

# STRATEGIE DE TESTS T3



## Table des matières

Table des matières.....	2
Introduction.....	3
I. Exigences de tests.....	3
II. Typologie de tests.....	3
III. Moyen de test.....	3
Fiche de test.....	5
Conclusion.....	6

# Introduction

## I. Exigences de tests

Notre jeu devra passer différentes phase de test. Ce sera un jeu en local, joué par un seul joueur à la fois ce qui nous évites plusieurs test comme la prise en compte des accès concurrents, l'intégration avec d'autre logiciels, des tests de surcharge...

Nous devons faire passer les tests :

- Fonctionnels : bon fonctionnement de notre application (relatif à notre code)
- Sécurités : aucun test car le jeu est en local jouer par une seule personne à la fois
- IHM : Facilité d'utilisation, cohérence/cohésion (charte graphique), comportement sur erreurs, chargements initiaux.
- Performances/Charges : Temps de réaction (d'affichage, de réponse)

## II. Typologie de tests

		Importance pour l'utilisateur		
		Stratégique	Important	Secondaire
Fréquence d'utilisation	Forte			
	Moyen			
	Faible			

## III. Moyen de test

On a décidé d'utiliser un numérotage de fiche de test sous la forme « Famille de test\_numéro de test »

Exemples :

- IHM\_001
- PERF\_001
- SECU\_001
- FONC\_001

Aux vues du projet qui nous a été assignés, nos moyens de tests seront les suivant :

- 1) Tout d'abord, essayer toutes les actions réalisables dans notre jeu. Afin de tester toutes les failles possibles du code. Ainsi, nous aurons testé la plus

grande partie des actions et leurs erreurs qu'elles soient des failles de sécurité, de performances ou autres.

- 2) Dans un second temps, nous pourrions faire essayer le jeu par un tiers, qui n'intervient pas autrement dans le projet. De ce fait, il pourrait, étant externe au projet, nous éclairer sur des points que nous n'aurions pas vus, étant donné notre implication dans le projet. Il pourra ainsi nous faire un retour sur la lisibilité du jeu (interface graphique), la compréhension du jeu ...
- 3) Puis, nous pourrions tester le temps de réaction du jeu. Grâce au timer disponible lors de l'exécution. Voir ainsi si le temps est plus ou moins long selon les modifications apportées au code.
- 4) Enfin, nous décrirons chacune des méthodes présentes dans le code et leur résultat escompté. Comparer cela aux résultats obtenus et ainsi voir s'il y a des incidents, qu'ils soient mineurs, majeurs ou bloquants.

Voici ci-dessous une estimation des charges prévisionnelles pour conduire notre campagne de tests :

Type de tests	Temps estimé du test	Temps réel
Test des actions réalisables	6h	
Test du jeu par un tier	4h	
Test du temps de réaction	3h	
Test de chacune des méthodes présente dans le code	5h	

Modèle de fiche de test ci-dessous

## Fiche de test

Personnes			
Concepteur du test :		Date de conception :	
Réalisateur du test :		Date d'exécution :	
Test			
N° du test :			
Type de test :	<input type="checkbox"/> Performances/Charges <input type="checkbox"/> Sécurité <input type="checkbox"/> Fonctionnel <input type="checkbox"/> IHM		
Durée prévisionnelle :		Prérequis pour le test :	
Description du test :			
Résultats			
Résultat attendu :			
Durée réelle :			
Résultat obtenu :	<input type="checkbox"/> OK <input type="checkbox"/> KO mineur <input type="checkbox"/> KO majeur <input type="checkbox"/> KO bloquant		
Problème(s) survenu pendant le test :			
Pièces jointes :			
En cas d'incident, prise en charge par :		Solution(s) apportée(s) :	

## Conclusion

Temps passé par chaque personne sur le projet (estimation car le projet n'est pas fini)

	Design	Code	Documentation	GitLab	Base de donnée	Total
Pierre NACE		5h	1h	1h		7h
Thomas HUSSER	6h	2h	1h	1h		10h
Adélie TRICARD		4h	1h	1h	1h30	7h30
Fanny EBER		4h	2h	2h		8h