

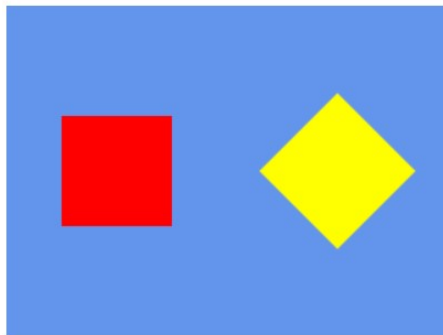
# PROGRAMMATION WEB AVANCEE

## TP4 suite

### Canvas

#### Exercice 1

Créer l'application ci-dessous (utiliser save et restore)



#### Exercice 2

Créer l'application ci-dessous en utilisant save et restore

Ajouter en dessous les mêmes carrées avec une rotation



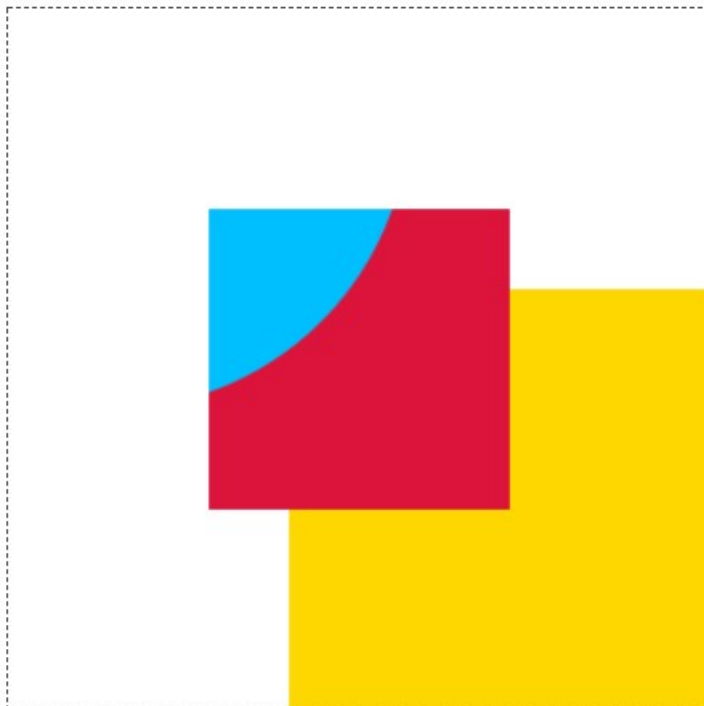
### **Exercice 3**

Créer le canvas ci-dessous :



### **Exercice 4**

Créer le canvas ci-dessous :



## **Exercice 5**

Créer un masque circulaire sur une image

## **Exercice 6 :: Editeur de dessin**

On explore le fonctionnement d'un `canvas` (html5) : comme y dessiner et comment y capturer un clic de souris.

1. Créer le `canvas` en html en spécifiant ses dimensions.
2. Écrire une fonction `initialisations()` qui récupère le *contexte 2d* du `canvas` et associer cette fonction à l'événement `onload` du document.
3. Définir une fonction `surclic(e)` qui lance une alerte affichant les coordonnées du clic contenu dans `e`. Compléter la fonction `initialisations()` pour associer un clic sur le `canvas` à l'appel de la fonction `surclic(e)`.
4. Remplacer le lancement de l'alerte par l'affichage d'un carré à l'endroit du clic. Régler la taille de ce carré.
5. Ajouter des boutons, chaque bouton permettant de choisir une couleur différente.
6. Ajouter deux boutons pour choisir la forme de la prochaine figure, soit un carré soit un disque.
7. Ajouter un bouton qui va lancer la composition aléatoire d'un dessin, par exemple en produisant une figure toutes les demi-secondes, figure dont la forme et la couleur sont choisies au hasard. Le dessin prend fin après 100 figures ou par un clic sur un dernier bouton.

## **Exercice 7**

1. Afficher un carré dans un *canvas*.
2. Permettre son déplacement avec les touches du clavier en veillant à ne pas dépasser les bords du *canvas*.
3. Ajouter un bouton pour lancer le déplacement aléatoire du carré, un autre bouton pour l'arrêter.
4. Comment faire disparaître la trace pour seulement déplacer un carré ?

## **Exercice 8**

Une cible apparaît à une position aléatoire, le joueur a quelques instants pour cliquer dessus.

1. Définir l'aspect du jeu à l'aide d'un *canvas* html5.
2. Faire apparaître un carré en guise de cible, à une position aléatoire du *canvas*. La cible doit rester apparente deux secondes, puis disparaître. Une autre cible est alors générée aléatoirement.

La production de cibles doit débiter dès le chargement de la page.

3. Capturer les clics de l'utilisateur.

Tester si le clic a touché la cible. Si c'est le cas, effacer la cible et en produire une autre.

4. Lors d'un clic, afficher un message indiquant si la cible est touchée ou non. Afficher également le nombre de cibles détruites.
5. Ajouter un bouton pour faire une pause dans la partie et un autre pour démarrer une nouvelle partie.
6. (bonus) Faire varier taille, forme et couleur des cibles.

## **Exercice 9**

Dans ce jeu, il s'agit de trouver un mot dont les lettres ont été éparpillées sur un *canvas*.

1. Définir en html un *canvas*, par exemple un carré de 400 pixels de côté. Prévoir également une zone dédiée au dialogue entre le jeu et le joueur.
2. En JavaScript, définir une variable contenant un mot, et une autre variable contenant la définition de ce mot.
3. Placer chaque lettre du mot à une position du *canvas* choisie aléatoirement. On pourra aussi choisir aléatoirement, pour chaque lettre, sa police, sa taille et sa couleur.
4. Ajouter en html un champ texte qui permet au joueur de saisir une proposition. Y associer une procédure JavaScript qui confronte cette proposition à la solution et affiche un message selon que le joueur a trouvé ou non.
5. Ajouter un bouton « indice » qui, lorsque l'on clique dessus, affiche la définition du mot. Et un autre bouton « solution » qui provoquera l'affichage du mot mystère.
6. Choisir au hasard le mot à trouver dans une liste préalablement définie (pour chaque mot de cette liste on dispose aussi de sa définition). Ajouter un bouton qui permet de commencer une nouvelle partie avec un nouveau mot.
7. Faire varier régulièrement l'affichage des lettres. Ajouter un bouton pour suspendre cette animation, un autre pour la reprendre. Imaginer enfin une animation graphique particulière pour le moment où le mot mystère est découvert.