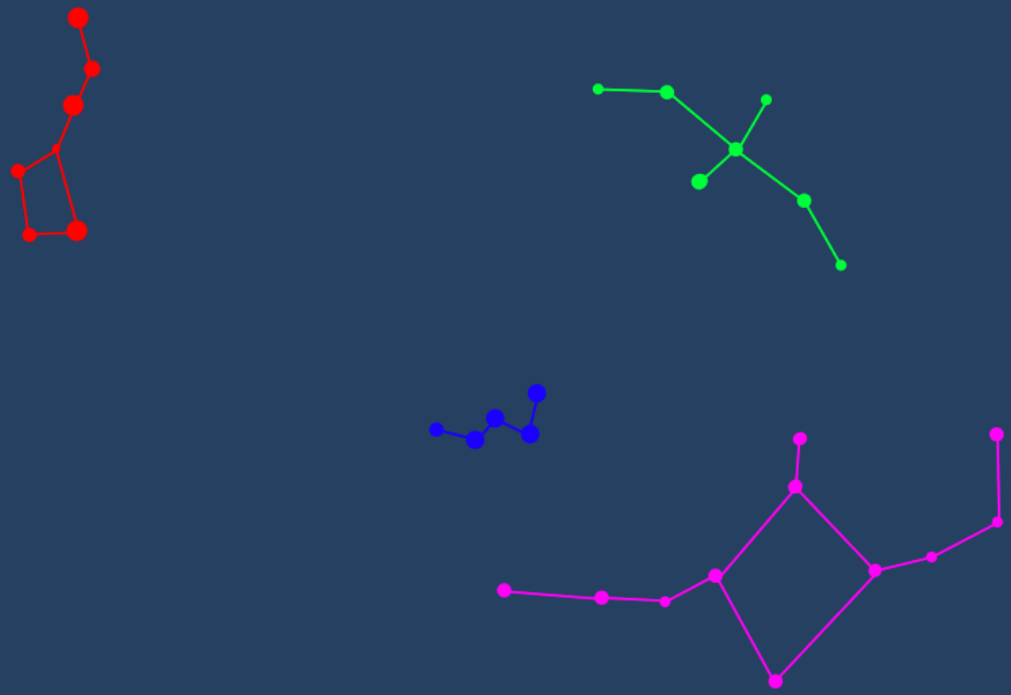
Ciel Original*



Photographies mises à disposition

*Fixé à une heure et journée précise (23h solstice d'hiver) car la position relative des étoiles change au cour du temps.

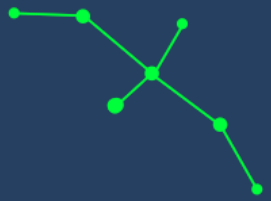
Domaine d'étude de Laïka

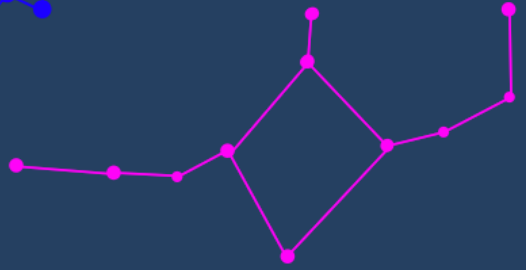


Constellations connues par Laïka

Domaine d'étude de Laïka

La grande Ourse

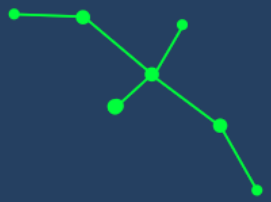




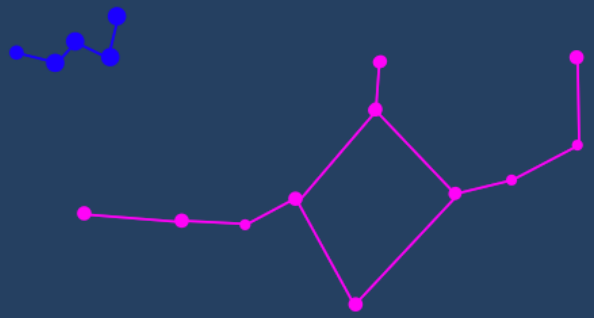
Constellations connues par Laïka

Domaine d'étude de Laïka

La grande Ourse



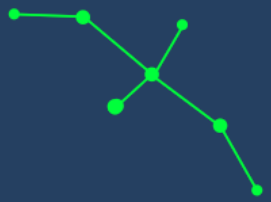
Cygne



Constellations connues par Laïka

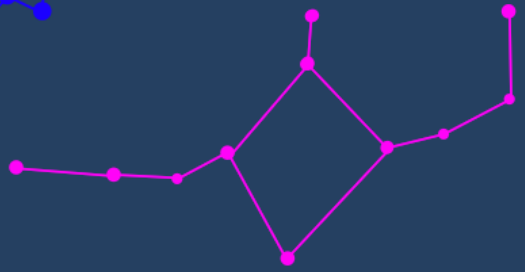
Domaine d'étude de Laïka

La grande Ourse



Cygne

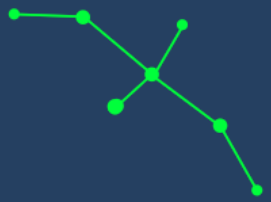
Cassiopée



Constellations connues par Laïka

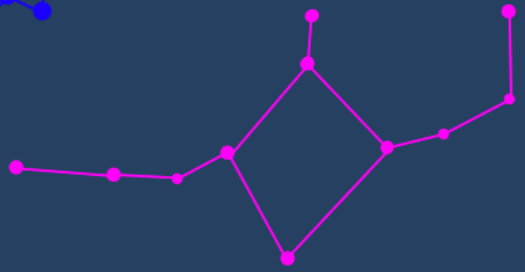
Domaine d'étude de Laïka

La grande Ourse



Cygne

Cassiopée

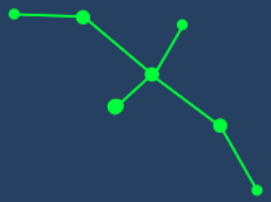


Orion

Constellations connues par Laïka

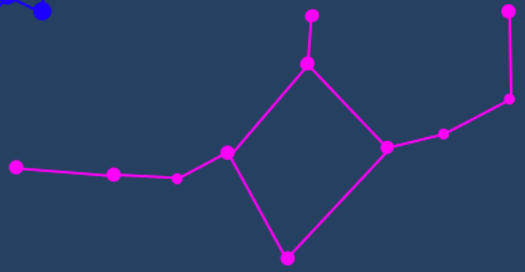
Domaine d'étude de Laïka

La grande Ourse



Cygne

Cassiopée



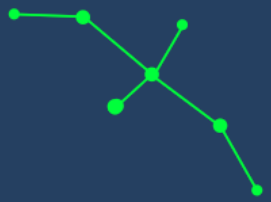
Pégase

Orion

Constellations connues par Laïka

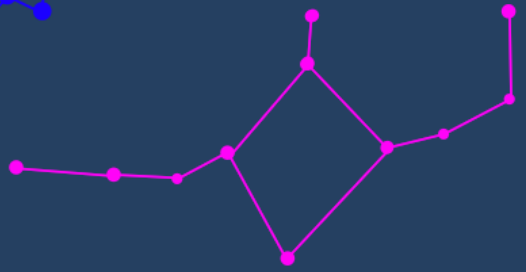
Domaine d'étude de Laïka

La grande Ourse



Cygne

Cassiopée



Pégase

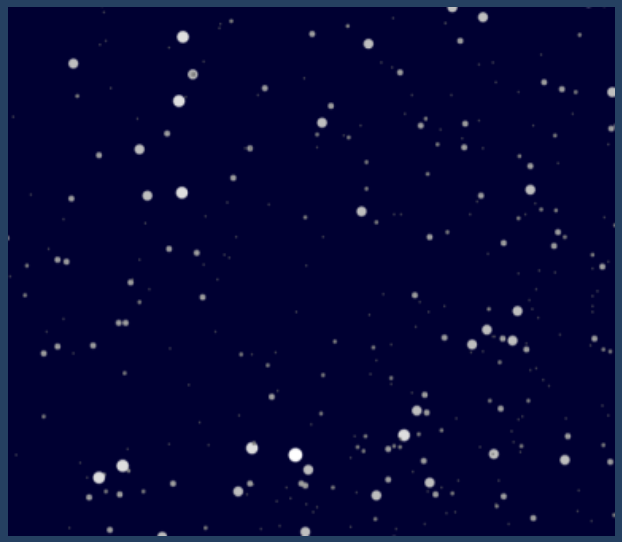
Orion

Constellations connues par Laïka

Laïka ne pourra pas reconnaitre d'autre constellations que celles présentent ici

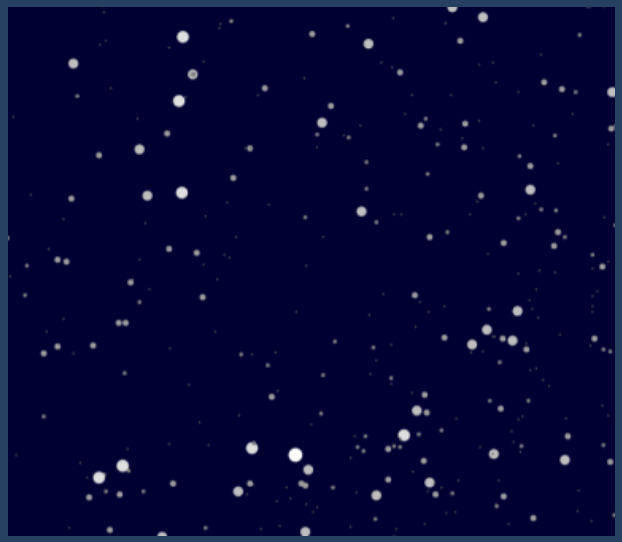
Fonctionnement de Laïka

Fonctionnement de Laïka



1. La photographie donnée par l'utilisateur est transformée en array (définit en rouge, vert, bleu et transparent)

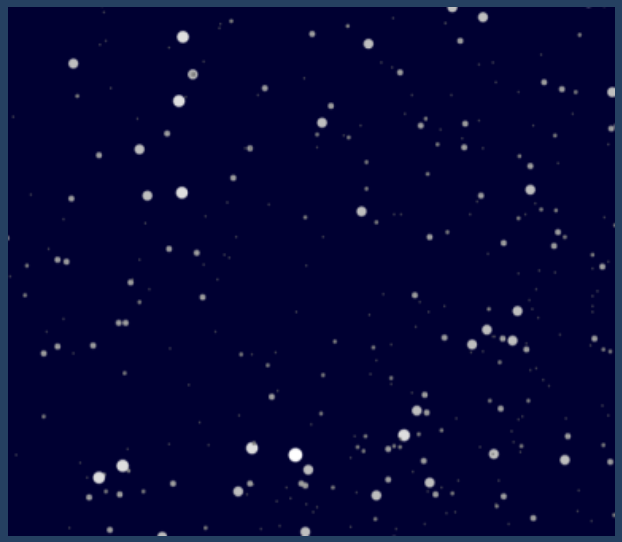
Fonctionnement de Laïka



1. La photographie donnée par l'utilisateur est transformée en array (définit en rouge, vert, bleu et transparent)

2. liste_etoile()

Fonctionnement de Laïka



1. La photographie donnée par l'utilisateur est transformée en array (définit en rouge, vert, bleu et transparent)

2. Liste_etoile()

[Liste_etoiles_ensemble_pixel] = [Etoile1, Etoile2, ... , EtoileN]

Fonctionnement de Laïka

[liste_etoiles_ensemble_pixel]

Fonctionnement de Laïka

[liste_etoiles_ensemble_pixel]

3. caractérisation()

Fonctionnement de Laïka

[liste_etoiles_ensemble_pixel]

3. caractérisation()

[liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

Fonctionnement de Laïka

[Liste_etoiles_ensemble_pixel]

3. Caractérisation()

[Liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

EtoileX = [Barycentre, Couleur_moy]

Fonctionnement de Laïka

[Liste_etoiles_ensemble_pixel]

3. Caractérisation()

[Liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

EtoileX = [Barycentre, Couleur_moy]

+

Suppression des étoiles trop petites

Fonctionnement de Laïka

[Liste_etoiles_ensemble_pixel]

3. Caractérisation()

[Liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

Définitions

EtoileX = [Barycentre, Couleur_moy]

+

Suppression des étoiles trop petites

Fonctionnement de Laïka

[Liste_etoiles_ensemble_pixel]

3. Caractérisation()

[Liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

Définitions

- Barycentre = les coordonnées du centre de notre étoile dans la photographie

EtoileX = [Barycentre, Couleur_moy]

+

Suppression des étoiles trop petites

Fonctionnement de Laïka

[Liste_etoiles_ensemble_pixel]

3. Caractérisation()

[Liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

Définitions

- Barycentre = les coordonnées du centre de notre étoile dans la photographie
- Couleur moyenne = un array retournant les couleur moyenne de rouge, vert, bleu, et transparent

EtoileX = [Barycentre, Couleur_moy]

+

Suppression des étoiles trop petites

Fonctionnement de Laïka

[Liste_etoiles_ensemble_pixel]

3. Caractérisation()

[Liste_etoile_caractérisées] = [Etoile1, Etoile3, ... , EtoileN]

Définitions

- Barycentre = les coordonnées du centre de notre étoile dans la photographie
- Couleur moyenne = un array retournant les couleur moyenne de rouge, vert, bleu, et transparent
- Taille = soit le nombre de pixel qui constitue notre étoile

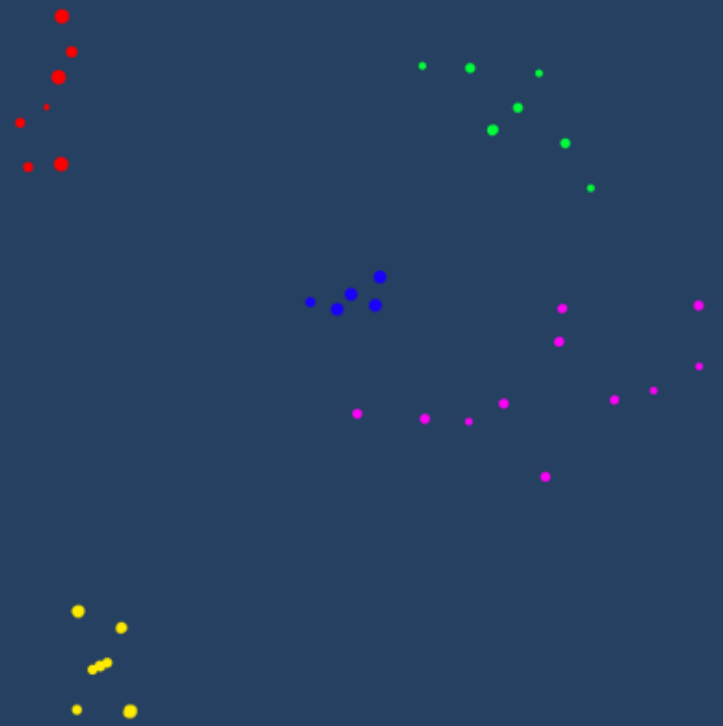
EtoileX = [Barycentre, Couleur_moy]

\+

Suppression des étoiles trop petites

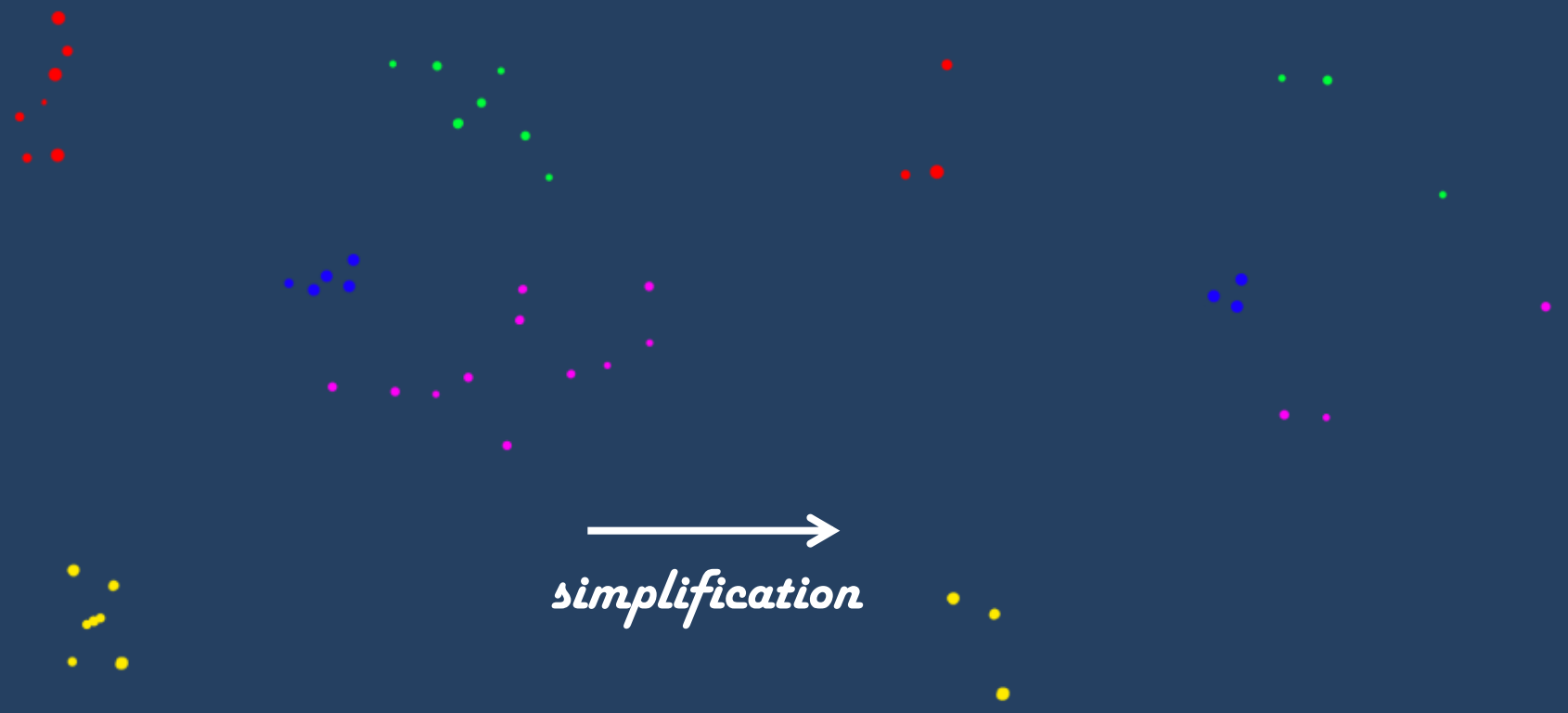
Fonctionnement de Laïka

[Listes des constellations]



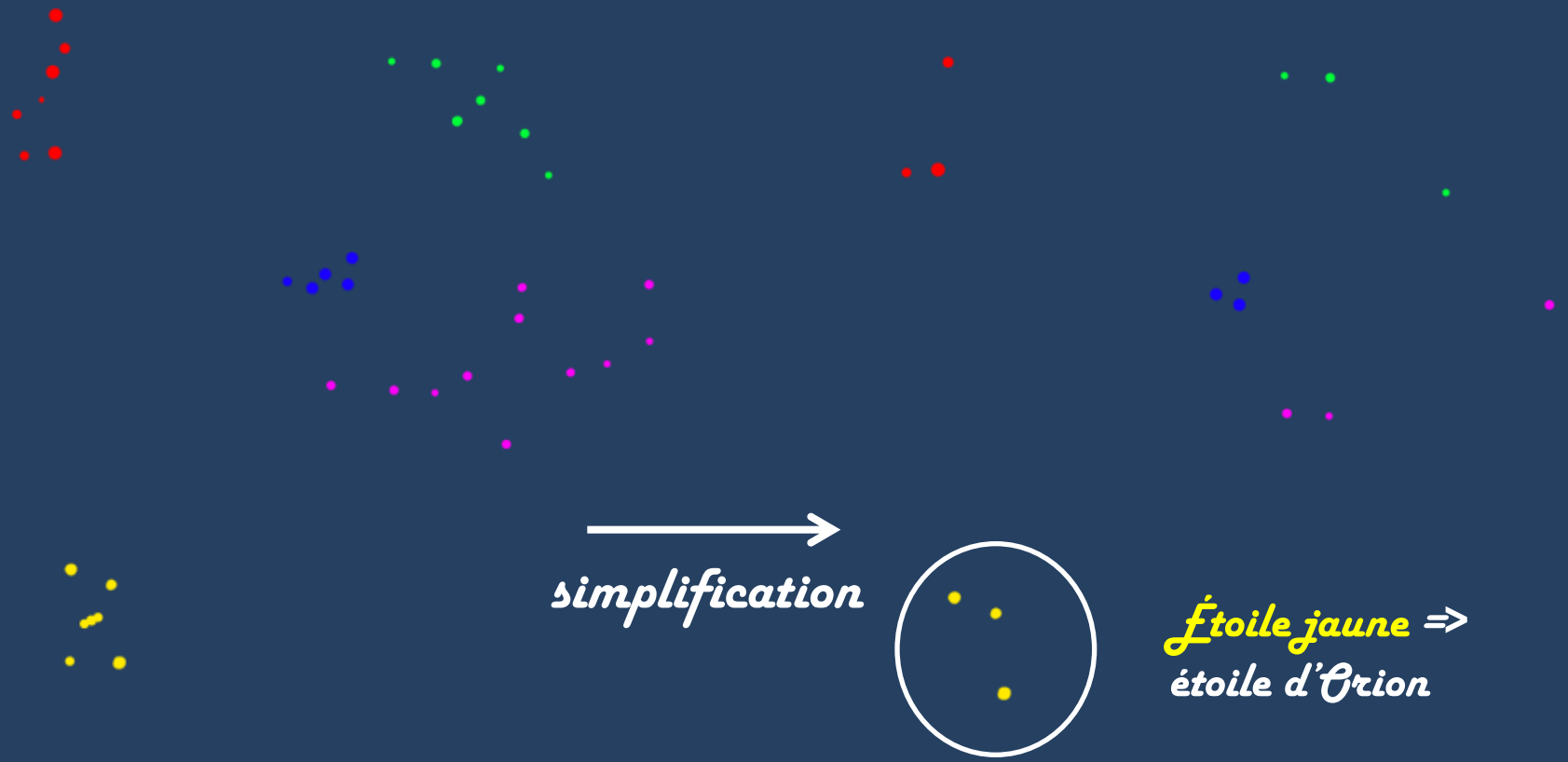
Fonctionnement de Laïka

[Listes des constellations]



Fonctionnement de Laïka

[Listes des constellations]



Fonctionnement de Laïka

[Listes des constellations]

Définitions

Fonctionnement de Laïka

[Listes des constellations]

Définitions

- [Liste de constellation] = [ConstellationRouge, ConstellationVerte, ...]

Fonctionnement de Laïka

[Listes des constellations]

Définitions

- [Liste de constellation] = [ConstellationRouge, ConstellationVerte, ...]

- ConstellationX = [Rapport(Triplet d'étoile de la Constellation), Triplet d'étoile de la Constellation]

Fonctionnement de Laïka

[Listes des constellations]

Définitions

- [Liste de constellation] = [ConstellationRouge, ConstellationVerte, ...]

- ConstellationX = [Rapport(Triplet d'étoile de la Constellation), Triplet d'étoile de la Constellation]

- Rapport() = distance des étoiles les plus proches dans le triplet / la distance maximale

Fonctionnement de Laïka

Fonctionnement de Laïka

[Liste_etoile_caractérisées]

Fonctionnement de Laïka

[Liste_etoile_caractérisées] + [Listes des constellations]

Fonctionnement de Laïka

[Liste_etoile_caractérisées] + [Listes des constellations]

4. Recherche_constellations()

Fonctionnement de Laïka

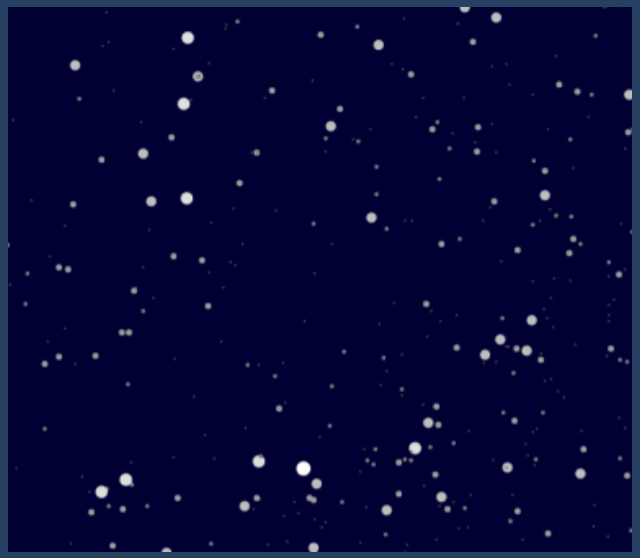
[Liste_etoile_caractérisées] + [Listes des constellations]

4. Recherche_constellations()

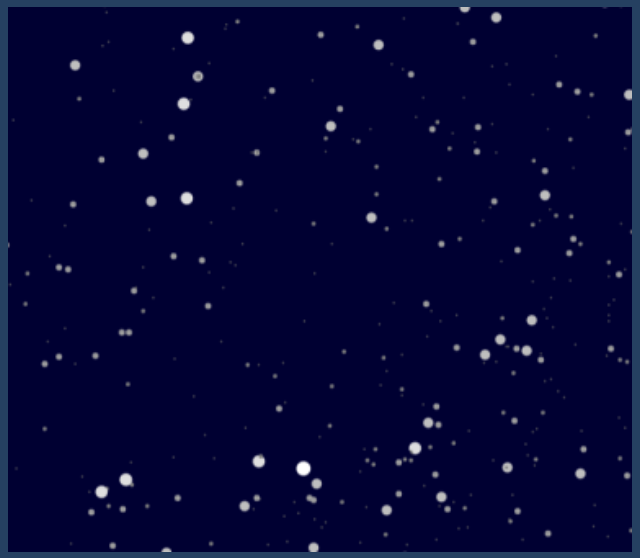
[Listes des constellations contenue dans la photographie] = [ConstellationX, ConstellationY, ...] ou [ ]

Fonctionnement de Laïka

Fonctionnement de Laïka

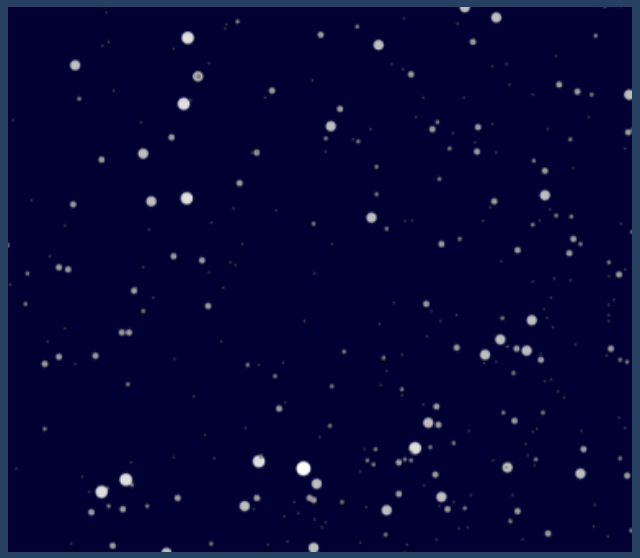


Fonctionnement de Laïka



[Listes des constellations contenue dans la photographie]

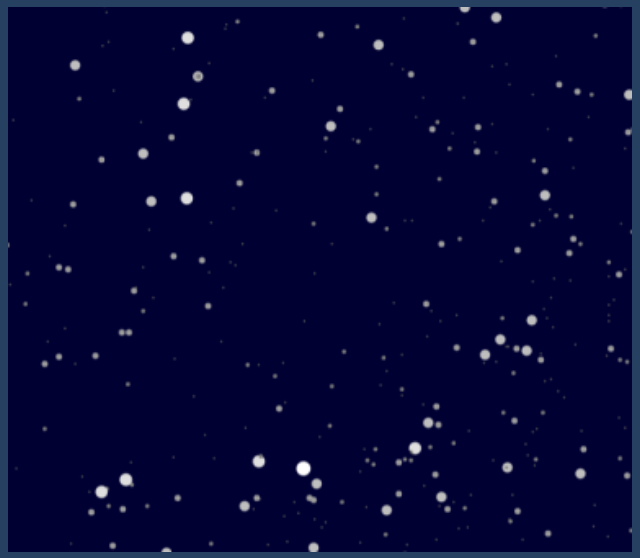
Fonctionnement de Laïka



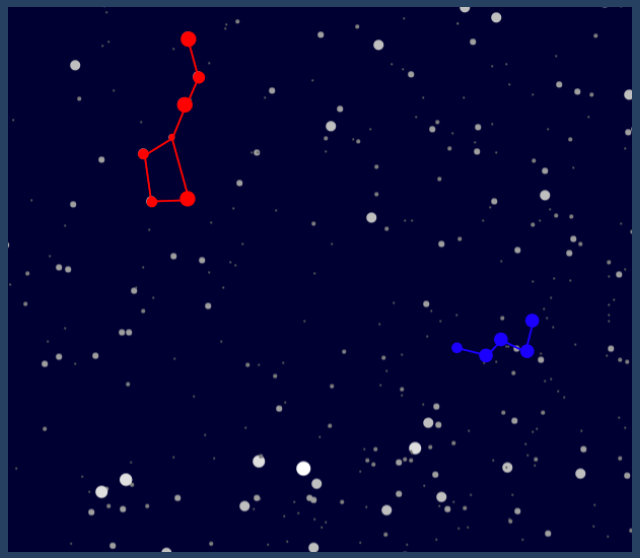
5. Tracer()

[Listes des constellations contenue dans la photographie]

Fonctionnement de Laïka

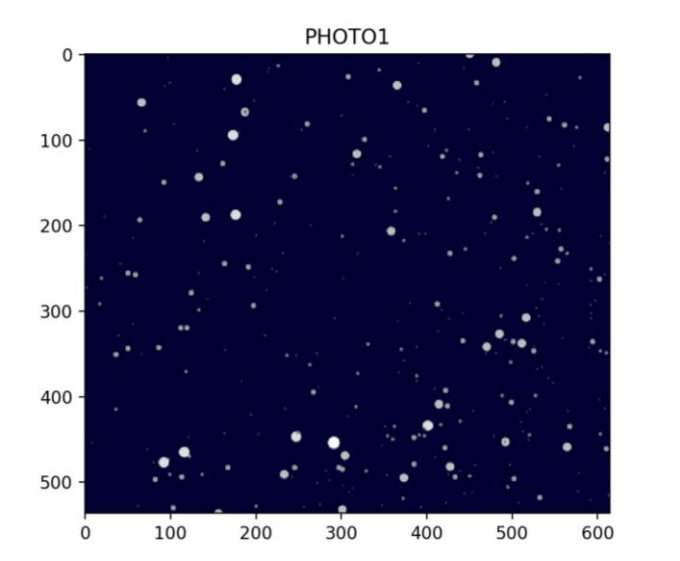


5. Tracer()

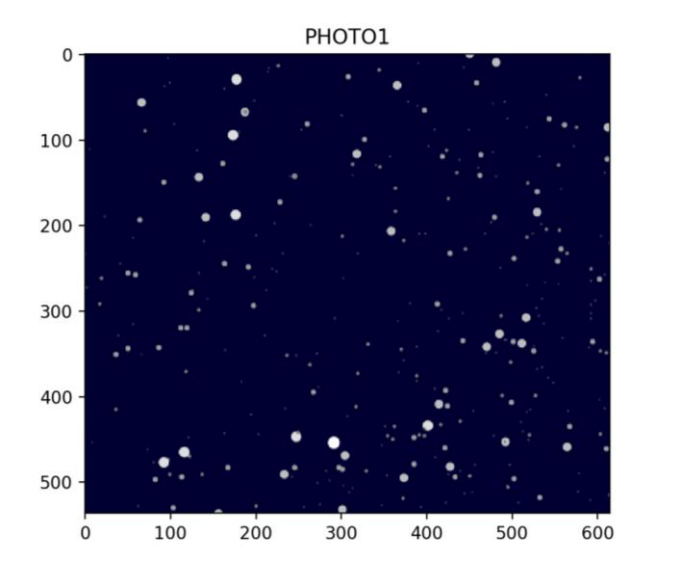


[Listes des constellations contenue dans la photographie]

Exemples : photo1

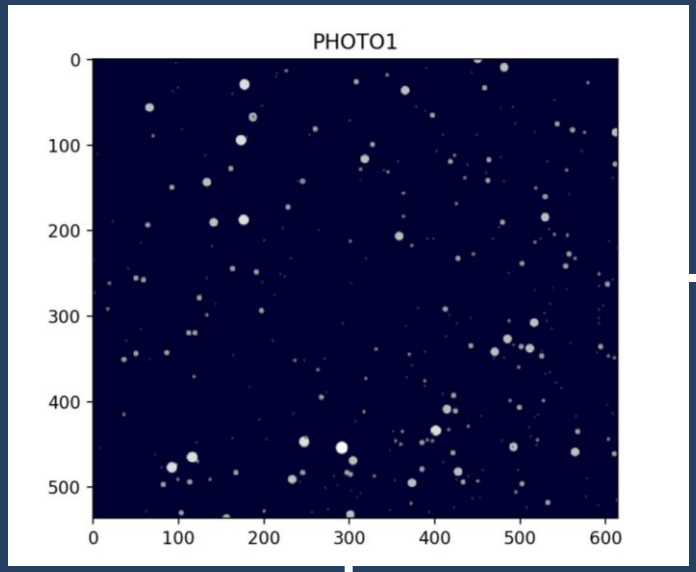


Exemples : photo1



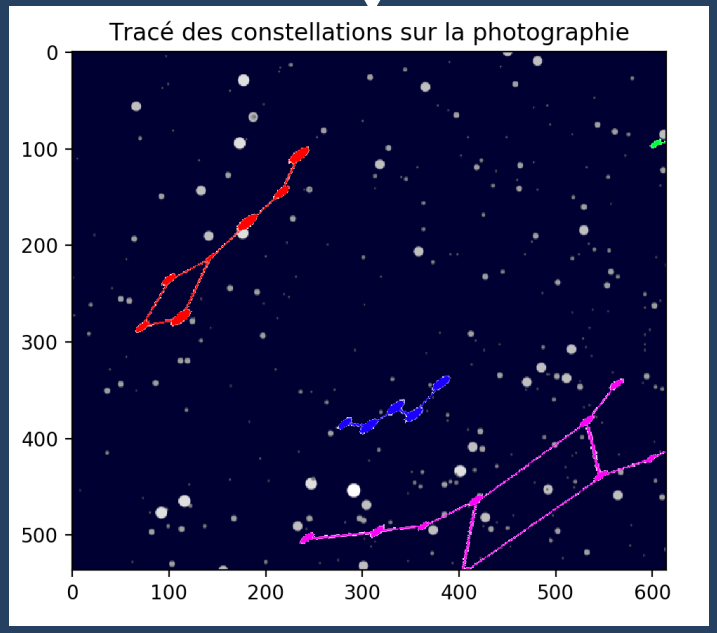

```
>>> Laika(PHOTO1,50)
['La constellation de La Grande Ourse (ROU
GE) est présente', 'La constellation de Ca
ssiopée (BLEU) est présente']
```

Exemples : photo1

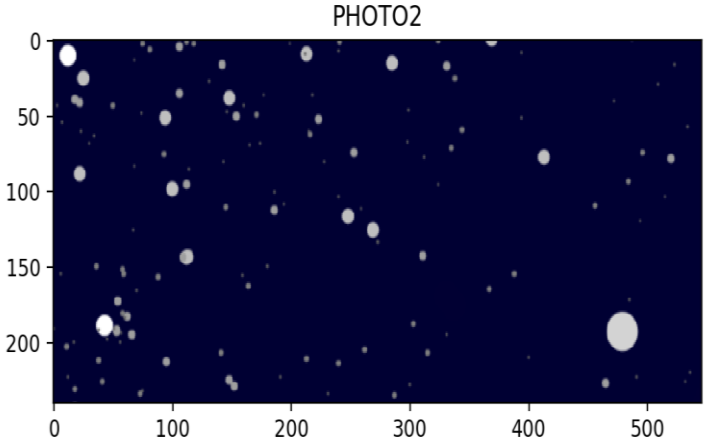


Laïka

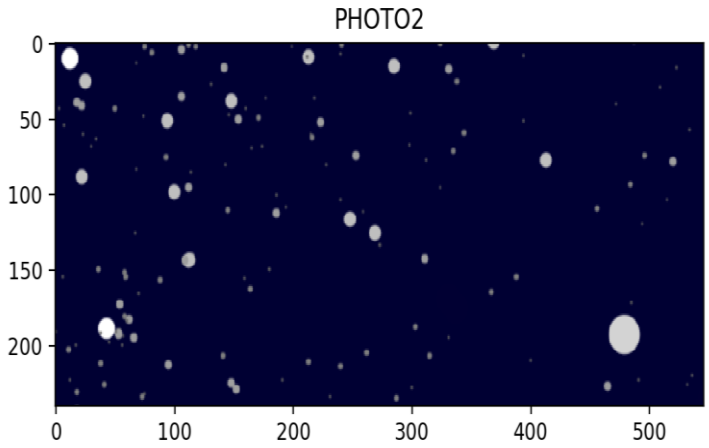
```
>>> Laika(PHOTO1,50)
['La constellation de La Grande Ourse (ROU
GE) est présente', 'La constellation de Ca
ssiopée (BLEU) est présente']
```



Exemples : photo2



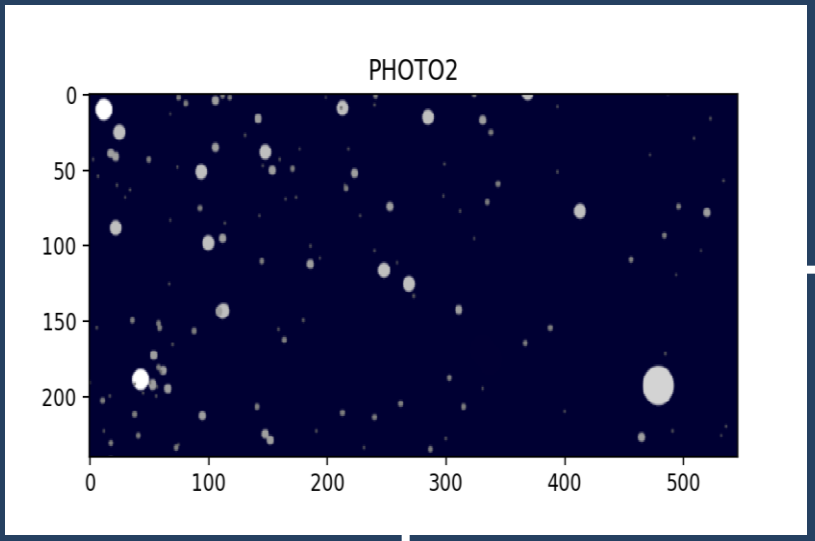
Exemples : photo2

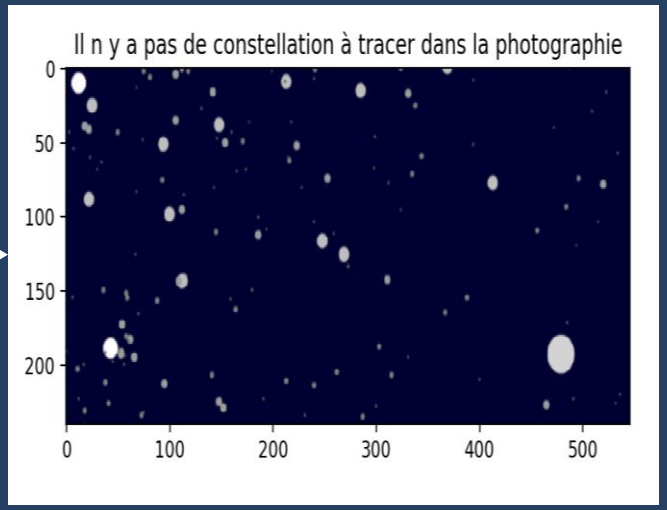


Laïka

```
>>> Laika(PHOTO2,25)
Il n'y a pas de constellation dans l'image
```

Exemples : photo2





Laïka

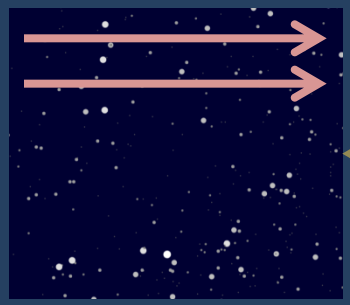
```
>>> Laika(PHOTO2,25)
Il n'y a pas de constellation dans l'image
```

Zoome sur : Liste_etoile()

Zoome sur : Liste_etoile()

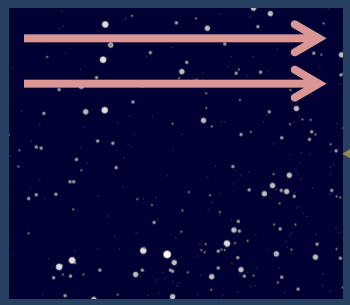
```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min:
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

Zoome sur : Liste_etoile()



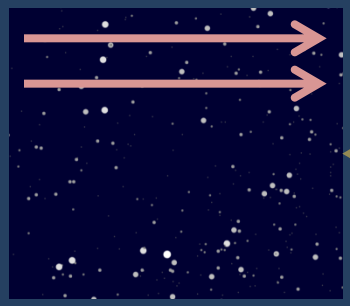
```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min:
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

Zoome sur : Liste_etoile()



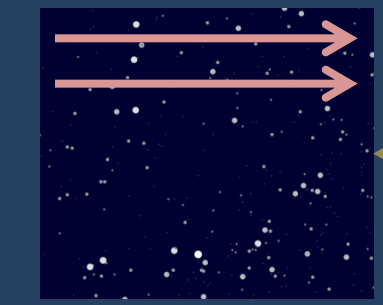
```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

Zoome sur : Liste_etoile()



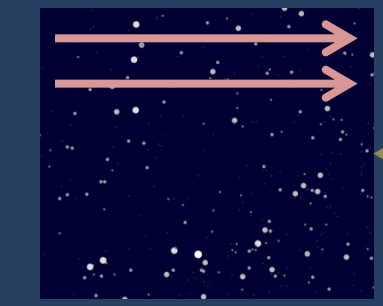
```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

Zoome sur : Liste_etoile()



```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

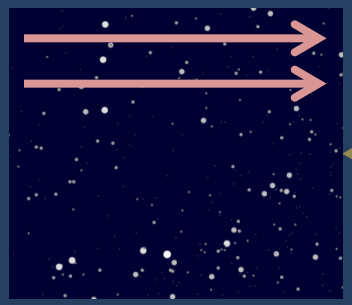
Zoome sur : Liste_etoile()



Etoile = [pixels blancs]

```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

Zoome sur : Liste_etoile()



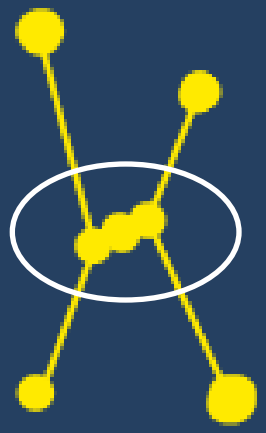
Etoile = [pixels blancs]

Liste_etoile += [Etoile]

```
121
122 ## Fonctions principales,une liste récoltant toutes les étoiles d'une photo
123
124
125 def liste_etoile(ciel_array_rouge,valeur_min):
126     """retourne la liste des étoiles présentent dans notre ciel étudier.
127     Chaque étoile est définit comme la liste des pixels qui la consitue"""
128     copie_ciel_array = np.array(ciel_array_rouge)
129     hauteur,longueur = copie_ciel_array.shape
130     liste_etoiles_ensemble_pixels = []
131     for i in range(hauteur):
132         for j in range(longueur):
133             #Si on rencontre une étoile
134             if copie_ciel_array[i,j] > valeur_min:
135                 etoile_ensemble_pixels =[]
136                 #Frontière correspond à l'ensemble des pixels qui appartiennent
137                 #à l'étoile et qui doivent être traitées
138                 frontiere = [(i,j)]
139                 while frontiere != [] :
140                     pixel = frontiere.pop(0)
141                     #Les pixels une fois traités sont mis en NOIR(=0) afin de ne
142                     #pas les re-rencontrer
143                     copie_ciel_array[pixel] = NOIR
144                     etoile_ensemble_pixels.append(pixel)
145                     for v in voisins(pixel,copie_ciel_array,valeur_min):
146                         if not v in frontiere :
147                             frontiere.append(v)
148                 liste_etoiles_ensemble_pixels.append(etoile_ensemble_pixels)
149     return(liste_etoiles_ensemble_pixels)
150
```

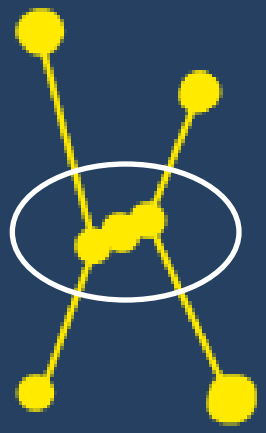
Limites du fonctionnement de liste_etoile()

Limites du fonctionnement de Liste_etoile()



3 étoiles accolées

Limites du fonctionnement de Liste_etoile()



3 étoiles accolées

Une grosse étoile

Zoome sur : Recherche_constellation()

```
304
305 def recherche_constellation(liste_etoiles_photo):
306     """ retourne, si elles existent, toutes les constellations présentes
307     dans notre ciel étoilé inconnu"""
308     LC = []
309     LR = []
310     LT = []
311     Liste_constellations_photo = []
312     LE_copie = list(liste_etoiles_photo)
313     #On fabrique des triplets, et on calcule les rapports associés à
314     #chacun d'eux
315     for E in LE_copie :
316         LE_copie.pop(0)
317         PPE = plus_proche(E,LE_copie)
318         LC  = couple(PPE)
319         for C in LC :
320             T   = list(C) + [E]
321             LR += [Rapport(T)]
322             LT += [T]
323     #On regarde si le triplet correspondent à une constellation
324     for r in range (len(LR)) :
325         for k in range(5):
326             if np.abs(LR[r] - Liste_constellations_caracterisees[k][0]) <=
10**(-8):
327                 Liste_constellations_photo += [[k,LT[r]]]
328                 break
329     return Liste_constellations_photo
330
331
```

Zoome sur : Recherche_constellation()

Formation des triplet d'étoiles proches et calcule de leur rapport

```
304
305 def recherche_constellation(liste_etoiles_photo):
306     """ retourne, si elles existent, toutes les constellations présentes
307     dans notre ciel étoilé inconnu"""
308     LC = []
309     LR = []
310     LT = []
311     Liste_constellations_photo = []
312     LE_copie = list(liste_etoiles_photo)
313     #On fabrique des triplets, et on calcule les rapports associés à
314     #chacun d'eux
315     for E in LE_copie :
316         LE_copie.pop(0)
317         PPE = plus_proche(E,LE_copie)
318         LC  = couple(PPE)
319         for C in LC :
320             T   = list(C) + [E]
321             LR += [Rapport(T)]
322             LT += [T]
323     #On regarde si le triplet correspondent à une constellation
324     for r in range (len(LR)) :
325         for k in range(5):
326             if np.abs(LR[r] - Liste_constellations_caracterisees[k][0]) <=
10**(-8):
327                 Liste_constellations_photo += [[k,LT[r]]]
328                 break
329     return Liste_constellations_photo
330
331
```

Zoome sur : Recherche_constellation()

Formation des triplet d'étoiles proches et calcule de leur rapport

Rapport triplet = Rapport constellation x

```
304
305 def recherche_constellation(liste_etoiles_photo):
306     """ retourne, si elles existent, toutes les constellations présentes
307     dans notre ciel étoilé inconnu"""
308     LC = []
309     LR = []
310     LT = []
311     Liste_constellations_photo = []
312     LE_copie = list(liste_etoiles_photo)
313     #On fabrique des triplets, et on calcule les rapports associés à
314     #chacun d'eux
315     for E in LE_copie :
316         LE_copie.pop(0)
317         PPE = plus_proche(E,LE_copie)
318         LC  = couple(PPE)
319         for C in LC :
320             T   = list(C) + [E]
321             LR += [Rapport(T)]
322             LT += [T]
323     #On regarde si le triplet correspondent à une constellation
324     for r in range (len(LR)) :
325         for k in range(5):
326             if np.abs(LR[r] - Liste_constellations_caracterisees[k][0]) <=
    10**(-8):
327                 Liste_constellations_photo += [[k,LT[r]]]
328                 break
329     return Liste_constellations_photo
330
331
```

Zoome sur : Recherche_constellation()

Formation des triplet d'étoiles proches et calcule de leur rapport

Rapport triplet = Rapport constellation x

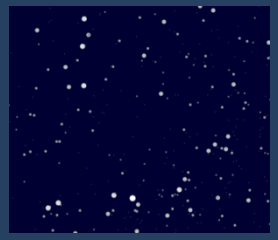
=>

triplet = constellation x

```
304
305 def recherche_constellation(liste_etoiles_photo):
306     """ retourne, si elles existent, toutes les constellations présentes
307     dans notre ciel étoilé inconnu"""
308     LC = []
309     LR = []
310     LT = []
311     Liste_constellations_photo = []
312     LE_copie = list(liste_etoiles_photo)
313     #On fabrique des triplets, et on calcule les rapports associés à
314     #chacun d'eux
315     for E in LE_copie :
316         LE_copie.pop(0)
317         PPE = plus_proche(E,LE_copie)
318         LC  = couple(PPE)
319         for C in LC :
320             T   = list(C) + [E]
321             LR += [Rapport(T)]
322             LT += [T]
323     #On regarde si le triplet correspondent à une constellation
324     for r in range (len(LR)) :
325         for k in range(5):
326             if np.abs(LR[r] - Liste_constellations_caracterisees[k][0]) <=
    10**(-8):
327                 Liste_constellations_photo += [[k,LT[r]]]
328                 break
329     return Liste_constellations_photo
330
331
```

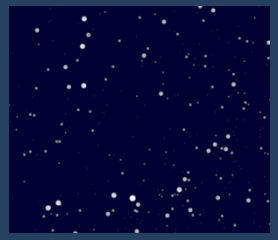
Bilan sur les limites du projets

Bilan sur les limites du projets



- Les photographies doivent toutes avoir la même résolution

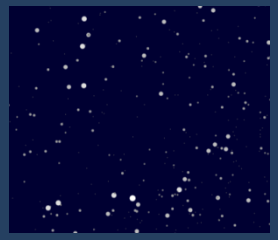
Bilan sur les limites du projets



- Les photographies doivent toutes avoir la même résolution

- Les rapports doivent être assez différents les un des autres

Bilan sur les limites du projets

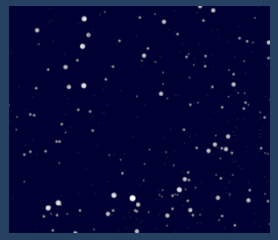


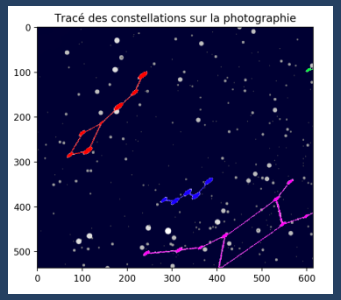
- Les photographies doivent toutes avoir la même résolution

- Les rapports doivent être assez différents les un des autres

- Trois étoiles accolées sont simplifier en une seule

Bilan sur les limites du projets





- Les photographies doivent toutes avoir la même résolution
- Les rapports doivent être assez différents les un des autres
- Trois étoiles accolées sont simplifier en une seule
- Laika a encore du mal à tracer les constellations

Conclusion

Pourquoi Laïka?

En hommage à cette chienne russe, premier être vivant envoyé dans l'espace, et malheureusement décéder une fois là haut, et cela dans d'atroces souffrances.