


-  Guide Pas-à-Pas : Créer un Modèle Kanban dans FlexSim
  - Phase 1 : Créer la structure de base (Flux produits)
    - Étape 1.1 : Nouveau modèle
    - Étape 1.2 : Placer les objets (de gauche à droite)
    - Étape 1.3 : Connecter les objets
    - Étape 1.4 : Configurer les objets
    - Étape 1.5 : Premier test
  - Phase 2 : Ajouter les Plannings Kanban
    - Étape 2.1 : Placer les plannings visuels
    - Étape 2.2 : Configurer l'affichage des cartes
  - Phase 3 : Logique Kanban (Scripts)
    - Étape 3.1 : Créer un Global Table pour les cartes
    - Étape 3.2 : Script OnReset (Initialiser les cartes)
    - Étape 3.3 : Logique de production (Flux tiré)
    - Étape 3.4 : Libération des cartes (Client)
  - Phase 4 : Test et Validation
    - Checklist de test
  - Prochaines étapes

## Guide Pas-à-Pas : Créer un Modèle Kanban dans FlexSim

---

### Phase 1 : Créer la structure de base (Flux produits)

---

#### Étape 1.1 : Nouveau modèle

1. Ouvrir FlexSim
  2. **File** → **New Model**
  3. Sauvegarder : **Kanban\_Simple.fsm**
-

## Étape 1.2 : Placer les objets (de gauche à droite)

Glissez depuis la **Library** les objets suivants :

#	Objet à placer	Nom à donner	Position X
1	Source	StockMatiere	0
2	Queue	Buffer0	3
3	Processor	Poste1	6
4	Queue	Buffer1	9
5	Processor	Poste2	12
6	Queue	Buffer2	15
7	Processor	Poste3	18
8	Queue	StockPF	21
9	Sink	Client	24

**Astuce** : Double-clic sur chaque objet → onglet General → changer le nom

## Étape 1.3 : Connecter les objets

1. Touche **A** pour activer le mode connexion
2. Cliquez sur **StockMatiere** puis sur **Buffer0**
3. Continuez : **Buffer0** → **Poste1** → **Buffer1** → **Poste2** → **Buffer2** → **Poste3** → **StockPF** → **Client**

Vous devriez avoir une ligne continue de gauche à droite.

## Étape 1.4 : Configurer les objets

**StockMatiere (Source) :**

- Inter-Arrival Time : 0 (génère tout au démarrage)

- Arrival Quantity : 20
- Arrival Schedule → Stop After : 1 arrival

#### Poste1 (Processor) :

- Process Time : 60 secondes

#### Poste2 (Processor) :

- Process Time : 55 secondes

#### Poste3 (Processor) :

- Process Time : 50 secondes

#### Client (Sink) :

- *(pas de config spéciale)*
- 

## Étape 1.5 : Premier test

1. Cliquer sur **Reset** puis **Run**
  2. Vérifier que les pièces circulent de gauche à droite
  3. **Stop** quand validé
- 

## Phase 2 : Ajouter les Plannings Kanban

---

### Étape 2.1 : Placer les plannings visuels

Ajoutez 3 **Queues** en dessous de la ligne principale :

Nom	Position X	Position Y
PlanningK1	6	-3
PlanningK2	12	-3
PlanningK3	18	-3

---

# Étape 2.2 : Configurer l'affichage des cartes

Pour chaque PlanningK :

1. Double-clic → onglet **Visuals**
2. Cocher **Draw Content As Shapes**
3. Shape : **Box**
4. Scale X/Y/Z : 0.3 / 0.3 / 0.05

Couleurs distinctives :

- PlanningK1 : Bleu
- PlanningK2 : Jaune
- PlanningK3 : Rouge

---

## Phase 3 : Logique Kanban (Scripts)

### Étape 3.1 : Créer un Global Table pour les cartes

1. Menu **Tools** → **Global Tables**
2. Créer : **CarteKanban**
3. Colonnes : **ID**, **Poste**, **EnCirculation**

*(Optionnel - on peut aussi gérer via labels)*

---

### Étape 3.2 : Script OnReset (Initialiser les cartes)

1. **Tools** → **Model Parameters** ou clic droit sur le modèle
2. Ajouter un trigger **OnReset**

```
// Créer 5 cartes Kanban par poste au démarrage
for (int poste = 1; poste <= 3; poste++) {
    Object planning;
```

```

if (poste == 1) planning = Model.find("PlanningK1");
if (poste == 2) planning = Model.find("PlanningK2");
if (poste == 3) planning = Model.find("PlanningK3");

for (int i = 1; i <= 5; i++) {
    Object carte = createinstance(getclass("Box"));
    carte.setLabel("Poste", poste);
    carte.setLabel("NumCarte", i);
    carte.setSize(0.3, 0.3, 0.05);
    if (poste == 1) carte.setColor(colorblue);
    if (poste == 2) carte.setColor(coloryellow);
    if (poste == 3) carte.setColor(colorred);
    moveobject(carte, planning);
}
}

```

---

## Étape 3.3 : Logique de production (Flux tiré)

### Poste3 - Trigger "OnEntry" :

```

Object item = param(1);
Object current = ownerobject(c);
Object planning = Model.find("PlanningK3");

// Vérifier si une carte Kanban est disponible
if (getcontentsize(planning) == 0) {
    // Pas de carte = pas de production, renvoyer la pièce au buffer
    return 0; // Bloquer l'entrée
}

// Prendre une carte du planning
Object carte = last(planning);
moveobject(carte, current); // Attacher la carte à la pièce

```

### Poste3 - Trigger "OnExit" :

```

Object item = param(1);

// La carte accompagne le produit vers le stock PF
// Elle sera libérée quand le client consomme

```

---

## Étape 3.4 : Libération des cartes (Client)

## Client (Sink) - Trigger "OnEntry" :

```
Object item = param(1);

// Récupérer la carte attachée et la renvoyer au planning
int poste = item.getLabel("DernierPoste");
Object carte = createinstance(getclass("Box"));
carte.setLabel("Poste", 3);
carte.setSize(0.3, 0.3, 0.05);
carte.setColor(colorred);

Object planning = Model.find("PlanningK3");
moveobject(carte, planning);

// Signaler le besoin en amont (cascade Kanban)
// Envoyer une carte au PlanningK2
Object carte2 = createinstance(getclass("Box"));
carte2.setColor(coloryellow);
carte2.setSize(0.3, 0.3, 0.05);
moveobject(carte2, Model.find("PlanningK2"));
```

---

## Phase 4 : Test et Validation

### Checklist de test

- ☐ Les cartes apparaissent sur les plannings au Reset
- ☐ La production ne démarre que s'il y a des cartes
- ☐ Les cartes circulent avec les produits
- ☐ Les cartes reviennent au planning après consommation
- ☐ Le système s'auto-régule (pas de surproduction)

---

## Prochaines étapes

Une fois ce modèle de base fonctionnel, on pourra ajouter :

- ☐ Visualisation Dashboard avec graphiques
- ☐ Calcul du nombre optimal de cartes
- ☐ Simulation de pannes et variabilité
- ☐ Comparaison avec flux poussé

---

**Prêt à commencer ? Ouvrez FlexSim et suivez la Phase 1 !**