1. Introduction

Constructing a narrative ecosystem is one of the important technologies for many computer games. This paper presents a reactive episode control system that allows for dynamic creation of a long-term story in response to the active interaction between multiple characters and the user’s actions. This system makes possible a story world where the user’s actions not only create reactions among the characters that are currently present, but also influence the long-term story events.

2. Distributed Episode Control

Episode control system stores a number of episode trees (components of the story) and dynamically connects them in a hierarchy, in parallel or in series to generate a variety of possible stories. This model organizes the flow of events into a hierarchy based on AND/OR conditions, creating many possibilities that could happen in a single episode. A single story is composed of a number of episodes organized in a hierarchy, in parallel or in series. Three types of episode tree are provided: narrative trees, reaction trees and schedule trees. At the end point of the tree is an event. Narrative trees contain story elements and create episodes in which many characters take part. Reaction trees create episodes based on reactions, such as returning greetings or chasing a butterfly. Schedule trees create episodes in line with daily routines when there are no other unique events.

3. Episode Coupling Algorithm

In this system, the story develops according to the spontaneous interactions of active characters and intervention by the user. There are two problems: one is that the current episode is ended in the mid-flow by the new episode, the other is that the situation between two episodes becomes discontinuous as the result of the former problem. The episode coupling algorithm resolves these problems by stacking events and by adding some supplementary processes preferentially. When a certain episode is suspended in mid-flow, this algorithm stacks new episode to the previous one and keeps it. At the same time, it inserts the post-process of the previous episode and pre-process of the new one, to connect episode continuously. The new episode is popped up after finishing, the stacked previous one resumes seamlessly because of process insertion.