Fiche d'exercices P5.js

Exercice 1 Tracer un carré bleu, coupé par ses deux diagonales tracées en vert.

Exercice 2 Tracer la figure suivante : un carré colorié en orange, et à l'intérieur faites deux triangles opposés par le sommet jaunes, le tout entouré de quatre demi-cercles violets.

Exercice 3 Créer une fonction rectrayé(x,y,l,h,r,v,b) qui trace un rectangle dont le coin gauche supérieur a pour coordonnées (x,y), qui a pour longueur l et pour hauteur h. sa couleur sera définie par les paramètres rvb et l'épaisseur de son trait sera de 2. Il sera coupé par ses deux diagonales (en noir).
Tracer une centaine de ces rectangles avec un placement et une couleur aléatoire.

Exercice 4 Chercher une image de petite taille, téléchargez-la, et incluez-la dans votre sketch.
Compléter l'intérieur de la fonction draw pour représenter sur le canevas un quadrillage composé de n lignes et n colonnes de répétitions de l'image.

Exercice 5 On se basera sur le sketch de base de l'exemple 0.
À l'aide de la commande "mouseIsPressed" (*vous pouvez trouver comment l'utiliser en tapant son nom sur le moteur de recherche interne de la page dédiée aux fonctions de P5.js que vous trouverez à l'adresse P5js.org/reference/*) faire en sorte qu'à chaque fois qu'on appuie sur le bouton de la souris on utilise une nouvelle couleur choisie aléatoirement
Aide : vous pouvez utiliser la fonction définie à l'exercice 27 de la fiche dédiée aux bases, il faudra la définir en dehors de setup() et draw() et vous l'appellerez trois fois dans cette dernière)

Exercice 6 en vous inspirant de l'exemple proposé dans la partie référence du site p5js.org pour la fonction "keyIsDown" tracer un carré gris au centre du canevas et faire en sorte que les flèches à gauche et à droite le rendent plus ou moins foncé, et les flèches vers le haut et le bas le rendent plus ou moins grand.

Exercices 7 Couvrir le canevas d'un damier noir et blanc de 10 par 10
Couvrir le canevas d'un damier avec des couleurs aléatoires.
Puis couvrir le canevas d'un damier ou les cases noires sont en dégradé de gris (tester plusieurs interprétations de la consigne)

Exercice 8 Couvrir le canevas (que l'on fera carré pour cet exercice) d'un damier interactif : la case survolée sera bleue

Exercice 9 Couvrir le canevas (que l'on fera carré pour cet exercice) d'un damier interactif : plus la case « noire » est loin de la souris plus la case sera foncée
Aide : créer une fonction qui donnera la distance du point dont on donnera les coordonnées en entrée.
Utiliser la commande map pour associer à la distance une intensité de gris

Exercice 10 Couvrir le canevas (que l'on fera carré pour cet exercice) d'un damier interactif : rendre la finesse du quadrillage interactive : plus la souris est à droite plus c'est fin (taille minimale de case = 2 pixels) plus la souris est à gauche plus c'est large (taille maximale de case = taille du canevas)

Exercice 11 Créer une balle rebondissant verticalement entre les bords haut et bas de votre canevas.

Exercice 12 Créer une balle lancée dans une direction aléatoire puis rebondissant dans tous les sens (utiliser la loi de réfraction d'optique pour prévoir la suite de son mouvement après avoir rebondit sur une des parois.

Avec P5.play.js

Exercice 13 Reprendre l'exemple de sprite réactif et faite en sorte qu'avec les flèches, le carré accélère dans la direction donnée par la flèche.

Exercice 14 Reprendre l'exemple « carré rebondissant » pour faire en sorte qu'il soit lancé dans une direction aléatoire et qu'il rebondisse en plus de sol aussi sur les murs.

Exercice 15 En vous inspirant de suivre la souris, écrire un programme où le carré va fuir plus ou moins fortement le curseur de la souris

Exercice 16 En vous inspirant du programme sprites multiples créer une pluie de billes descendant verticalement, raffinez en utilisant la durée de vie, la zone d'apparition, et la vitesse de descente.

Exercice 17 Combinez le programme groupes de sprites avec ceux qui suivent pour faire en sorte que les spirites attirés se retrouvent bloqués par des murs placés aléatoirement au début de l'exécution, et engraissent en se nourrissant d'une autre catégorie de sprites verts qui une fois consommé vont réapparaitre aléatoirement et maigrisse en se nourrissant de sprites rouges qui réapparaîtront eux aussi de manière aléatoire.

Aide (exercice 4) *(voir https://www.openprocessing.org/sketch/624078):*
l'idée c'est de généraliser le programme pour faire du 3x3, du 4x4, 5x5 etc
pour faire du 2x2 mon miffy est un carré de taille : tailleFenetre/2 par tailleFenetre/2
Quelle est la taille de miffy si je veux faire du 3x3 ? etc généralisation : du nxn
sur la première ligne (notée ligne 0) de mon 2x2 les coordonnées du coin haut gauche de mes miffy sont : (0,0) et (tailleFenetre/2,0) sur la seconde ligne notée ligne 1 les coordonnées sont : (tailleFenetre/2,0) et (tailleFenetre/2,tailleFenetre/2)
Généraliser ça pour le cas où l'on a nxn miffys et que l'on s'intéresse à la i ième colonne et la j ième ligne (rappel i prend ses valeurs de 0 à n-1 et j aussi)
Du coup quelle va être la commande pour afficher ce miffy à la bonne taille et au bon endroit ?
Il ne vous reste plus qu'à organiser deux boucles (une en i et une en j) autour de cette commande pour avoir votre quadrillage.