

Jeu Factory et le coding

Christophe Brun <christophe@cbrun.fr>

2024



2026-07-21

Jeu Factory et le coding

Jeu Factory et le coding
Christophe Brun <christophe@cbrun.fr>
2024

- Les jeux d'usines
- Les règles de bases
- La 1ère partie
- les suivantes
- Optimisations, refactor,
- La fin du jeu



2026-07-21

Jeu Factory et le coding

└─ Sommaire

Sommaire

- Les jeux d'usines
- Les règles de bases
- La 1ère partie
- les suivantes
- Optimisations, refactor,
- La fin du jeu

Les jeux de constructions



2026-07-21

Jeu Factory et le coding

└ Les jeux de constructions

Les jeux de constructions



- Jeu 2D
- 2012 : Lancement du projet 1 dev puis 6 personnes
- Campagnes participatives
- 2020 : Sortie
- Extensible, code en Lua
- Programmation Logique in Game



2026-07-21

Jeu Factory et le coding

└ Factorio

Factorio



- Jeu 2D
- 2012 : Lancement du projet 1 dev puis 6 personnes
- Campagnes participatives
- 2020 : Sortie
- Extensible, code en Lua
- Programmation Logique in Game



- Paradox Interactive
- Encore en version Alpha
- Particularité : Map Générative, proche du système de Minecraft
- Rust



2026-07-21

Jeu Factory et le coding

Foundery

Foundery



- Paradox Interactive
- Encore en version Alpha
- Particularité : Map Générative, proche du système de Minecraft
- Rust

ShapeZ 2



- principe de construire des forme par assemblage pour envoyer dans un Vortex
- fait travailler l'optimisation mais aussi les stratégies
- Graphismes simples mais la progression est hyper motivante
- Javascript



2026-07-21

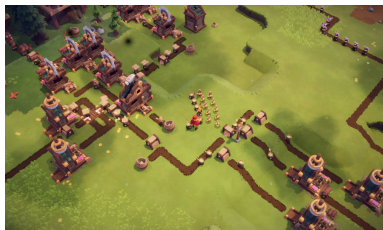
Jeu Factory et le coding

└ ShapeZ 2

ShapeZ 2



- principe de construire des forme par assemblage pour envoyer dans un Vortex
- fait travailler l'optimisation mais aussi les stratégies
- Graphismes simples mais la progression est hyper motivante
- Javascript



- Orienté plus sur la collaboration des process (Sparks)
- Histoire sympa, pas mal de biomes
- Construire ses sparks en plus des ressources
- un peu de combat
- Mignon
- Attention aux comportements non voulus (Bug)
- Golang

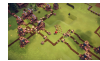


2026-07-21

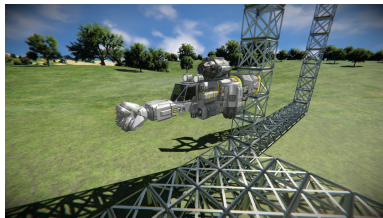
Jeu Factory et le coding

└ OddSparks

OddSparks



- Orienté plus sur la collaboration des process (Sparks)
- Histoire sympa, pas mal de biomes
- Construire ses sparks en plus des ressources
- un peu de combat
- Mignon
- Attention aux comportements non voulus (Bug)
- Golang



- C'est le C++ des jeux d'Usine
- on a que des modules, faut tout construire
- Physique réaliste
- Bugs de construction peuvent être fatal
- Demande un temps de montée en compétence
- Logique de programmation in Game



2026-07-21

Jeu Factory et le coding

└ Space Engineers

Space Engineers



- C'est le C++ des jeux d'Usine
- on a que des modules, faut tout construire
- Physique réaliste
- Bugs de construction peuvent être fatal
- Demande un temps de montée en compétence
- Logique de programmation in Game

Satisfactory



- Sortie de la 1.0 ce mois
- beau, bien codé, et surtout offre pour une seule map des biomes qui changent la stratégie
- Histoire, découverte ou optimisation
- Python



2026-07-21

Jeu Factory et le coding

└ Satisfactory

Satisfactory



- Sortie de la 1.0 ce mois
- beau, bien codé, et surtout offre pour une seule map des biomes qui changent la stratégie
- Histoire, découverte ou optimisation
- Python



2026-07-21

Jeu Factory et le coding

└ Règles de Base

Règles de Base

Les ressources



- Elles sont limitées
- Il faut les découvrir
- Plus ou moins facile à trouver
- Certaines sont rares
- Plus on avance et plus il faut de ressources



2026-07-21

Jeu Factory et le coding

└ Les ressources

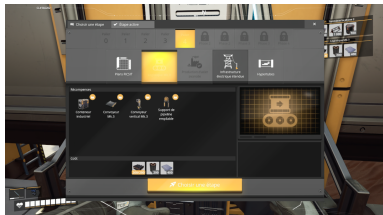
1. Idem ressources machines
2. Vitesse ou pas
3. Place : mémoire
4. IO : tapis, transports, ...

Les ressources



- Elles sont limitées
- Il faut les découvrir
- Plus ou moins facile à trouver
- Certaines sont rares
- Plus on avance et plus il faut de ressources

Les connaissances



- elles augmentent au fur et à mesure
- certaines ne sont pas accessibles
- Parfois une documentation
- parfois il faut découvrir (poc)



2026-07-21

Jeu Factory et le coding

└ Les connaissances

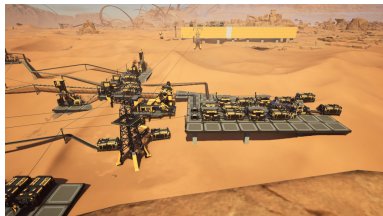
1. Comme les connaissances pour un langage
2. idem doc, parfois pas
3. faut tester avant de se lancer
4. Frustration quand on ne comprend pas

Les connaissances



- elles augmentent au fur et à mesure
- certaines ne sont pas accessibles
- Parfois une documentation
- parfois il faut découvrir (poc)

Les optimisations



- Des améliorations des machines
- Des recettes alternatives
- Des automatismes



2026-07-21

Jeu Factory et le coding

└ Les optimisations

1. Librairies
2. Add-ons

Les optimisations



- Des améliorations des machines
- Des recettes alternatives
- Des automatismes



La Première Partie

2026-07-21

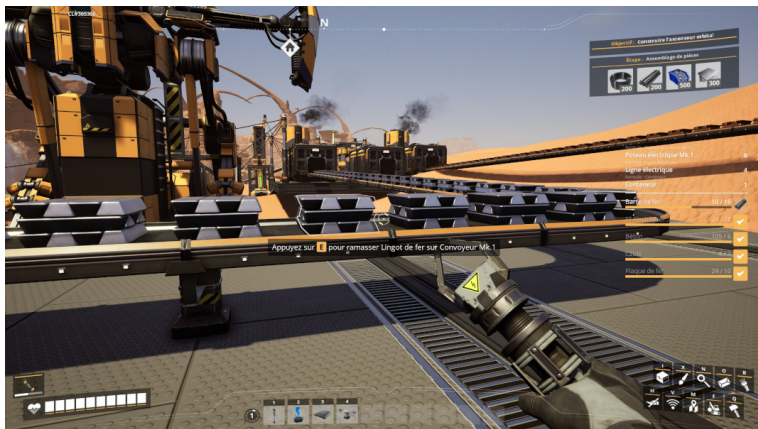
Jeu Factory et le coding

└─ La Première Partie

La Première Partie



└ Découverte



1. Prendre le temps de regarder
2. de se documenter
3. tester, explorer en dehors d'une partie ou d'un vrai code



ça fonctionne mais ...



Jeu Factory et le coding

2026-07-21

└ ça fonctionne mais ...

ça fonctionne mais ...



1. C'est une solution qui m'a permis de finir le jeu en early
2. Au final, de la perte de temps en trajet ou chose manuelle
3. code pas optimisé mais permis de comprendre les interactions

C'est moche



2026-07-21

Jeu Factory et le coding

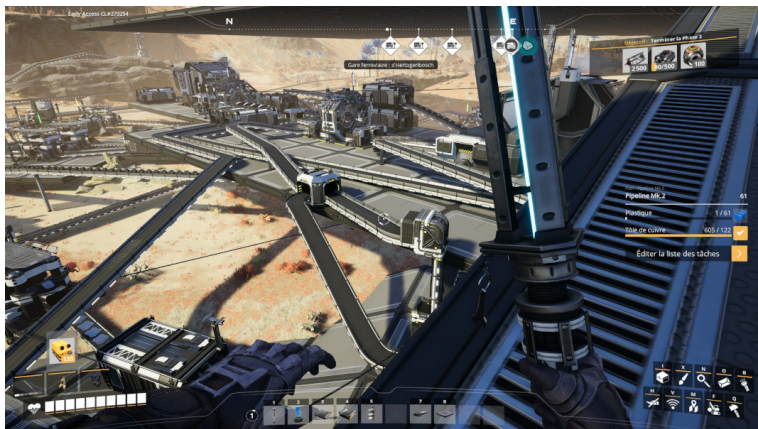
└ C'est moche

C'est moche



1. C'est pas l'opéra de Sydney
2. c'est pas réaliste (planter dans le vide)
3. Et à part si on veut taper le challenge spaghetti, c'est bof

Evolution complexe



2026-07-21

Jeu Factory et le coding

└ Evolution complexe

1. Compliqué, même si on empile, le passage des tapis devient difficile
2. Ajout de ligne a des effets directs sur les autres
3. Baisse de perf sur certaines lignes

Evolution complexe





Les Suivantes

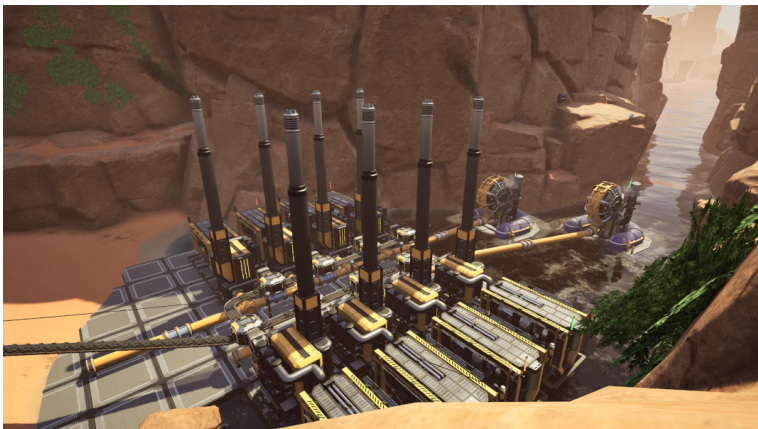


2026-07-21

Jeu Factory et le coding

└─ Les Suivantes

Les Suivantes



2026-07-21

Jeu Factory et le coding

└ Organisé

Organisé



1. On commence à dédier certains espaces à des fonctions
2. Là, c'est de la production d'énergie
3. Contraint par l'environnement : EAU
4. Comme pour les fonctions, je n'ai que les entrées nécessaires et sortie

Des patterns



2026-07-21

Jeu Factory et le coding

└ Des patterns

Des patterns



1. Suite aux parties précédentes, on commence à avoir des automatismes : pattern de fabrication
2. On connaît les contraintes et les limites de certaines lignes
3. Sans optimiser, on produit plus efficacement
4. Début d'architecture



2026-07-21

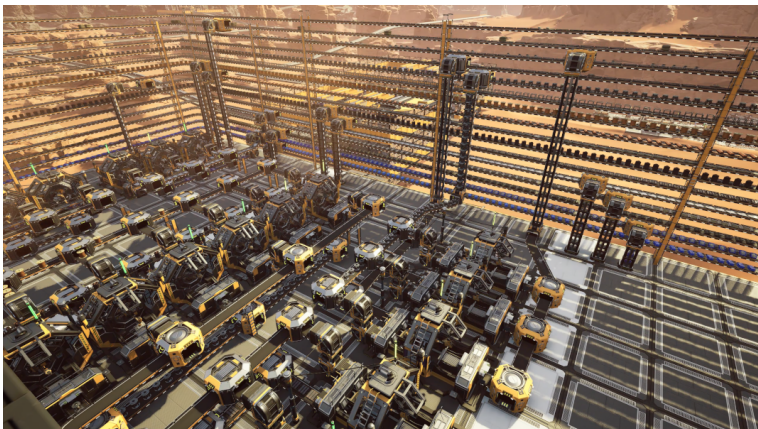
Jeu Factory et le coding

└─ Modulaire

Modulaire



1. Approche modulaire, mais comme pour le code, on découpe la complexité
2. Starter en bas, permet de produire les 1ère briques
3. Pour aller ensuite vers le fond



2026-07-21

Jeu Factory et le coding

└─ Evolutif

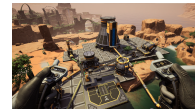
Evolatif



1. Les lignes sont indépendantes
2. On peut ajouter sur la surface
3. L'évolution se fait verticale dans cette architecture
4. Architecture de Main Bus



└ Encore des POC



1. Finir le jeu dans les 1ère alpha : pas de nucléaire
2. pour comprendre le fonctionnement d'une nouveauté du langage
3. Un poc pour tester, donc pas d'architecture
4. On essaye de comprendre les contraintes, dépendances, et les besoins de la solutions
5. On test

Optimisations, refactor, ...



2026-07-21

Jeu Factory et le coding

└─ Optimisations, refactor, ...

Optimisations, refactor, ...

Load Balancer



2026-07-21

Jeu Factory et le coding

└─ Load Balancer

Load Balancer



1. Avantages, inconvénients
2. Pas tout à fait comme une architecture réseaux, mais le principe est là
3. Bien pour débuter car c'est logique dans la compréhension du jeu
4. ça peut même en opti être une solution efficace sur la fin du jeu : donc à ne pas négliger

- Nécessite de calculer en amont pour le rendement
- Evolution difficile
- Démarrage maîtrisé
- Optimisation des ressources
- Gestion de la surface compliquée



2026-07-21

Jeu Factory et le coding

└─ Load Balancer

Load Balancer

- Nécessite de calculer en amont pour le rendement
- Evolution difficile
- Démarrage maîtrisé
- Optimisation des ressources
- Gestion de la surface compliquée

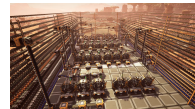


2026-07-21

Jeu Factory et le coding

└─ Manifold

Manifold



1. Maitrisable, si on comprend ce qu'on fait
2. Risques quand même parfois d'essoufflement (pas effondrement, mais temps allongés)
3. efficace en terme d'avancement dans les level de connaissances, car on augmente les lignes en fonction des besoins
4. Peu inter blocage

- Surface minimale
- Evolution complète, il suffit d'ajouter des tranches
- Démarrage progressif
- Mise en puissance longue
- Pas de réflexion en amont



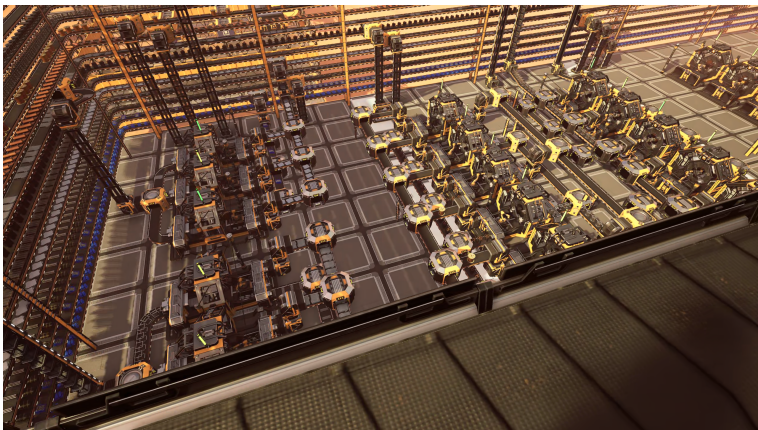
2026-07-21

Jeu Factory et le coding

└─ Manifold

Manifold

- Surface minimale
- Evolution complète, il suffit d'ajouter des tranches
- Démarrage progressif
- Mise en puissance longue
- Pas de réflexion en amont

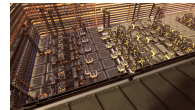


2026-07-21

Jeu Factory et le coding

└ Main Bus

Main Bus



1. C'est le mot à la mode dans le jeu
2. Pas forcément nécessaire depuis la 1.0 : souvent augmenté avec un Stockage centralisé
3. Plusieurs type : linéaire, Etoiles, circulaire
4. Attention, à l'engorgement
5. Pas simple au début
6. Évolution sur les 4 axes : surface (2 axes), Hauteur (1 axe) et temps (1 axe)

- Fun à faire, mais après des heures de jeu car on a overpack, sinon faut construire des rampes
- Besoins en ressources énorme et temps de fabrication
- Plus facile à maintenir
- Main bus simple vs Main Bus en boucle
- Risque d'effondrement



2026-07-21

Jeu Factory et le coding

└ Main Bus

Main Bus

- Fun à faire, mais après des heures de jeu car on a overpack, sinon faut construire des rampes
- Besoins en ressources énorme et temps de fabrication
- Plus facile à maintenir
- Main bus simple vs Main Bus en boucle
- Risque d'effondrement



La fin du jeu



L'histoire (pas de spoil)



2026-07-21

Jeu Factory et le coding

└ L'histoire (pas de spoil)

1. En théorie c'est l'objectif
2. Mais pas nécessaire
3. On n'avait pas d'histoire jusqu'à la V1
4. Reste un bac à sable

L'histoire (pas de spoil)

NO SPOILERS

L'optimisation



- Calculateur
- Des fous d'optimisation
- Mode speedrunner : challenge d'arriver à un level donné en un minimum de temps
 - un peu nouveau sur satisfactory : 27mn pour le 1er elevateur , c'est level 0 , 1 et 2
 - Factorio : 1h27 pour le 100



2026-07-21

Jeu Factory et le coding

└ L'optimisation

L'optimisation



- Calculateur
- Des fous d'optimisation
- Mode speedrunner : challenge d'arriver à un level donné en un minimum de temps
 - un peu nouveau sur satisfactory : 27mn pour le 1er elevateur , c'est level 0 , 1 et 2
 - Factorio : 1h27 pour le 100

1. Maitrisable, si on comprend ce qu'on fait
2. Risques quand même parfois d'essoufflement (pas effondrement, mais temps allongés)
3. efficace en terme d'avancement dans les level de connaissances, car on augmente les lignes en fonction des besoins
4. Peu inter blocage



2026-07-21

Jeu Factory et le coding

└ Design

Design



1. certains recherche à pousser les limites de la construction
2. Sydney
3. photo sur la communauté



2026-07-21

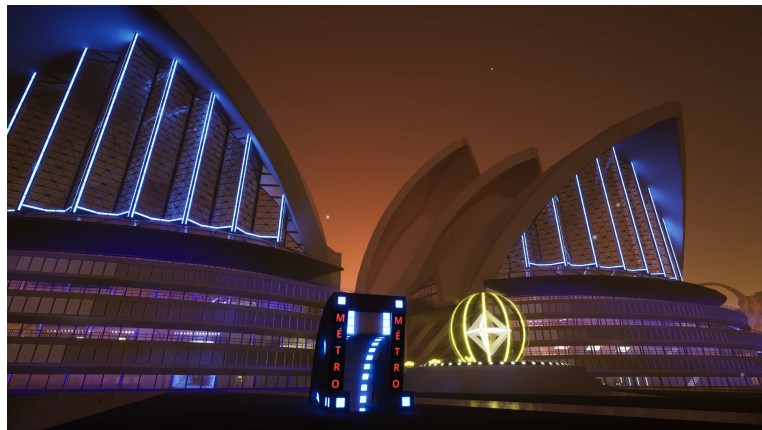
Jeu Factory et le coding

└ Next Level

Next Level



1. Voilà un exemple : +600h pour l'opéra, mais la game à plus de 1500h et ça fait plus d'un an qu'il est dessus
2. D'autres challenges existes, par exemple, essayer d'aller le plus haut
3. Bloquer le système en construisant sur la map
4. atteindre des max de production

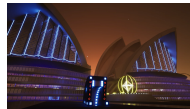


2026-07-21

Jeu Factory et le coding

└ Next Level

Next Level



Conclusion

2026-07-21

└ Conclusion

1. C'est pareil que coder non ?
2. ça marche dans un sens, mais l'autre aussi
3. le tout est de connaître les objectifs,
4. Avancer pas à pas
5. Améliorer nos connaissances
6. Faire un peu attention aux ressources à l'environnement (sécurité, résilience, effets de bord)
7. Chacun y trouve son compte
8. Il n'y pas pas une solution meilleure que l'autre (pour info, j'ai atteint sur 2 games le même level en temps de sauvegarde identique en mode organisé et en mode sphagetti)

