

TIPE - Sélection naturelle urbaine

Eric et Sylvain

Table des matières

1	Chunks	3
1.1	Chunks adjacents	3
1.2	Biomes	3
1.2.1	Définition	3
1.2.2	Mode de génération des biomes	3
1.3	Altitude	3
1.4	Coefficient d'hostilité environnementale	4
1.5	Coefficient d'accessibilité	4
2	Villages	4
2.1	Définition	4
2.2	Décisions	4
2.3	Chunks exploitables	4
2.4	Code génétique	5
2.4.1	Définition	5
2.4.2	Matrice de décision	5
2.5	Bâtiments	5
2.5.1	Mode d'apparition	5
2.5.2	Types de bâtiments	5
2.6	Tableau récapitulatif	6

1 Chunks

La carte est découpée en chunks notés de 4x4 cases. On note $c_{i,j}$ le chunk contenant la case de coordonnée $(4i, 4j)$. Autrement dit, $c_{i,j}$ est le i -ème chunk horizontal et le j -ème chunk vertical. On note \mathcal{C} l'ensemble des chunks de la carte.

1.1 Chunks adjacents

Un chunk $c_{a,b}$ est adjacent à un chunk $c_{i,j}$ si $i \leq a \leq i+1$ ou $j \leq b \leq j+1$. Ainsi, tout chunk qui n'est pas situé en bordure de carte possède 8 chunks adjacents.

On note $\mathcal{V}(c)$ l'ensemble des chunks adjacents à c et $n_v(c)$ le nombre de chunks voisins à c .

1.2 Biomes

1.2.1 Définition

On associe à chaque chunk un biome (*cf. Mode de génération des biomes*) qui modifie la production des bâtiments, l'accessibilité et les aléas qui peuvent frapper la région. Chaque biome possède aussi une altitude qui modifie le coût de construction des bâtiments, l'accessibilité et les distances entre les villes.

Par exemple, une forêt sera peu vulnérable aux sécheresses mais aura un taux de mortalité plus élevé à cause de la faune locale. On ne pourra pas produire de grain dans un désert froid ou chaud mais on pourra y implanter des mines qui auront une production accrue.

1.2.2 Mode de génération des biomes

Au départ de la simulation, l'algorithme génère deux bruits de Perlin de taille $\frac{L}{4}$ puis les combine afin d'obtenir deux valeurs (h, q) par chunk. La première valeur h est un indicateur d'humidité locale, tandis que la deuxième valeur q est un indicateur de chaleur. Les biomes sont définis par le tableau ci-dessous. La valeur en abscisse est h , tandis que la valeur en ordonnée est q . On considère que h et q sont des grandeurs adimensionnées.

	0 (sec)	1 (tempéré)	2 (humide)
0 (froid)	désert froid	tundra	taïga
1 (tempéré)	steppe	plaine	forêt
2 (chaud)	désert chaud	savane	jungle

On pose \mathcal{B} l'ensemble des biomes.

On pose $B : \mathcal{C} \rightarrow \mathcal{B}$ la fonction qui à tout chunk c associe son biome.

On pose $\mathcal{I} : \mathcal{C} \rightarrow \mathbb{N}$ l'indice du biome du chunk défini par

$$\forall c \in \mathcal{C}, \mathcal{I}(c) = h(c) + 3q(c)$$

1.3 Altitude

L'altitude d'un chunk c est notée $z(c)$ et est générée à partir d'un bruit de Perlin de façon similaire aux biomes.

On pose $z_r(c)$ l'altitude relative du chunk c , ç-à-d la différence de l'altitude du chunk et de la moyenne des altitudes des $n_v(c)$ chunks adjacents telle que:

$$z_r(c) = \frac{1}{n_v(c)} \sum_{c_a \in \mathcal{V}(c)} z(c_a) - z_r(c)$$

On pose aussi $\overline{z_r}$ l'altitude relative moyenne de la carte et $\sigma(z_r)$ l'écart-type de l'altitude relative de la carte.

$$\bar{z}_r = \frac{1}{N^2} \sum_{0 \leq i, j < N} z_r(c_{i,j}) \quad \text{et} \quad \sigma(z_r) = \sqrt{\frac{1}{N^2} \left(\sum_{0 \leq i, j < N} (z_r(c_{i,j}))^2 \right) - \bar{z}_r^2}$$

A partir d'une certaine altitude la case sera considéré comme inutilisable et offre un bonus d'humidité au case adjacents. C'est case sont appelés rivières.

1.4 Coefficient d'hostilité environnementale

Pour tout chunk c , on note $\mathcal{H}(c)$ le coefficient d'hostilité environnementale de c définie selon $h(c)$ et $q(c)$, respectivement l'humidité et la chaleur du chunk c tel que

$$\mathcal{H}(c) = (|h(c) - 1| + 1)(|q(c) - 1| + 1)$$

Remarque: On a $0 \leq h, q \leq 2$, donc $\mathcal{H} \in \{1, 2, 4\}$

1.5 Coefficient d'accessibilité

Le coefficient d'accessibilité ou accessibilité d'un chunk c est noté $\mathcal{A}(c)$ et est calculée à partir de la formule suivante

$$\mathcal{A}(c) = \frac{\sigma(z_r)}{\mathcal{H}(c)|\bar{z}_r - z_r(c)|}$$

Remarque: $\mathcal{A} \in \overline{\mathbb{R}^+}$, en effet si $\bar{z}_r = z_r(c)$ alors $\mathcal{A} = +\infty$

2 Villages

2.1 Définition

Au début de la simulation, l'algorithme place des villages tout les d_0 chunks. Un village qui possède au moins un bâtiment dans une des cases d'un chunk possède celui-ci.

Chaque village possède des ressources qu'elle peut stocker indéfiniment sans limite de quantité. Chaque village est possède également une population r_0 qui varie.

Les informations propres aux villages sont stockés dans le vecteur d'état $r(r_0, \dots, r_{n_r})$ avec n_r le nombre de ressources stockés, pour tout $1 \leq i \leq n_r$, r_i la quantité de la i -ème ressource stockée et r_0 la population du village.

2.2 Décisions

Chaque village peut effectuer n_α actions de façon autonome, rassemblés dans une famille $(\alpha_i) \in \mathbb{R}^n$.

Si la décision α est prise, alors on note $P(\alpha)$, sinon on note $\overline{P(\alpha)}$

Pour $n_\delta \in \mathbb{N}$, on pose $\delta(\delta_0, \dots, \delta_{n_\delta}) \in \mathbb{R}^{n_\delta+1}$ le vecteur décision.

$$\forall i < n_\delta, \delta_i \geq 1 \iff P(\alpha_i)$$

2.3 Chunks exploitables

Un chunk exploitable est soit un chunk contenant au moins un bâtiment du village ou bien adjacents à un chunk vérifiant cette propriété.

Soit n_ϵ le nombre de chunks exploitables par village.

On pose $(\epsilon_i) \in \mathbb{R}^{n_\epsilon}$ la famille des chunks adjacentes à tous les bâtiments du village.

On pose $(\beta_i)_{i \leq 8} \in \mathbb{R}^9$ tel que

$$\forall i \leq 8, \begin{cases} \beta_i = 1 & \text{si } \exists j < n_\epsilon, i = \mathcal{I}(\epsilon_j) \\ \beta_i = 0 & \text{sinon} \end{cases}$$

Si un des chunks exploitables est un chunk appartenant au i -ème biome, alors $\beta_i = 1$, sinon $\beta_i = 0$

2.4 Code génétique

2.4.1 Définition

On appelle *code génétique* d'un village l'ensemble des matrices qu'il utilise pour prendre des décisions de façon autonome.

2.4.2 Matrice de décision

Pour pouvoir évaluer les coordonnées du vecteur décision Δ , on définit $D \in \mathcal{M}_{n_\delta, n_\alpha}$. D est généré de façon aléatoire en début de simulation.

Ces grandeurs sont reliés par la relation:

$$r \times D = \Delta$$

D'où

$$\begin{pmatrix} r_1 \\ \vdots \\ r_{n_\alpha} \end{pmatrix} \begin{pmatrix} D_{0,0} & \dots & D_{0,n_\alpha} \\ \vdots & \ddots & \vdots \\ D_{n_\delta,0} & \dots & D_{n_\delta,n_\alpha} \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \vdots \\ \delta_{n_\delta} \end{pmatrix}$$

2.5 Bâtiments

Un village est constitué de bâtiments. Chaque bâtiment nécessite un certain nombre d'habitants pour son fonctionnement. Les habitants ont besoin lors de chaque tour pour survivre d'une ressource logement et d'une ressource nourriture. Un bâtiment est détruit s'il n'y a pas assez d'habitants pour le maintenir. Certains bâtiments ont aussi besoin de matières premières pour produire les ressources associés lors du tour.

2.5.1 Mode d'apparition

Au début de la simulation, l'algorithme muni chaque village de 4 bâtiments. Le village est également muni d'actions de construction gérés par la matrice de décisions.

Le coût de construction net $\mathcal{C}(c)$ est réévalué par rapport au coût de construction brut par le coefficient d'accessibilité du chunk. Avec \mathcal{C}_b le coût de construction brut du bâtiment,

$$\mathcal{C}(c) = \mathcal{A}(c) \times \mathcal{C}_b$$

2.5.2 Types de bâtiments

- **Maison:** fait apparaître 20 habitants dans le village.
- **Champ:** fait apparaître 20 grain dans le village, emploie 8 personnes. Le grain doit être transmis au moulin pour être transformé en nourriture.
- **Moulin:** fait apparaître 20 nourriture dans le village, emploie 4 personnes. Une nourriture permet de faire survivre une personne pour un tour.
- **Mine:** fait apparaître 20 minerais dans le village, emploie 8 personnes. Le minerai doit être transmis à une usine pour être transformée en machines.
- **Usine:** booste la production des bâtiments du village par 1.1, utilise 8 minerais. Emploie 4 personnes.

2.6 Tableau récapitulatif

	production	emploi	matières premières	coût de construction brut
Maison	20 logement	0	-	20 grain
Champ	20 grain	8	-	40 nourriture
Moulin	20 nourriture	4	10 grain	60 grain
Mine	20 minerais	8	-	40 minerais
Usine	$\times 1.1$	4	20 minerais	80 minerais

Disclaimer: Les valeurs sont indicatives et varient selon les environnements