

Catégorie	Ordre de priorité	Nom de la tâche	Difficulté	Compétences	Aloïs	Samuel	Maxence	Mathis	Description
Communication Client/Server	Important	Découverte webSocket Java + JavaScript (version simple)	1	3,4	25%	25%	25%	25%	Recherche et test des technologies disponibles
	Important	Création des classes pour la gestion du réseau	3	1,2,3, 4	100%	0%	0%	0%	librairie permettant de connecter des namespaces entre le client et le serveur (pour éviter de mettre des if)
	Important	Gestion des connexions multiples	0	1,2, 3, 4	100%	0%	0%	0%	Mise en place de threads pour les connexions simultanées
	Important	Envoi des données en JSON	2	2, 3, 4	100%	0%	0%	0%	Envoi de la liste des tuiles+ et des vents sous forme de classe
	Important	Envoi des images	3	1,2, 3, 4	90%	10%	0%	0%	Redimensionnement des images avant envoi, conversion en base 64, et envoi par paquets pour les images de grande taille
Création du client (front)	Moins important	Page d'accueil	1	1	0%	10%	0%	90%	
	Important	Page de saisie des tuiles	1	1, 4	0%	0%	50%	50%	
	Important	Page de saisie des vents et fleurs	1	1, 4	0%	0%	10%	90%	
	Important	Pages des résultats	1	1, 4	0%	40%	55%	5%	
	Pas important	Page À propos	1	1	0%	0%	0%	100%	
	Pas important	Page Photo	1	1	0%	100%	0%	0%	
	Moins important	Css	2	1	0%	20%	40%	40%	
Création du client (back)	Important	Création des classes pour l'ajout des tuiles	4	1, 2, 4	0%	0%	50%	50%	Création des classes : Tile, Slot, Hand
	Important	Génération des tuiles	2	1	0%	0%	50%	50%	Au chargement de la page, les tuiles disponibles sont instanciées de manière à les rendre utilisable par le joueur.
	Important	Tiroir d'ajout des tuiles	3	1, 2	0%	0%	0%	100%	Gestion des interaction (drag) pour ouvrir et fermer le tiroir d'ajout des tuiles
	Important	Ajout des tuiles : par clic ou par drag & drop	3	1, 2, 4	0%	0%	60%	40%	Utilisation des eventListener pour appeler les fonction lors de l'appui sur les tuiles
	Important	Ajout d'un bouton pour cacher les tuiles	1	1, 4	0%	0%	100%	0%	Utilisation des eventListener pou modifier les attributs du slot
	Important	Gestion du choix des bonnus (vents et fleurs)	2	1, 4	0%	0%	0%	100%	Création d'une classe pour enregistrer les bonus du joueur et les exporter pour l'envoi
	Pas important	Création des classes pour le menu de navigation	2	1, 2	0%	0%	0%	100%	Génération du menu de navigation de manière dynamique pour le rendre commun à toutes le pages
	Pas important	Créations des classes pour le footer	2	1, 2	0%	0%	0%	100%	Génération du footer de manière dynamique et synchronisée avec le header
	Important	Stockage des données dans le localSotrage	1	4	33%	1%	33%	33%	Enregistrement des données de l'utilisateur (tuiles, vents, historique des score) pour les conserver lors de sa prochaine visite sur le site
	Important	Mise en place de l'architecture web	1	4	100%	0%	0%	0%	Optimisation et refactor du js
	Pas important	Création des classes pour les animations	2	1, 2	0%	0%	0%	100%	Classes scrollAnimaiton et AnimationFolder
	Important	Modification de l'image	2	4	0%	0%	50%	50%	Mise en nuances de gris, puis flou pour réduire le bruit, puis déterminer le treshold de manière automatique pour obtenir une image binaire
	Important	Détection des contours	4	1, 2	0%	0%	90%	10%	utilisation de la fonction findContours() après avoir déterminé les paramètres adaptés
Server : reconnaissance des tuiles (OpenCV)	Important	Détection du nombre de tuiles (avec leurs positions)	2	1,2,4	0%	0%	50%	50%	Création d'instances le la classe ImageTile pour chaque tuiles trouvée, et enregistrement des coordonnées
	Important	Détourage et correction de la perspective	4	1,2,4	0%	0%	80%	20%	Simplification des contours en quadrilatères, et détournage de l'image originale en fonction de la forme obtenue, puis correction de perspective pour obtenir un rectangle
	Important	Retrouver le nom de la tuile en se basant sur un dataSet	5	1,2,4	40%	0%	35%	25%	Parcours du dataset pour trouver la tuiles qui a le plus de points en commun avec celle extraite de l'image (en utilisant le feature matching)
	Important	Restituer les groupements de tuiles (clustering)	3	1,2,4	25%	0%	75%	0%	Utilisation de l'algorithme K-means pour détecter les tuiles qui sont physiquement proches afin de les répartir en 5 clusters
	Moins important	Création des dataSets	1	4	0%	0%	0%	100%	Création de plusieurs dataSets pour déterminer ceux qui obtiennes des meilleurs résultats (un dataSet contient les images de chaques tuile avec leur nom)
	Important	Création des classes pour la reconnaissance des tuiles	4	1, 2, 3	34%	0%	33%	33%	Répartition des fonctionnalités dans plusieurs classes : DataSet, ImageService, ImageTile, PointList, TileDetector, TilesView
	Moins important	Optimistaion du temps de détection	4	2	100%	0%	0%	0%	Refactor du code pour réduire le temps de détection (environ 30s de moins)
	Pas important	Afficher les résultats et affiner la précision de l'agroithme en conséquence	2	1,2,4	25%	25%	25%	25%	Modification des paramètres des fonctions pour obtenir les meilleurs résultats
	Important	Trouver le desing pattern	5	2	100%	0%	0%	0%	Utilisation de Factory, Compostie, Singleton...
	Important	Implémenter le desing pattern	4	1, 2	100%	0%	0%	0%	Création des classes pour implémenter les différents types de tuiles, ainsi que le classes pour la gestion des règlements
Server : calcul du score	Important	Écrire les règles du jeu	1	1, 2	25%	25%	25%	25%	Création d'une nouvelle classe pour chaque règle du jeu du Mahjong
	Moins important	page web pc	2	1, 2	0%	100%	0%	0%	Création du site web pour la version pc du site web
	Moins important	figma web pc	1	1	0%	100%	0%	0%	Création d'un figma pour le site pc
	Pas important	animation dans site	1	1, 2	0%	100%	0%	0%	petites animation dans le site web pc
	Pas important	Page d'accueil	1	1,2	0%	100%	0%	0%	première version de la page d'accueil
	Pas important	Test placement des tuile sur site	1	1,4	0%	100%	0%	0%	Premier test en javascript où l'on pouvait rajouter des tuile grâce au clic
	Pas important	prendre en photo depuis pc	2	1,2,4	0%	100%	0%	0%	Pouvoir prendre une photo via la camera du pc
	Important	test envoi d'image entre site et java	2	1,2,3	50%	50%	0%	0%	Premier test d'envoi d'image entre le site web et java via des websockets
Dossiers analyse/conception	À chaque séance	Rédaction des dossiers d'analyse et de conception	1	1,5,6	25%	25%	25%	25%	Rédaction des dossiers d'analyse et de conception
Trello	À chaque séance	Mise à jour des tâches sur le Trello	1	1,5,6	25%	25%	25%	25%	Mise à jour des tâches sur le Trello
Git	À chaque séance	Commits réguliers sur le Gitlab	1	6	25%	25%	25%	25%	Commits réguliers sur le Gitlab
Documentation du code	Moins important	Documentation et commentaires dans le code	1	1,2	25%	25%	25%	25%	Documentation et commentaires dans le code

Résultats non représentatifs (résultats des calculs)	Moyenne Compétence 1	19%	19%	27%	35%
	Moyenne Compétence 2	36%	13%	24%	27%
	Moyenne Compétence 3	72%	5%	12%	12%
	Moyenne Compétence 4	28%	9%	33%	29%
	Moyenne Compétence 5	25%	25%	25%	25%
	Moyenne Compétence 6	25%	25%	25%	25%
Total		39%	12%	24%	26%

100%

Résultats à prendre en compte	Moyenne Compétence 1	30%	10%	30%	30%
	Moyenne Compétence 2	28%	16%	28%	28%
	Moyenne Compétence 3	40%	10%	25%	25%
	Moyenne Compétence 4	20%	10%	35%	35%
	Moyenne Compétence 5	25%	25%	25%	25%
	Moyenne Compétence 6	25%	25%	25%	25%
Total		29,5%	11,5%	29,5%	29,5%

100%