



DIRECTIVES ET INFORMATIONS RELATIVES AU TRAVAIL

- Ce travail compte pour 20% de la note finale.
- Ce travail se fait en équipe.
- La remise du travail pratique complet (format zip) se fait sur Léa avant la date butoir (23h59 le 17 octobre).
- Il est interdit de partager des artefacts avec d'autres équipes. Attention au plagiat.

LE JEU

Une bombe. Cinq câbles. Chaque câble associe un pays à une capitale. Si l'association est fausse, il faut couper. Mais attention à ne pas couper le mauvais câble...

Au début du jeu, le programme demande au joueur de se nommer. Après un commentaire désobligeant, il lui présente une bombe à désamorcer. Le nombre de câbles à couper varie d'une bombe à l'autre (minimum 1, maximum 4).

Si le joueur sectionne un câble qu'il ne faut pas couper (bonne association pays-capitale) :

Le programme affiche **"*** BOOM! ***"** pour indiquer que la bombe a explosé.

Le programme demande ensuite au joueur (ou plutôt ce qu'il en reste) s'il veut jouer à nouveau.

Le joueur répond par oui (**o**) ou par non (**n**) :

Dans l'affirmative, on relance une partie en lui présentant une nouvelle bombe.

Dans le cas contraire, le programme cesse son exécution.

Si le joueur sectionne un bon câble qu'il faut couper (mauvaise association pays-capitale) :

Le programme réaffiche tous les câbles (ceux coupés et ceux toujours reliés).

Le programme demande ensuite au joueur quel autre câble il veut couper.

Si, par une chance extraordinaire (ou en trichant), le joueur parvient à désamorcer la bombe, donc à couper tous les câbles qu'il doit couper (toutes les mauvaises associations pays-capitale) :

Le programme le félicite en affichant **"OMG!!! Bombe désamorcée. Bravo {nom du joueur}!"**

Le programme demande ensuite au joueur survivant s'il veut jouer à nouveau.

Le joueur répond par oui (**o**) ou par non (**n**) :

Dans l'affirmative, on relance une partie en lui présentant une nouvelle bombe.
Dans le cas contraire, le programme cesse son exécution.

MANDAT ET LIVRABLES

Une équipe de codeurs : vous. Une application, du code atroce, plein de bogues. Vos claviers et votre courage en main, saurez-vous décortiquer ce code, le corriger et l'améliorer?

Étape 1 Tests boîte noire

L'équipe élabore une liste de 5 tests manuels qui permettront de valider le bon fonctionnement du jeu.

Chaque test doit avoir un titre, une description et une condition de succès.

L'équipe exécute les tests boîte noire et consigne les résultats.

Exigence : au moins deux tests doivent échouer (il faut trouver au moins deux bogues en exécutant ces tests boîte noire).

Livable : plan de test et résultats des tests

10 pts

2 pts par test (pénalité de 3 pts par échec manquant)

Étape 2 Revue de code

Chaque membre de l'équipe effectue sa préparation et produit une liste d'observations.

Exigence : un minimum de 10 observations par préparation individuelle

L'équipe tient une revue de code et note les observations dans une fiche de revue de code.

Exigence : un minimum de 20 observations notées à la fiche de revue de code

Livable : toutes les préparations individuelles

Pénalité de 2 pts par préparation individuelle manquante ou ne rencontrant pas le minimum de 10 observations valides

Livable : fiche de revue de code dûment complétée

15 pts

Pénalité de 5 pts si la fiche compte moins de 20 observations valides

Étape 3 Correction

L'équipe corrige les bogues découverts

Exigence : correctifs seulement, mais pas de réusinage (voir étape suivante)

Livable : une copie du code corrigé

10 pts

Étape 4a Réusinage

L'équipe procède au réusinage du code (structure, découpage, algorithmes, validations, gestion des exceptions, etc.).

Exigence : respect des bonnes pratiques vues en classe jusqu'à maintenant (et dans les cours précédents) et du guide de style PEP 8

L'équipe réfléchit aux tests unitaires (voir plus bas) et les développe en parallèle.

Livable : une copie du code réusiné

20 pts

Étape 4b Tests unitaires

L'équipe conçoit et réalise des tests unitaires avec le module `unittest` pour vérifier le code réusiné.

Exigence : un minimum de 10 tests offrant la plus grande couverture de code possible et vérifiant les cas d'exception

Exigence : tous les tests doivent passer

Livable : une copie de la classe de test (fichier complet)

15 pts

Pénalité de 2 pts par test manquant

Étape 5 Ajout de fonctionnalités

Permettre au joueur d'abandonner en cours de partie en appuyant sur la touche Q.
Ramener poliment le joueur à l'ordre (ex. : Aie! On se concentre svp...) s'il tente de couper un câble déjà coupé.
Calculer et afficher un score (nombre de bombes désamorçées sans explosion).
Rendre l'explosion plus dramatique en ajoutant un effet sonore.
Proposer votre propre fonctionnalité et développez-la.

Livrable : une copie du code amélioré 30 pts

6 pts par amélioration bien implémentée (3 pts si fonctionnelle, 3 pts pour la qualité du code)