

Indice Niveau 1 :



Objectif : Ramasser la clé puis ouvrir le coffre.

Fonctions de contrôle : avancer(), droite(), gauche(), ouvrir()

Contraintes : Le programme ne doit pas dépasser 10 lignes.

Solution attendue :

Ramasser la clé en enchainant séquentiellement les instructions de déplacements élémentaires.

Parcourir la ligne droite de 16 blocs à l'aide d'une **boucle (for)** jusqu'au coffre.

Ouvrir le coffre.

Indice Niveau 2 :



Objectif : Ramasser la clé puis ouvrir le coffre.

Fonctions de contrôle : avancer(), droite(), gauche(), sauter(), coup(), ouvrir()

Contraintes : Le programme ne doit pas dépasser 14 lignes.

[Solution attendue :](#)

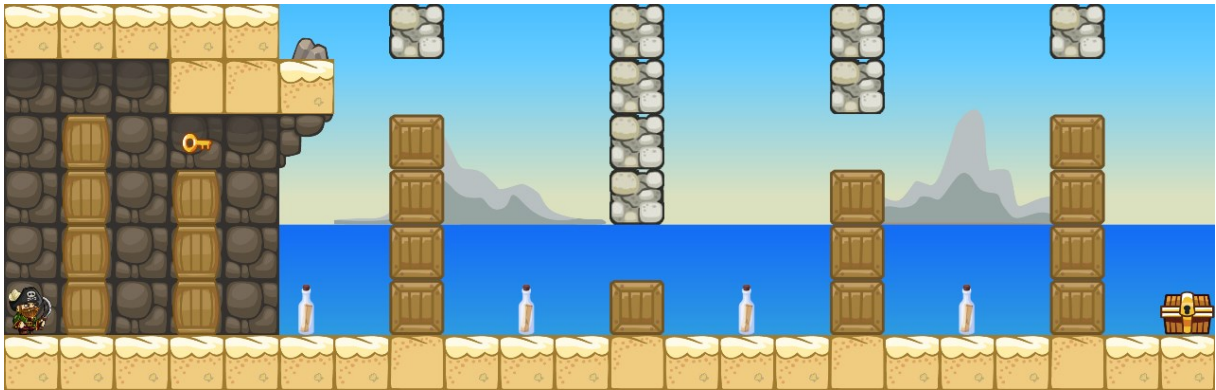
Sauter par-dessus chacun des 6 tonneaux à l'aide d'une boucle bornée (for).

Descendre sur la plateforme inférieure.

Détruire les 9 pots à l'aide d'une **boucle (for)**.

Approcher puis ouvrir le coffre.

Indice Niveau 3 :



Objectif : Ramasser la clé puis ouvrir le coffre. La hauteur des piles de caisses est aléatoire et change à chaque exécution. Il faut donc lire le message dans la bouteille avant chaque pile qui indique sa hauteur.

Fonctions de contrôle : avancer(), sauterHauteur(hauteur), lireNombre(), ouvrir()

Contraintes : Le programme ne doit pas dépasser 14 lignes.

Solution attendue :

Sauter les deux piles de tonneaux avec l'aide de la fonction sauterHauteur.

Utiliser une **boucle (for)** pour répéter quatre fois les instructions :

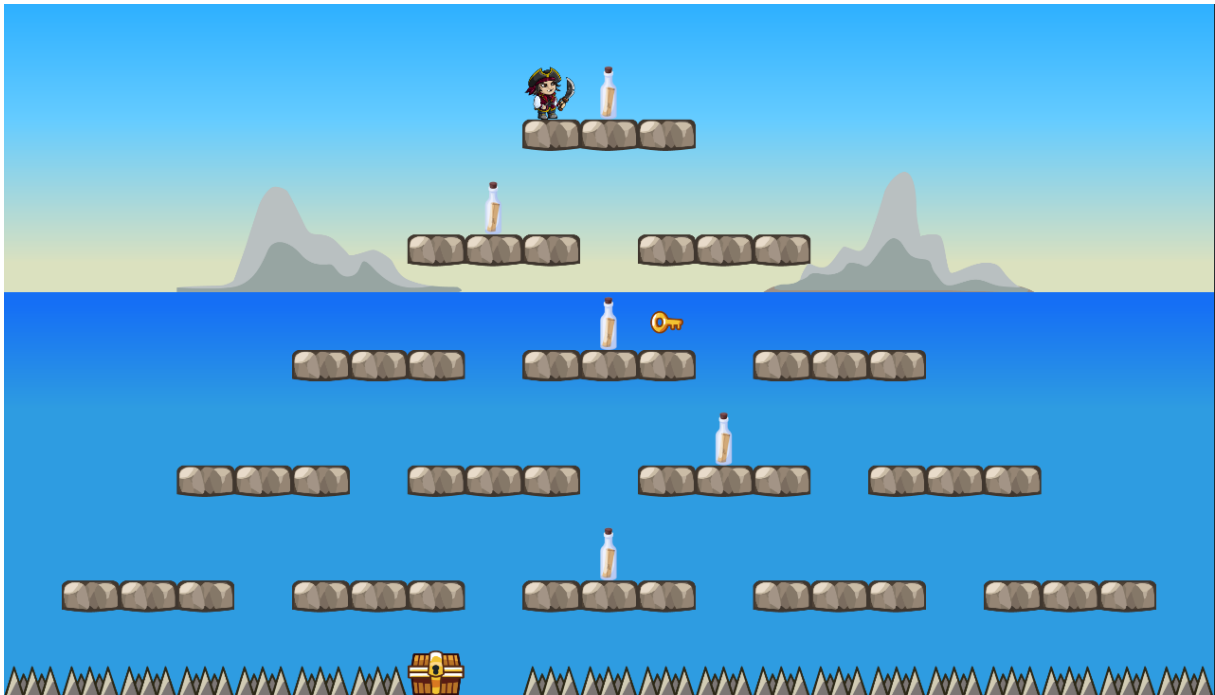
Lire dans la bouteille la hauteur des caisses à l'aide de la fonction lireNombre et affecter une variable du retour de cette fonction.

Se rapprocher de la pile puis sauter à la hauteur indiquée dans la variable.

Se rapprocher de la bouteille suivante.

Ouvrir le coffre.

Indice Niveau 4 :



Objectif : Ramasser la clé puis ouvrir le coffre. L'emplacement du coffre et de la clé sont aléatoires et changent à chaque exécution. Il faut donc lire les messages dans les bouteilles qui indiquent le chemin à suivre (gauche ou droite) à chaque étage. Le fait de tomber dans les piques fait perdre puis recommencer le niveau.

Fonctions de contrôle : avancer(), gauche(), droite(), lireChaine(), ouvrir()

Contraintes : Le programme ne doit pas dépasser 14 lignes.

Solution attendue :

Avancer d'un bloc afin de se positionner sur la bouteille.

Utiliser une **boucle (for)** pour répéter cinq fois les instructions :

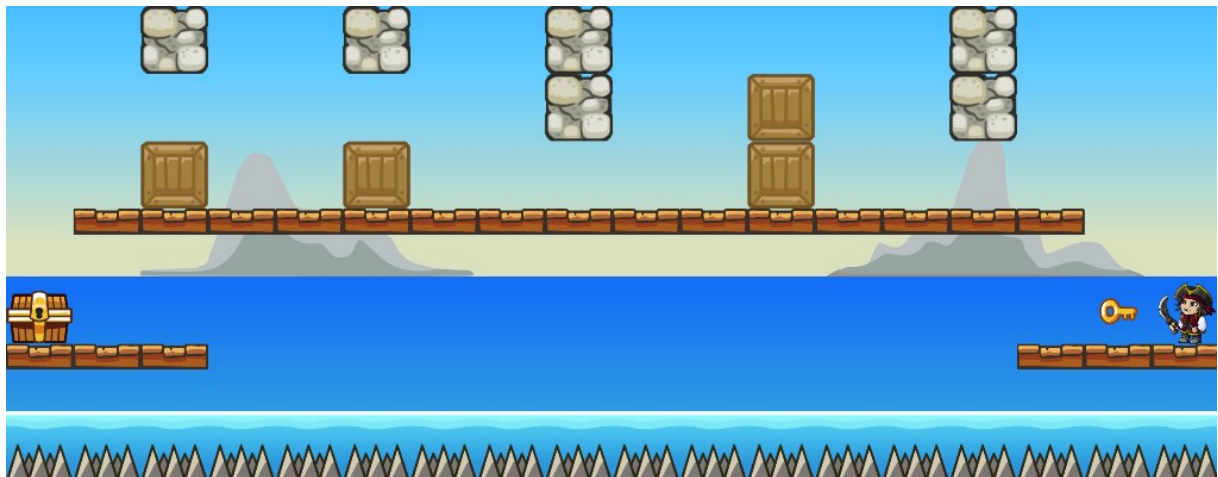
Appeler la fonction lireChaine et affecter une variable du retour de cette fonction.

Réaliser un test sur cette variable à l'aide d'une **structure conditionnelle if-else** permettant d'aller à gauche ou à droite.

Avancer de deux blocs pour descendre d'un étage.

Ouvrir le coffre.

Indice Niveau 5 :



Objectif : Ramasser la clé puis ouvrir le coffre. Sur la plateforme supérieure, la hauteur des caisses est aléatoire et change à chaque exécution. Le fait de tomber dans les piques fait perdre puis recommencer le niveau.

Fonctions de contrôle : avancer(), sauter(), sauterHaut(), mesurerHauteur(), ouvrir()

Contraintes : Le programme ne doit pas dépasser 18 lignes.

Solution attendue :

Ramasser la clé sur la plateforme inférieure, puis monter à l'aide d'un saut haut.

Utiliser une **boucle (for)** pour répéter cinq fois les instructions :

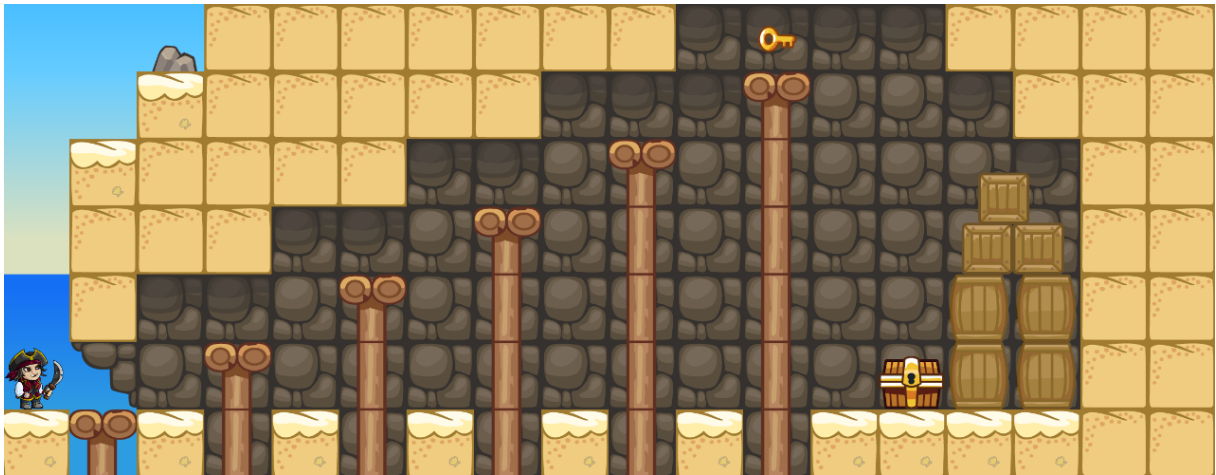
Mesurer la hauteur des caisses se situant devant soi à l'aide de la fonction `mesurerHauteur` puis affecter une variable à l'aide de cette valeur.

Utiliser une **structure conditionnelle if-elif-else** afin d'effectuer l'action adéquate en fonction de la hauteur des caisses (avancer, sauter ou sauterHaut).

Avancer deux fois jusqu'aux prochaines caisses.

Avancer pour descendre sur la plateforme inférieure puis ouvrir le coffre.

Indice Niveau 6 :



Objectif : Ramasser la clé puis ouvrir le coffre.

Fonctions de contrôle : avancer(), sauterHauteur(hauteur), ouvrir()

Contraintes : Le programme ne doit pas dépasser 4 lignes.Solution attendue :

Solution attendue :

Gravir les piliers à l'aide d'une **boucle (for)** en utilisant la variable de boucle afin de sauter à la bonne hauteur.

Ouvrir le coffre.

Indice Niveau 7 :



Objectif : Tirer sur toutes les noix de coco puis sur le coffre sans jamais atteindre l'autre pirate. Les noix de coco apparaissent par séries alternativement à droite puis à gauche, leur nombre augmente à chaque fois. L'autre pirate se place toujours derrière la dernière. Le fait de tirer sur ce pirate fait perdre puis recommencer le niveau.

Fonctions de contrôle : tourner(), tirer(distance)

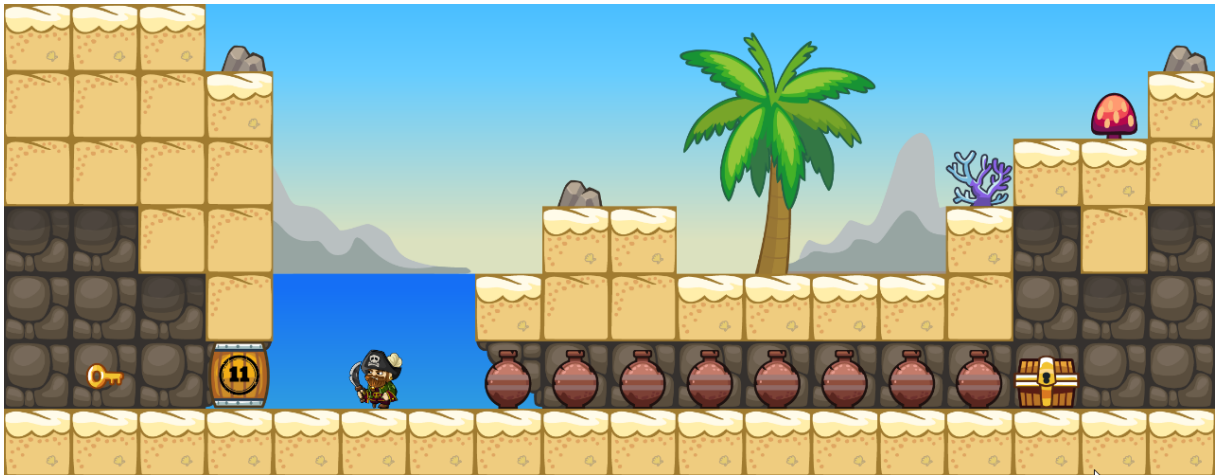
Contraintes : Le programme ne doit pas dépasser 5 lignes.

Solution attendue :

Tirer alternativement à droite puis à gauche et de plus en plus loin sur les différentes séries de noix de coco à l'aide d'une **boucle (for)** et de sa variable de boucle initialisée à 2 et prenant 8 pour dernière valeur.

Tirer sur le coffre.

Indice Niveau 8 :



Objectif : Ramasser la clé puis ouvrir le coffre. La solidité du tonneau (nombre dans le cercle), le nombre de pots ainsi que l'emplacement du coffre sont aléatoires et changent à chaque exécution. Un fois le tonneau détruit, de la dynamite apparait à sa place. Il ne faut pas donner de coup dans cette dynamite sous peine d'une explosion qui fait perdre puis recommencer le niveau.

Fonctions de contrôle : avancer(), gauche(), droite(), coup(), detecterObstacle(), ouvrir()

Contraintes : Aucunes

Solution attendue :

Avancer jusqu'au tonneau.

Utiliser une **boucle (while)** afin de donner des coups de sabre tant qu'un obstacle (le tonneau) se trouve devant soi.

Avancer de trois blocs jusqu'à la clé.

Se retourner puis avancer de 5 blocs jusqu'au premier pot.

Utiliser une **boucle (while)** afin de porter un coup de sabre et d'avancer tant qu'un obstacle (un pot) se trouve devant nous.

Ouvrir le coffre.