

Informatique

Présentation du sujet

Le sujet aborde le thème de la génération d'images bidimensionnelles à partir d'une scène en trois dimensions contenant des sphères éclairés par des sources lumineuses. Une technique de lancer de rayons est proposée en relation avec l'utilisation de lois physiques de propagation, de réflexion et de diffusion de la lumière.

Avec trente questions réparties sur cinq parties, le sujet fait très largement appel aux connaissances algorithmiques et pratiques du programme de première année. Beaucoup de questions relèvent de la programmation en langage Python. Quelques questions traitent de la complexité des solutions proposées. Quatre questions abordent spécifiquement la partie du programme consacrée aux bases de données.

Le problème envisage successivement :

- les outils géométriques nécessaire à la représentation d'une scène et des rayons lumineux ;
- les lois physiques de l'optique régissant les rayons lumineux ;
- une structure de base de données adaptée à la gestion des scènes ;
- l'algorithme de lancer de rayons ;
- quelques améliorations possibles de l'algorithme.

Analyse globale des résultats

Au regard de la nature, de la longueur et de la difficulté de l'épreuve, le jury observe que les connaissances informatiques semblent globalement acquises, avec une maîtrise variable des langages Python et SQL. Quelques rares candidats ont visiblement négligé la formation, leurs copies conduisent à des notes généralement très faibles.

La moitié des copies corrigées aborde près de 75 % des questions. Un très faible pourcentage traite moins de 20 % des questions.

Quelques candidats rendent des copies d'excellentes qualités. Le jury regrette le niveau parfois très bas d'autres copies. Il serait souhaitable que les candidats mesurent l'importance de la formation initiale en informatique pour la suite de leurs études mais également pour leur avenir d'ingénieur et de citoyen.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Les compétences en matière de programmation élémentaire semblent acquises chez le plus grand nombre de candidats. Néanmoins, le jury souhaite attirer l'attention des futurs candidats sur les points suivants.

- La notion de *type* est essentielle en informatique. Il convient d'en tenir compte lors de la manipulation d'objets informatiques. En particulier, les candidats doivent s'interroger sur la pertinence et les limites de certaines opérations effectuées sur ou entre objets de même type. Par ailleurs, le type d'un résultat renvoyé par une fonction doit être respecté.
- La *complexité* est souvent estimée au regard du nombre d'opérations effectuées dans tout ou partie d'un code. Un minimum d'explications est attendu pour justifier le résultat qui doit, en outre, être exprimé avec les notations strictes de l'énoncé.

- Les *requêtes SQL* doivent faire l'objet d'une attention particulière. Si les questions sur les bases de données sont quasi-systématiquement abordées, peu de candidats obtiennent tous les points en raison d'une syntaxe SQL parfois approximative et d'une maîtrise insuffisante de la notion de jointure.
- Les codes sont globalement syntaxiquement corrects et lisibles. Leur lecture révèle toutefois une écriture au fil de l'eau. Préalablement à l'écriture d'une fonction, il serait souhaitable que chaque candidat s'interroge sur l'*organisation et la structure logiques des codes* qu'il propose.
- Les *commentaires* sont nécessaires mais il convient de ne pas surcharger inutilement les codes avec des paraphrases. Un commentaire n'a de sens que s'il apporte une information utile et pertinente pour comprendre le code. Les docstrings (documentation placée immédiatement après la définition d'une fonction) sont toujours utiles en pratique mais généralement pas attendues sur une copie dans le cadre d'une épreuve de concours en temps limité.
- Le sujet propose une *annexe* détaillant la syntaxe et le rôle de fonctions spécifiques. Certains candidats semblent ne pas avoir lu cette annexe. Par exemple, la fonction **EXTRACT** nécessaire à l'écriture d'une requête n'a été utilisée que par très peu de candidats. Le jury attire l'attention sur la nécessité de lire le sujet dans son ensemble et de savoir extraire les informations des annexes pour répondre efficacement à certaines questions.
- Un *malus* a pu être appliqué à des copies qui ne respectaient pas certains critères de présentation : lisibilité de l'écriture, respect de la langue, clarté de l'expression, propriété de la copie, identification des questions, mise en évidence des résultats. Ils ne constituent aucunement une nouveauté pour les candidats qui, au cours de leur formation, sont invités par leurs professeurs à appliquer ces consignes. Malheureusement, quelques candidats continuent de ne pas le faire !

Signalons par ailleurs quelques erreurs de syntaxe fréquemment rencontrées :

- écriture à l'*envers* des affectations : `10 = a;`
- gestion parfois maladroite des bornes des `range` ;
- *syntaxes incorrectes* comme `for i in range(n) and i!=j:`.

Sont présentées ci-dessous quelques remarques relatives à chaque question de l'énoncé.

I Géométrie

Q1. La fonction `vec` renvoie un vecteur, c'est-à-dire un objet de type `np.array`. Certaines réponses définissent une fonction qui renvoie une liste. Il convient de respecter le typage des fonctions.

Q2. Là encore, de nombreuses réponses ne respectent pas le typage de la fonction. En particulier, certaines renvoient un vecteur au lieu d'un flottant. Certains candidats pensent que le produit `*` de deux tableaux `numpy` renvoie un scalaire alors qu'il renvoie un tableau `numpy` de même forme. Des candidats calculent le cosinus de deux vecteurs `v1` et `v2` avec `np.cos(v1, v2)`

Q3. Dans le calcul de la norme d'un vecteur, la racine carrée est parfois oubliée et `*` est pris pour un produit scalaire.

Q5. La fonction `pt` est parfois mal comprise. Dans certaines copies, on peut lire *la fonction renvoie un point sur la droite alors qu'elle renvoie un point à distance bien définie d'un point donné*.

Q6. La fonction `ra` a parfois été utilisée à la place de la fonction `norme`.

Q7. Si la question est relativement bien traitée dans l'ensemble, quelques candidats essaient de berner le correcteur en remontant le calcul... Ajoutons que cette question demande une condition nécessaire et suffisante. Certaines réponses ne présentent qu'une condition suffisante.

Q8. La condition $t > 0$ est souvent oubliée.

II Optique

Q9. Cette question attend un minimum de justification. Ne pas hésiter à faire un schéma, même simple, pour illustrer et appuyer une argumentation. Beaucoup de réponses se perdent dans des considérations de normes.

Q10. Cette question est généralement bien traitée quand la Q9 est correcte.

Q11. Le test de visibilité est souvent omis et le traitement du cas $i == j$ souvent oublié.

Q12. Le signe moins est quasi-systématiquement oublié.

Q13. L'écriture de la fonction requiert l'établissement d'une relation définissant le vecteur \vec{w} de l'énoncé. Ce travail, mené de manière maladroite, mène souvent à des expressions fausses. Là encore, un schéma simple peut grandement aider.

III Enregistrement des scènes

Q14. Cette requête élémentaire vérifie la connaissance de la syntaxe `SELECT ... FROM ... WHERE ...` mais également la capacité à utiliser la fonction `EXTRACT` définie dans l'annexe. Cette dernière est souvent absente des réponses.

Q16. Cette question est traitée de manière très inégale, certaines jointures étant incorrectes.

Q17 Cette question est très peu abordée.

IV Lancer de rayons

Q18. Les décalages $N/2$ et $1/2$ ont visiblement posé des difficultés. Un schéma de la discréétisation du plan, avec les notations de l'énoncé, permet souvent de clarifier la situation. En particulier, son annotation évite l'inversion de j et i dans l'expression du résultat renvoyé par la fonction.

Q20. Certaines réponses utilisent la fonction `sor`t pour extraire le minimum d'une liste ! Avant de proposer une solution, s'interroger sur sa pertinence et son efficacité. De nombreux candidats renvoient *le premier point d'intersection rencontré lors du parcours de la liste* `Objet` au lieu du *premier au sens du parcours du rayon lumineux*.

Q21. La variable de sortie est souvent initialisée à 0 alors que le résultat est un tableau. En outre, la variable globale `noir` n'est pas intégrée dans les réponses.

Q22. La question est plutôt bien réussie mais le tableau renvoyé par la fonction `lancer` n'est pas initialisé correctement (tableau (N, N) au lieu de $(N, N, 3)$), quand l'initialisation n'est pas oubliée !

Q23 et Q24. Trop de candidats annoncent les complexités sans justification. Une complexité quadratique doit être justifiée, généralement par la présence de deux boucles imbriquées contenant des opérations en temps constant.

V Améliorations

Q25. Des candidats appellent plusieurs fois la fonction `interception` avec les mêmes arguments dans le corps d'une boucle alors qu'un seul appel suffit. Éviter les calculs redondants et inutiles.

Q27. Comme pour Q22, l'initialisation de la variable renvoyée par la fonction est souvent absente ou incorrecte.

Q28 à Q30. Ces questions ont été très peu abordées.

Conclusion

Les résultats à cette épreuve montrent que les étudiants, soutenus par leurs professeurs, ont acquis des compétences en informatique. Le jury encourage les futurs candidats à travailler l'informatique en alliant réflexion sur feuille de papier et mise en œuvre des algorithmes sur ordinateur.