

## MINIPROJET 5

### → Une version (simplifiée) du jeu CITADELLES

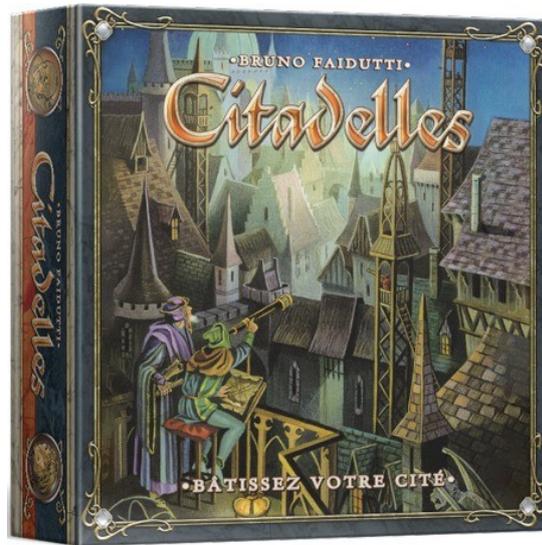
L'objectif est de créer en python un programme capable de réaliser le jeu CITADELLES en tour-par-tour.

Le jeu est décrit ici

<https://www.regledujeu.fr/citadelles/#tableau>

Les cartes sont accessibles ici :

<https://citadellesjeudecartes.wordpress.com/cartes-de-quartier/>



### MVP

Le jeu se déroule pour 4 joueurs.

Le programme python doit traiter pour chaque tour :

- Le **choix des personnages** pour chaque joueur parmi une liste de 8 dictionnaires (1 carte personnage est représentée par 1 dictionnaire)  
On écarte de façon visible 2 cartes personnages, et 1 autre carte personnage face cachée.  
Chaque joueur dans l'ordre choisit sa carte personnage.
- Les **actions** des joueurs.  
Les joueurs sont appelés dans l'ordre (Assassin, Voleur, etc. voir règles) et agissent :  
**1ère étape**  
→ Le joueur gagne autant de pièces qu'il a de quartier de la couleur de sa carte personnage

### 2ème étape

- Soit ils tirent 2 cartes quartier et en garde une en main (l'autre est défaussée)
- Soit ils récupèrent 2 pièces

### 3ème étape

→ Le joueur peut « construire » un quartier en payant le nombre de pièces associé à la carte quartier. La carte quartier correspondante peut être posée.

(Note : à chaque étape, on modulera les actions en fonction des caractéristiques du personnage choisi !)

Il faudra également traiter une **vérification** de « fin de partie » et le **comptage des points**.

## Evaluation ( /42 points)

### Fonctionnalités minimales demandées :

<ul style="list-style-type: none"><li>Structures de données adaptées (chaînes, tuples, liste, dico)</li></ul>	<i>12</i>	<ul style="list-style-type: none"><li>Action de l'étape 3 (construction de quartier)</li></ul>	<i>12</i>
<ul style="list-style-type: none"><li>Choix des personnages</li></ul>	<i>14</i>	<ul style="list-style-type: none"><li>Action de l'étape 2 (tirage de cartes quartier)</li></ul>	<i>12</i>
<ul style="list-style-type: none"><li>Action de l'étape 1 (gain de pièces)</li></ul>	<i>12</i>	<ul style="list-style-type: none"><li>Prise en compte des actions spécifiques de chaque personnage (0,5 / perso)</li></ul>	<i>14</i>
<ul style="list-style-type: none"><li>Gestion de la fin de partie et détermination du vainqueur</li></ul>			<i>14</i>

### Fonctionnalités plus : (+1pt par fct)

<ul style="list-style-type: none"><li>Interface graphique</li></ul>		<ul style="list-style-type: none"><li>...</li></ul>	
---	--	---	--

### Code :

<ul style="list-style-type: none"><li>Lisibilité du code, clarté</li></ul>	<i>12</i>	<ul style="list-style-type: none"><li>Utilisation efficaces de fonctions</li></ul>	<i>12</i>
<ul style="list-style-type: none"><li>Variables explicites</li></ul>	<i>12</i>	<ul style="list-style-type: none"><li>Utilisation de boucles (for ou while)</li></ul>	<i>12</i>
<ul style="list-style-type: none"><li>Commentaires au fil du code</li></ul>	<i>12</i>	<ul style="list-style-type: none"><li>Documentation de fonctions</li></ul>	<i>12</i>

## Modalités :

Semaine 1 (2h)	Analyse	Découverte du projet. Analyse des besoins et identification de tâches. Répartition du travail.
	Sprint #1	
Semaine 2 (2h)	Daily Meeting	Bilan et mise en commun du travail réalisé. Difficultés ? Réussites ? Nouvelle répartition des tâches.
	Sprint #2	

Semaine 3 (2h)	Daily Meeting	Bilan et mise en commun du travail.  Détermination des tests à effectuer pour valider.  Nouvelle répartition des tâches.
	Sprint #3	Rendu final
Semaine 4 (2h)	Bilan / Présentation	Présentation des travaux réalisés

## **BILAN COMPETENCES**

<b>REA</b>	<ul style="list-style-type: none"> <li>Imaginer et concevoir une solution modulaire : décomposer en blocs, se ramener à des sous-problèmes simples et indépendants</li> </ul>			
	Je n'écris pas de fonctions.	J'écris et utilise quelques fonctions.	J'écris et utilise des fonctions que je documente.	J'écris et utilise les fonctions les plus adaptées au problème. Je les documente précisément.
	<ul style="list-style-type: none"> <li>Concevoir, décrire une structure de données permettant de résoudre le problème</li> </ul>			
	Je décris une structure de données incomplète, ou inadaptée.	Je décris une structure de données correcte pour répondre au problème, sans l'utiliser complètement.	Je décris et utilise une structure de données adaptée pour répondre au problème.	Je décris et utilise efficacement la meilleure structure de données adaptée au problème.
<b>VAL</b>	<ul style="list-style-type: none"> <li>Justifier et critiquer une solution.</li> </ul>			
	Les choix ne sont ni compris, ni justifiés.	Je justifie seulement certains choix.	Je justifie l'ensemble des choix (algo et structures de données).	Je justifie clairement l'ensemble des choix (algo et structures de données). Je précise les solutions alternatives possibles.
<b>COM</b>	<ul style="list-style-type: none"> <li>Communiquer à l'oral en utilisant un langage rigoureux et des outils pertinents</li> </ul>			
	Je donne les explications demandées en quelques phrases courtes et générales.	J'explique globalement les méthodes et l'utilisation des fonctions. Le vocabulaire reste peu précis et général.	Je détaille les méthodes de calculs. Mes variables et fonctions sont évoquées, dans un vocabulaire correct.	J'utilise un vocabulaire recherché et adapté au propos. J'utilise des exemples qui précisent mes explications. Toutes mes variables et sont détaillées (non, type) et les fonctions sont documentées (entrées/sorties)
	<ul style="list-style-type: none"> <li>Coopérer au sein d'une équipe dans le cadre d'un projet</li> </ul>			
	<i>Pas de répartition des tâches. Travail centralisé.</i>	Répartition des tâches efficaces, mais peu efficace. Mise en commun difficile.	Travail par répartition des tâches. Mises en commun régulières.	Travailler à plusieurs de façon efficace. Répartition précise des tâches, en fonction des capacités de chacun. Utilisation d'outils collaboratifs.