# Automatisation des scènes

## Introduction

Certaines parties du jeu doivent être automatisées. Autrement dit, dans ces parties, le joueur n’a qu’un accès d’interaction limité voire nul.

Il s’agit des scènes de cinématique (comme la scène d’introduction) ou de scène de dialogues (pour lesquelles le joueur peut seulement répondre parmi un panel de r »ponses possibles).

## Machine d’état

L’automatisation implique la mise en œuvre d’une machine d’état pendant le déroulement de laquelle, les PJs sont sous le contrôle du CPU.

La machine d’état doit avoir un point d’entrée (à partir duquel, les interactions du joueur sont réduites) et un point de sortie (à partir duquel le joueur retrouve son autonomie).

La machine d’état disposera des ressources suivantes :

* Fichiers de dialogue audio
* Chaîne de caractères de réponses
* Inventaire du personnage.

## Impact sur le backend des personnages

* Les personnages doivent hériter d’une interface IControlable contenant un flag booléen PullingStrings, qui, s’il est true, indique que le personnage n’est plus sous le contrôle du joueur.
* Toutes les actions des PJs doivent être appelables programmatiquement.

Conséquence : Une architecture séparant le code métier du front est indispensable. L’inversion de contrôle est à privilégier dans tout le code.

## Description fonctionnelle de la machine d’état

Chaque état de la machine représente une action (un déplacement, un dialogue, etc…)

Entre chaque état les transitions sont contraintes par une condition.

**Condition** : porte atteinte

**Etat** : Dialogue

**Action** : Jouer le fichier audio : « Cette porte à l’air fermée. »

Exemple : Lorsqu’un personnage est déplacé de manière automatique, il passe à l’état suivant sous réserve qu’il ait atteint son emplacement cible.

**Etat** : Marcher vers la porte

**Action** : Goto coordonnées de la porte.