

Etudiants :

Critère	Pts	Inacceptable	Passable	Bon	Excellent
Travail réalisé	6	Le projet n'a quasiment pas avancé, utilisation majeure de ressources externes non signalées ou avec peu d'ajouts personnels (0-0.5)	Quantité de travail faible par rapport au nombre d'heures, en prenant en compte l'éventuelle utilisation de ressources externes (1.5-2)	Quantité de travail adaptée au nombre d'heures, peu de personnalisation ou fonctionnalités avancées (2.5-4)	Quantité de travail adaptée au nombre d'heures, présence de fonctionnalités avancées (algo, graphisme...), montre l'investissement des étudiants dans leur projet (4.5-6)
Développement et génie logiciel	4	Bonnes pratiques de code non suivies, tout est dans un ou quelques fichiers (0-0.5)	Effort d'organisation du code en fichiers et fonctions (1-2)	La conception a été réfléchie, avec utilisation d'une organisation adaptée au projet (objet ou autre) (3)	Utilisation de méthode de design logiciel avancé : design pattern, décorateurs, classes abstraites, interfaces... (4)
Suivi des guidelines de code Python (nommage, documentation, sauts de lignes...)	2	Globalement non suivi (0)	Guidelines assez peu suivies (1)	Guidelines globalement bien suivies (1.5)	Guidelines parfaitement suivies, au niveau de codes open source (2)
Compréhension	2	Quasiment aucune réponse aux questions sur le code, l'étudiant ne comprend pas ce qui a été fait (0)	Réponses partielles aux questions (0,5)	Réponses correctes aux questions posées, mais il faut tirer les informations de l'étudiant, ou l'étudiant a des doutes qui montrent une maîtrise imparfaite du code (1)	Réponses à toutes les questions, l'étudiant donne de lui-même des éléments montrant sa maîtrise du projet et du code qu'il a produit (2)
Gestion de projet	3	Outil non mis en place ou pas à jour (0-0.5)	Outil mis en place mais éléments manquants (découpage peu précis, pas de répartition, peu de mises à jours, pas de réévaluation). (1)	Outil mis en place et utilisé dans le cadre de ce qui est imposé : découpage précis, répartition, suivi régulier, ré-évaluations (2)	Utilisation personnelle de l'outil de gestion de projet, avec utilisation de fonctionnalités avancées (3)
Versionning	3	Git non mis en place ou avec des problèmes majeurs : très peu de commits, noms de commits non informatifs (date ou nom de fichier) (0-0.5)	Des problèmes dans le git : fréquence, noms de commits, potentiellement résolu sur la fin (1-1.5)	Commits de petites tailles avec descriptions claires, utilisation par tous les membres. (2)	Utilisation de fonctionnalités avancées, bonne documentation du git qui permettrait une mise en ligne pour vitrine CV (2.5-3)