

La Caravane d'Albâtre

Au cœur des vastes étendues désertiques de l'Égypte antique, une caravane mystique, connue sous le nom de la Caravane d'Albâtre, se frayait un chemin à travers les sables brûlants. Les chameliers guidaient leurs bêtes, chacune parée de plaques d'albâtre étincelantes, créant une lumière aveuglante qui jouait avec les rayons du soleil. Ces derniers étaient renvoyés au point de rendre discrète la progression de la caravane, rendue presque invisible de loin.

Sa mission était honorable : rassembler les richesses nécessaires à l'acquisition d'une barque de commerce majestueuse, capable de naviguer avec grâce le long du fleuve sacré du Nil.

La caravane traversait des dunes sans fin, bravant les tempêtes de sable et les vents arides. Les étoiles servaient de guides dans l'obscurité silencieuse du désert. Chaque marche était un pas de plus vers le destin de cette communauté, une épopée d'endurance et de persévérance. Les porteurs éprouvés rêvaient de marchandises transportées en douceur sur l'eau, telle une offrande au dieu Nil lui-même.



Musique de motivation et de concentration pour le développement :



Introduction

La Caravane d'Albâtre est un jeu de gestion se déroulant en Égypte antique, en tour par tour. Le but est de faire survivre la caravane et ses occupants en répondant à leurs besoins, tout en accumulant des richesses via le commerce qui nécessite des déplacements quasi constants.

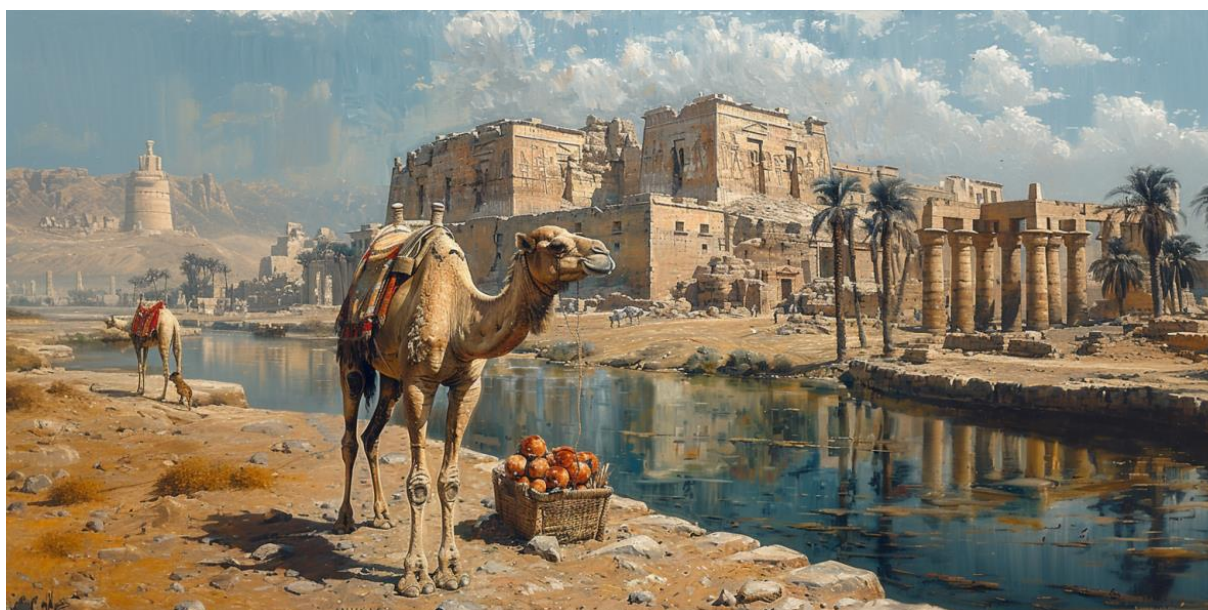
Données

Avant de détailler les fonctionnalités (actions possibles du joueur et leurs effets, événements de jeu...), voici la liste des données cruciales au fonctionnement (autrement dit, les principales variables). Les données étant affichées au joueur, il faudra mettre à jour l'élément de l'interface correspondant en cas de modification (par exemple, si de l'eau est utilisée, la variable devra être diminuée et ce changement répercuté dans la page web, via une mise à jour du code HTML.

Caravane

Propriétés générales (peuvent varier en fonction des actions du joueur, de l'écoulement du temps, d'événements...) : nom français et nom technique, type et valeur en début de partie :

- Troupe totale (**crew**) : entier : 20 personnes au départ
- Moral (**morale**) : décimal affiché arrondi : 50% au départ
- **Debens** (la monnaie égyptienne) : entier : 1000 au départ
- Autorité (**authority**) : entier : 10 au départ
- Eau (**water**) : entier : 50 au départ
- Tour résolu (**resolved**) : booléen : false au départ (le joueur n'a pas encore donné des ordres à la troupe et cliqué sur le bouton « résoudre »)
- Tours de famine (**famineTurns**) : entier : 0 au départ (pas de famine en début de jeu)
- Inventaire (**resources**) : liste du nombre d'armes, de chameaux, de chevaux, de rations (nourriture), de papyrus (c'est la liste exhaustive des ressources du jeu basique et qui peuvent être achetées ou vendues, contrairement à l'eau). En JS, il s'agit d'un objet contenant la liste des ressources que transporte la caravane et pouvant être commercées. Liste de départ : `{weapons: 3, camels: 1, horses: 2, rations: 80, papyrus: 10}`



Remarque sur les noms techniques (ie noms de variables) : pour éviter des problèmes de compréhension étant donné la relative complexité du projet, son descriptif est en français. Mais l'interface du jeu lui-même, et surtout son code source, *devront être en anglais* (ce qui est une bonne pratique pour n'importe quel projet web ou logiciel). C'est pourquoi il est donné en gras le nom des variables en anglais correspondant aux ressources.

Propriétés modifiables directement par le joueur (sous certaines conditions et dans certaines limites) : les *assignments* (le joueur va gérer son équipage en assignant des membres à diverses tâches) :

- Nombre d'éclaireurs (**scouts**) : contraintes : nombre de chevaux (chaque éclaireur est monté)
- Nombre de gardes (**guards**) : contraintes : nombre d'armes (on ne peut pas avoir plus de gardes que d'armes)
- Nombre de porteurs (**carriers**) : contraintes : leur nombre doit être inférieur au total des gens composant la caravane (*crew*). Ils permettent à la caravane de se déplacer si la charge des marchandises est trop élevée pour les chameaux, mais plus le nombre de porteurs est élevé, plus le coût en moral est important.

Remarque : d'autres assignments sont possibles et liés aux lieux, cf ci-après.

Par ailleurs, concernant tous les assignments hormis les porteurs (donc ceux listés ci-dessus + ceux des lieux décrits plus bas), il faut respecter une contrainte globale logique : la somme de toutes affectations doit être inférieure ou égale au total de troupe totale. Ainsi, si le *crew* est seulement de 5, même avec 5 chevaux et 5 armes, on pourra avoir au maximum 2 scouts et 3 gardes, ou 5 scouts et 0 garde, etc. Autrement dit, $scouts + guards \leq crew$

A noter que l'assignment des porteurs se fait après la résolution du tour (on commence par ramasser des ressources, qui sont comptées pendant la résolution et qui seront ensuite à porter), durant laquelle tous les assignments sont mis à 0 : les porteurs ne sont donc limités que par le nombre total de gens. La logique est qu'une fois que les membres de la caravane ont rempli leurs différents rôles (éclaireur, récolteur...) ceux-ci peuvent ensuite être mobilisés pour devenir porteurs si les chameaux ne suffisent pas.

Propriétés calculées (dépendent uniquement des autres variables de la caravane : si vous avez besoin de les recalculer à plusieurs endroits, vous pouvez créer une fonction *refreshCaravanProperties()* et l'appeler quand nécessaire, ie quand les dépendances ont été modifiées) :

- Personnes inoccupées (**idle**) : la différence entre le total de l'équipage et tous les assignments (y compris les assignments de lieu décrits plus loin) : c'est une information utile à afficher pour l'utilisateur
- Portage courant (**currentPorterage**) : chaque porteur humain (*carrier*) assigné peut soulever 10 unités de poids (nommées *heqat*) et chaque chameau peut porter 100 unités
- Portage nécessaire (**neededPorterage**) : le nombre total de ressources et d'eau que possède la caravane (chaque item pesant 1 *heqat*)

Lieux

Au cours de ses pérégrinations, le joueur aura l'occasion de découvrir toutes sortes de lieux. Par nature, une caravane a vocation à se déplacer afin d'acheter des marchandises dans un lieu et les revendre dans un autre lieu. La principale source de profit pour la caravane du jeu est précisément d'acheter des *papyrus* dans une *cit  fluviale* et les revendre dans une *cit  du d sert*. L'une se trouve en cherchant le long des berges du Nil, l'autre en s'enfon ant dans le d sert...

Les propri t s des lieux varieront selon leur type, qu'il s'agisse des ressources qu'on peut y collecter et celles qu'on peut  ventuellement commercer. Concernant les ressources librement accessibles, 3 peuvent  tre trouv es et collect es (suivant le lieu, telle ou telle ressource sera pr sente et collectable gratuitement), d'o  3 nouveaux types d'assignements qui suivent.

Propri t s de lieux modifiables directement par le joueur :

- Collecteurs d'eau (**waterGatherers**) : chaque personne assign e ram nera une certaine quantit  d'eau en fonction du rendement du lieu (par exemple, si le rendement est de 3 pour l'eau, une personne ram nera 3 unit s d'eau)
- R colteurs de nourriture (**foodHarvesters**) : m me principe, le nombre de rations r colt es sera le nombre de personnes assign es x le rendement local en nourriture
- Gens en permission/fl neurs (**strollers**) : afin de remonter le moral de vos gens, vous pouvez en envoyer en permission : chaque personne prenant l'air remonte le moral moyen du groupe, suivant l'int r t du lieu

Voici la liste d taill e de chaque lieu (6 au total).

Le premier lieu est le lieu de d part de la caravane : le bord du Nil (**nil_shore**). On peut y r colter beaucoup d'eau par collecteur (**waterYield**  lev ) et un peu de nourriture (**foodYield**).

```
{  
  type: 'nil_shore',  
  data: {waterYield: 3, foodYield: 1},  
}
```



Pour rappel, on collecte les ressources d'un type donné suivant la formule : *nombre de personnes assignées à la collecte de cette ressource * rendement de cette ressource*.

Par exemple sur ce lieu, le rendement de collecte de l'eau est de 3 unités pour un collecteur, contre 1 unité de nourriture pour un collecteur. Ainsi 5 collecteurs affectés à la collecte d'eau rapporteront 15 unités d'eau tandis que 5 collecteurs affectés à la collecte de nourriture ne ramèneront que 5 nourritures (rations) !

Si on croise peu de monde sur les berges du Nil (pas de commerce possible), on peut tomber sur des légionnaires en patrouille de temps en temps et qui exigent des « taxes », nous y reviendrons...

Enfin, pour ce qui est de l'après-collecte, voilà où on peut se rendre à partir de ce lieu : on peut se rendre à coup sûr dans un *Village*, poursuivre sur le *bord du Nil*, aller dans le *désert* et si on en a trouvé une (avec un peu de chance et/ou des éclaireurs), aller dans une *Cité fluviale* !

Un second lieu un peu plus complexe est le **village** : il s'agit d'un petit lieu sécurisé où on peut y faire quelque commerce en plus de puiser de l'eau ou se promener. On peut y acheter les rations les moins chères (en ville, la nourriture coûte plus cher) et même des chevaux ! Dans l'objet de données suivant, vous pouvez lire que chaque ration coûte 4 debens, et le village peut en fournir jusqu'à 100 (« volume »).

```
let moraleYield = crew <= 10 ? 4 : 2;  
location = {type: 'village', data: {waterYield: 2, moraleYield:  
moraleYield, sell: [{type: 'rations', volume: 100, price: 4}, {type: 'horses',  
volume: 3, price: 500}]}};
```



Remarque : afin d'éviter une défaite par manque de moral si le nombre de gens composant la caravane a trop diminué, les gains de moral par personne prenant un quartier libre sont doublés si l'équipage est inférieur ou égal à 10 : par exemple une personne se promenant dans un village haussera le moral de 4% au lieu de 2%.

A partir d'un village, on peut se rendre à coup sûr dans un autre *Village*, poursuivre sur les *berges du Nil*, ou aller dans le *désert*. En réalité, depuis n'importe quel lieu, on peut aller dans ces 3 lieux. Les lieux plus spécifiques nécessitent d'avoir eu de la chance et de chercher dans un lieu approprié.

En voici le détail :

- pour trouver une cité fluviale, il se trouve (la recherche est faite automatiquement au moment du clic sur « Resolve ») sur les berges du Nil et après résolution du tour, l'option de se déplacer vers une cité fluviale apparaîtra
- pour trouver une cité du désert, c'est le même principe il faut chercher depuis un désert ou une oasis
- pour trouver une oasis, il suffit de se déplacer vers un désert : on a alors une chance de tomber immédiatement sur une oasis au lieu d'un terrain désertique

Avant de se rendre dans le désert pour y trouver une cité où vendre du papyrus, il faut trouver une cité fluviale (**fluvial_city**) en errant sur les bords du Nil (ou en y campant, cela peut suffire car des éclaireurs peuvent être envoyés). Avec un peu de chance (améliorée par l'usage d'éclaireurs), on finira par en trouver une. On pourra y récolter de l'eau et un fort moral si on s'y promène, et surtout on peut y acheter des papyrus (ainsi que des rations assez onéreuses). Accessoirement on peut y acheter ou y vendre des armes (si jamais on a récupéré des surplus suite à des combats avec des légionnaires).

```
let moraleYield = crew <= 10 ? 6 : 3;  
location = {type: 'fluvial_city', data: {waterYield: 3, moraleYield:  
moraleYield, buy: [{type: 'weapons', volume: 20, price: 100}], sell: [{type:  
'papyrus', volume: 50, price: 20}, {type: 'rations', volume: 100, price: 6},  
{type: 'weapons', volume: 20, price: 200}]}};
```



Une fois du papyrus récupéré, la caravane peut chercher des acheteurs en s'enfonçant dans les terres. Elle va devoir traverser le désert, éventuellement des oasis, avant de trouver une cité du désert. Comme on peut le constater, il n'y a pas grand-chose à faire dans le désert, et il vaut mieux avoir des réserves d'eau et de nourriture avant de s'y aventurer !

```
{type: 'desert', data: {}}
```



Si on est chanceux, au lieu de tomber sur un désert, on tombera sur une oasis qui fournit eau et nourriture en quantité :

```
{type: 'oasis', data: {waterYield: 3, foodYield: 3}}
```

Attention néanmoins, après chaque tour il y a déplétion (diminution de la quantité) de chaque rendement de l'oasis, si au moins une personne a été assignée à la récolte de la ressource correspondante. Ainsi, si 5 personnes ont été assignées à la collecte d'eau et 0 à la collecte de nourriture, *waterYield* diminuera de 1 tandis que *foodYield* restera à 3.



Par ailleurs, des bandits ont tendance à attendre les voyageurs dans les oasis pour les détrousser... Nous en reparlerons dans la section dédiée aux combats.

Enfin, le but du voyage peut apparaître aux yeux ébahis de la caravane : une cité du désert où on pourra vendre avec un beau bénéfice les papyrus acquis dans une cité fluviale. L'eau s'y trouve en quantité limitée et les rations y sont chères, mais l'équipage pourra y retrouver du moral après sa traversée du désert. On y trouve des chameaux à la vente (ce qui permettra par la suite de transporter davantage de denrées, et sans épuiser les porteurs humains).

```
let moraleYield = crew <= 10 ? 6 : 3;  
location = {type: 'desert_city', data: {waterYield: 1, moraleYield:  
moraleYield, buy: [{type: 'papyrus', volume: 50, price: 40}], sell: [{type:  
'rations', volume: 100, price: 8}, {type: 'camels', volume: 2, price:  
1000}]}};
```



Déroulé global d'un tour de jeu

Un tour de jeu se résume essentiellement en deux phases :

- 1) Une phase d'affectation (et éventuellement d'actions supplémentaires réalisées dans le lieu, comme le commerce) clôturée par une résolution instantanée (au clic sur le bouton « Resolve ») qui va calculer les ressources récoltées.
- 2) Une phase liée au déplacement : ajustement du nombre de porteurs (en fonction du nombre de ressources totales qu'il faut transporter) et choix de la prochaine destination parmi un certain nombre de possibilités listées par des boutons (ou autre type d'interface), suivi de la consommation nécessaire au bon fonctionnement de la caravane.

En particulier, la phase 1 va remettre tous les assignements à 0 (sauf les éclaireurs par commodité pour le joueur, donc les personnages inoccupés augmenteront considérablement), et déterminer si on a trouvé une cité (fluviale ou du désert, selon la probabilité donnée dans la section déplacement). Par ailleurs, et préalablement à la remise à 0 des assignements, la collecte des ressources du lieu où se trouve la caravane sera exécutée (les assignements sur le lieu pourront permettre de récolter nourriture, eau ou moral), avec une éventuelle déplétion des rendements si on se trouve sur une oasis.

A l'issue de la phase 2, que la caravane reste sur place ou bouge, on tente de répondre aux besoins vitaux de l'équipe, les porteurs redeviennent disponibles (l'assignement correspondant passe à 0 donc les inoccupés augmentent d'autant) et on vérifie que la partie n'est pas perdue. Voilà cette fin de tour en détails :

- a) Répondre aux besoins : le moral baisse de 5% automatiquement, et par ailleurs, il baisse de 1% pour chaque porteur assigné. Par ailleurs, le stock d'eau est diminué par le nombre de personnes de la caravane, sauf si celle-ci reste sur place, dans ce cas la caravane ne consomme qu'une quantité d'eau égale à la moitié de l'équipage (tronquée à l'inférieur, cf *Math.floor()*). Si de l'eau venait à manquer, chaque unité d'eau manquante cause la mort d'une personne ! Concernant le stock de rations, celui-ci diminue en situation normale de 1 par personne. Sinon, s'il n'y en a pas suffisamment, les rations restent dans le stock et il y a famine jusqu'à ce qu'il y ait suffisamment de rations pour nourrir les survivants. La famine engendre des pertes humaines plus progressivement que le manque d'eau (mais aussi une baisse de moral de 10 par tour !), il faut donc compter le nombre de tours de famine, et les pertes seront appliquées selon ce calcul (les pertes sont nulles ou faibles au début puis exponentielles) : $\text{Math.pow}(2, \text{famineTurns}) - 2$ où famine vaut 1 le premier tour de famine.
- b) Vérifier les conditions de défaite : si la population de la caravane passe à 0, la partie est perdue. Si le moral est inférieur ou égal à 0, il y a une révolte et le joueur perd le pouvoir, sauf si son autorité est suffisante pour stopper la révolte. On va donc tester si l'autorité permet de contrebalancer le déficit de moral (on vise 1 de moral : par exemple, si on se trouve à -5 de moral, on regarde si on a au moins 6 d'autorité) : si oui le moral sera mis à 1 et l'autorité diminuera autant qu'il en fallait pour ramener le moral à 1.
- c) Déterminer si une rencontre hostile a eu lieu : détaillé dans la section Combats. Si on se trouve au tour prochain sur les bords du Nil ou dans une oasis, une rencontre peut se produire avec des légionnaires ou des bandits.

Actions

Déplacements

En fin de tour de jeu, le joueur peut décider de rester sur place (l'action « Park », ce qui consommera deux fois moins d'eau et permet de rester dans un lieu intéressant comme une oasis). Sinon, il va tenter de changer de lieu : pour ce faire, il faut que la capacité de portage courant de la caravane (chameaux + porteurs humains) soit supérieure ou égale au portage nécessaire, sinon une pop-up s'affichera et le mouvement sera annulé (il pourra être retenté en attribuant plus de gens au portage ; si jamais il n'y a pas assez de monde, rester sur place pourrait devenir un choix obligatoire si une fonctionnalité d'allègement des marchandises n'a pas été implémentée et s'il n'y a pas de marché sur le lieu courant pour s'alléger).

Pour rappel, si la caravane se dirige vers un désert, elle a une chance de tomber sur une oasis. Cette probabilité est la suivante :

```
const PROBABILITY_FIND_OASIS = 0.25
```

Cette probabilité est augmentée de 0.1 par éclaireur (*scout*) assigné ! D'une manière générale, il est conseillé au joueur de toujours avoir des éclaireurs assignés (le jeu pourrait sauvegarder cet état pour éviter au joueur de les réassigner chaque tour...).

Par ailleurs, à la fin de la phase 1 d'un tour (après le clic sur le bouton « Resolve » *ou autre nom pertinent si vous en trouvez un*), le joueur peut découvrir une cité fluviale s'il se trouve sur les bords du Nil, ou une cité du désert s'il se trouve dans un désert ou une oasis. Cette probabilité est la suivante :

```
const PROBABILITY_FIND_CITY = 0.33
```

Cette probabilité est aussi augmentée de 0.1 par éclaireur assigné.

Astuce : voir la fonction *Math.random()* pour générer un nombre aléatoire entre 0 et 1, qui sera à comparer avec la probabilité cible.

Combats

Certains lieux sont risqués : des légionnaires ayant une tendance à la corruption patrouillent le long des berges du Nil et « taxent » (ou extorquent) les honnêtes marchands, et des bandits attendent tapis dans l'ombre des oasis.

Voilà les probabilités d'être interceptés :

- si on se dirige ou reste vers les bords du Nil :

```
const PROBABILITY_LEGIONARIES_HALTING = 0.33
```

- si on se dirige ou reste dans une oasis

```
const PROBABILITY_BANDITS_HALTING = 0.5
```

Le nombre d'assaillants est aussi aléatoire mais encadré par un intervalle et s'adaptera à la taille de l'équipage si celui-ci croît :

Pour les légionnaires : `getRandomIntInclusive(3, Math.max(5, Math.round(crew/4)))`

Pour les bandits : `getRandomIntInclusive(5, Math.max(10, Math.round(crew/2)))`

Pour les légionnaires, qui sont à nombre égal plus puissants que des bandits (ou même les gardes de la caravane), il est possible de verser un pot-de-vin (donc de les corrompre) avec une somme qui dépend de la formule ci-dessous : `getRandomIntInclusive(20,50)*nbEnemies`

Autrement, il faudra se battre en nommant des gardes et en les équipant d'armes. Si, alors qu'un combat a lieu, le joueur n'a assigné aucun garde, quand il tente de cliquer sur « Resolve », il faudra le prévenir qu'il est certain de perdre le combat s'il ne met personne en défense.

Malheureusement, si un combat a eu lieu, les personnes affectées au rôle de garde ne pourront pas être réaffectées ailleurs (par exemple pour récolter), on ne nomme donc des gardes que quand c'est nécessaire.

Les combats ne sont pas au cœur du jeu, nous allons donc en limiter la complexité.

Nous allons définir la valeur de combat de la caravane comme suit : $2 * \text{nombre de gardes}$

Il faudra comparer cette valeur suivant l'adversaire à $3 * \text{nombre de légionnaires}$ ou $1 * \text{nombre de bandits}$

Ainsi puisque 3 légionnaires ont une valeur totale de 9, il faudrait 5 gardes pour les vaincre (par ailleurs, le joueur remporte les égalités).

Même si le joueur remporte le combat, il aura des pertes aléatoires entre 0 et le nombre de gardes. Vous pouvez utiliser la fonction suivante pour générer ce nombre :

```
function getRandomIntInclusive(min, max) {  
    min = Math.ceil(min);  
    max = Math.floor(max);  
    return Math.floor(Math.random() * (max - min + 1)) + min;  
}
```

Les armes des gardes morts sont récupérées en cas de victoire.

Les armes des légionnaires sont de bonne qualité et peuvent être également récupérées en cas de victoire.

Si le joueur perd le combat, tous ses gardes sont morts et les armes perdues. Le moral et l'autorité sont divisés par deux, et la caravane a fui automatiquement dans le désert !



Si vous en avez envie, vous pouvez inventer un système de combat avancé. Contraintes : les Romains plus résistants et puissants que les bandits, qui sont eux plus nombreux. Vos gardes surpassent en tous points les bandits mais sont moins résistants (pas forcément moins bons en attaque) que les légionnaires.

Commerce

Le commerce est possible dans les villes et villages. Les denrées disponibles diffèrent selon qu'il s'agit d'une ville ou d'un village, mais aussi du type de la ville (désertique ou fluviale). La liste de denrées, leur volume maximal et leur prix ont été détaillés dans la section Lieux.

Concernant l'acte de commerce en lui-même, pour qu'une transaction d'achat d'un volume X sur une denrée Y soit valide, il faut :

- Que la localité vende bien ledit produit
- Que le volume qu'on tente d'acheter ne soit pas supérieur au stock du lieu
- Que le coût de la transaction (volume * prix) ne soit pas supérieur à la trésorerie (en debens) de la caravane

Concernant la vente, le principe est le même, mis à part que le produit de la vente est en quelque sorte taxé : en fait l'équipage de la caravane est rémunéré via le fruit des ventes. Le taux de prélèvement en % est :

```
const crewPayRate = 5
```

Il faudra donc multiplier la recette de la vente par $(1 - \text{crewPayRate}/100)$ et arrondir pour obtenir le nombre de debens réellement perçus par le joueur.



Design

Aucune piste de design et d'UI (user interface) n'est fournie, pour laisser place à votre créativité tant au niveau du style graphique que de l'agencement de l'interface utilisateur (et plus largement l'ergonomie).

La seule contrainte est d'expliquer le système de jeu au nouvel utilisateur (soit via une pop-in arrivant au chargement du jeu pouvant utiliser textes et schémas, soit via d'autres moyens plus progressifs).

Pour vous aider sera fourni un cours d'ergonomie, que vous pourrez lire notamment à partir de la slide 21 et jusqu'à la 33 pour connaître quelques notions telles que :

- la loi de proximité
- la loi de similarité
- la loi de Fitz
- l'affordance

Vous pourrez faire aussi du benchmarking (slide 42) de jeux existants.

Idées bonus

Carnet de bord

Pour aider le joueur à comprendre ce qui se passe, un Carnet de bord (**logbook**) pourrait afficher des messages au joueur sur les événements qui se produisent, les résultats de ses actions...

Exemple de messages :

- « Your scouts found a fluvial city ! It may produce papyrus. »
- « Your scouts found a desert city ! Would be interested by water products like papyrus. »
- « Water missing ! One person dies for each unit missing. »
- « Famine since "+famineTurns+" turn(s), morale lowered and crew died: "+nbDeathsFamine" »
- « Famine is over ! »
- « Morale was so low you needed authority to keep crew in line ! »
- « You found an oasis with clean water and some game to hunt ! »
- « Victory ! You got their weapons and increased authority ! »
- « Defeated by enemies, you fled to desert ! »

Nouvelle ressource

Outre le papyrus, on pourrait penser à commercer... l'albâtre bien sûr ! Mais où l'achète-t-on (dans un nouveau lieu, comme une carrière) ? Où la vend-on ?

UX des assignments

Afin d'éviter le seul recours à des chiffres pour dire au joueur que tel nombre de personnes est assigné à telle tâche, on pourrait faire figurer le nombre d'affectés (et d'inoccupés) par des petits bonhommes et pas seulement des chiffres, ce qui permettrait d'avoir une overview plus intuitive.

Bruitages

Ajouter des bruitages lors des actions, des déplacements...

Divinités

En se basant sur les divinités égyptiennes, ajouter un temple aléatoire dans chaque ville permettant d'obtenir une bénédiction.

Recrues

Possibilité de recruter de nouveaux membres dans les tavernes (**beerhouse**) des villes, moyennant quelque-chose en échange.

Compétences du joueur

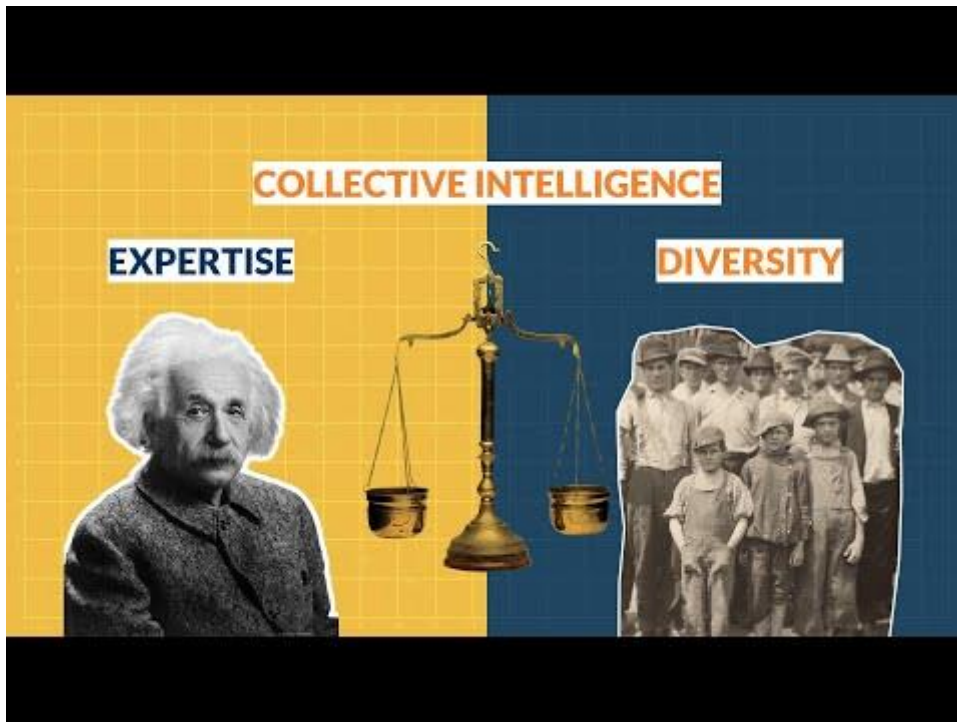
En jouant, il débloquent de nouvelles compétences (marchandage quand il fait du commerce, tactique quand il gagne des combats... par exemple) qui lui procurent des bonus au fur et à mesure qu'il gagne en expérience.

Vos propres idées

Vous pouvez bien sûr implémenter des idées supplémentaires qui vont sont propres !

Miscellanées

Travailler en groupe en utilisant l'intelligence collective : <https://youtu.be/HNQ3GccCAI0?t=515>



La recette de l'intelligence collective

