Videogame Dialogue Systems through AI Behavior Composition

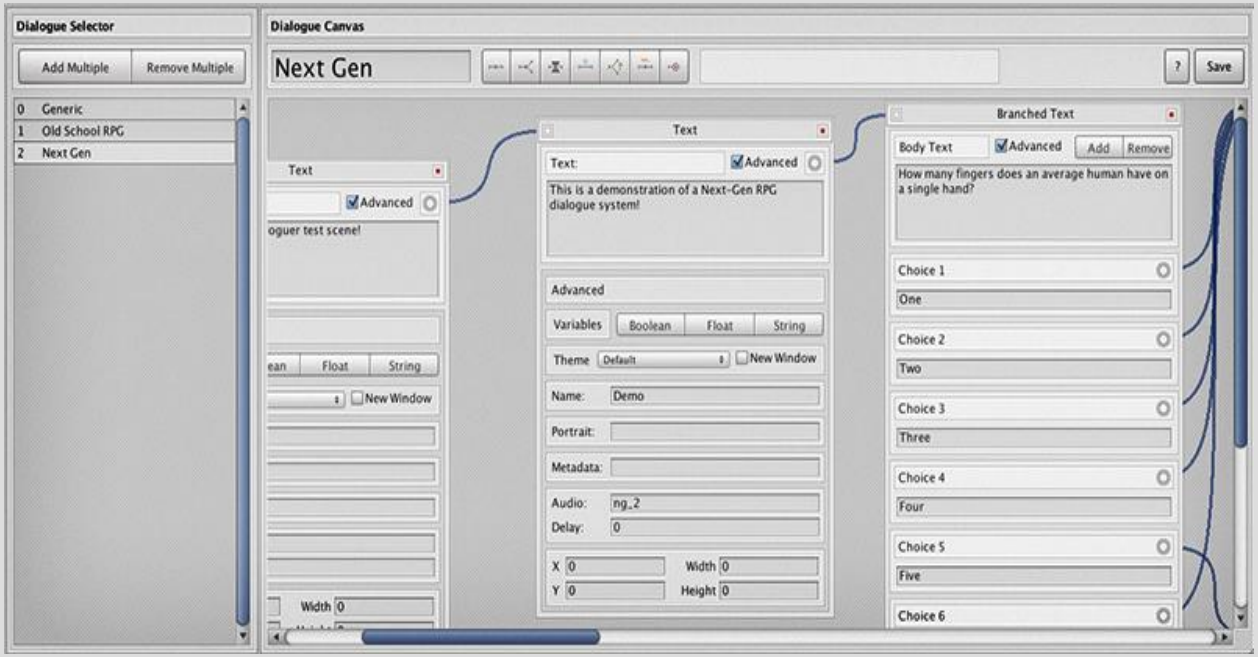
Daniele Riccardelli
Joint work with: Giuseppe De Giacomo, Stavros Vassos

SAPIENZA
UNIVERSITÀ DI ROMA

Dialogue Systems in Videogames 1.0

Typical case in videogames

- Dialogue storyline is represented as hardcoded conversation graphs with hardcoded dialogue lines
- Example: a node-based dialogue creator and editor for Unity (www.dialoguer.info)



SAPIENZA UNIVERSITÀ DI ROMA

Dialogue Systems in Videogames 2.0

Motivation for our work

- AI-powered flexibility and adaptiveness
- Better tools for developing, maintaining and debugging

How

- Applying and extending “Behavior Composition”, a well-studied technique from AI
- Appealing to the popular Unity game engine as target platform for our prototype tools

Dialogue Systems in Videogames 2.0 Behavior Composition

Input

- A repository of questions and answers
- The role of each character in the game expressed as a transition system based on question-answer pairs to switch states
- A target intended storyline expressed also as a transition system of possible question-answer interactions

Output

- A computed strategy expressed as a lookup-table that says which character should engage with a possible question-answer interaction

Dialogue Systems in Videogames 2.0
Behavior Composition

Input

- A **repository** of **questions** and **answers**

  q1: ***Hey there, what's going on?***
  a1: *A terrible thing just happened here! Go inside and investigate!*
  a2: *Are you still here?!? Run inside! The criminal might well still be around!*

  q2: ***You, little kid: do you know anything about this crime?***
  a1: *I could tell you if you could give me something in return first.*

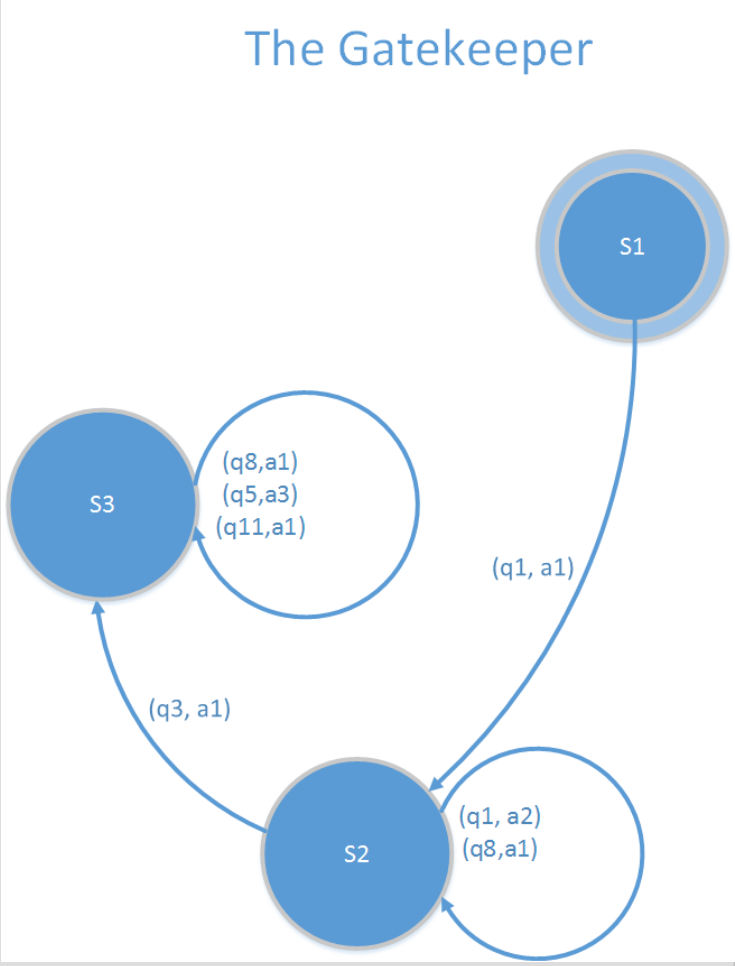
  q3: ***The little kid is looking for something he's lost. Do you know what that is?***
  a1: *Oh, I guess I do! I found this in the yard, this morning.*
  a2: *Hmm, I have no idea. I barely see that kid around.*

  …

Dialogue Systems in Videogames 2.0 Behavior Composition

Input

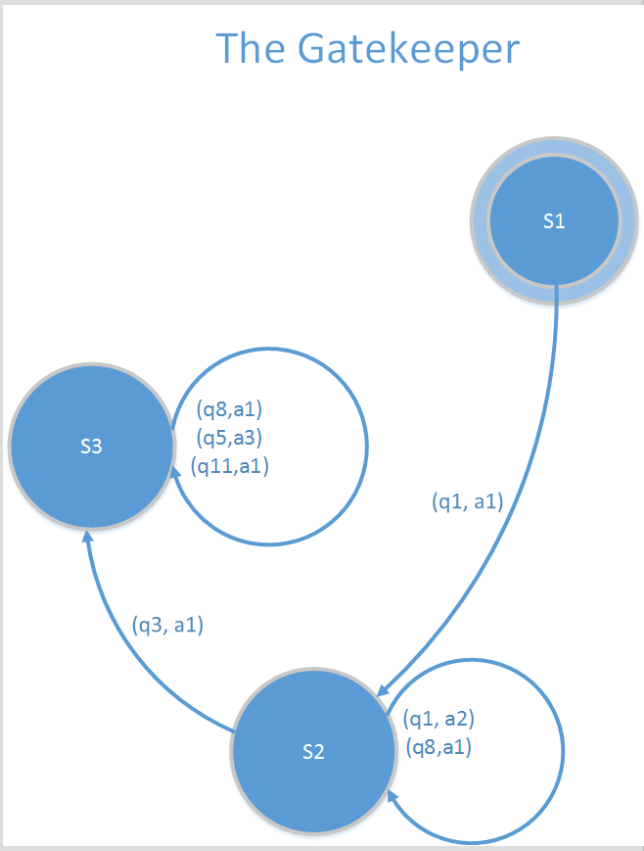
- The **role of each character** in the game expressed as a **transition system** based on **question-answer pairs** to switch states
- E.g., **(q1,a1)** is an interaction that may introduce the player to the story:
  q1: *Hey there, what's going on?*
  a1: *A terrible thing just happened here! Go inside and investigate!*
- This transitions the role of the character to **state S2** where a different answer will be given to **q1** after the first time: **(q1,a2)**

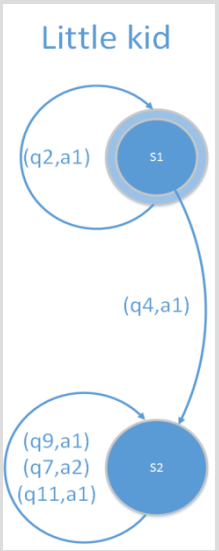


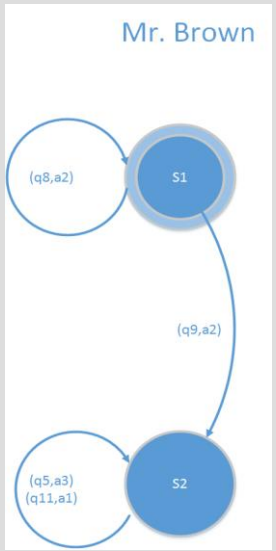
Dialogue Systems in Videogames 2.0 Behavior Composition

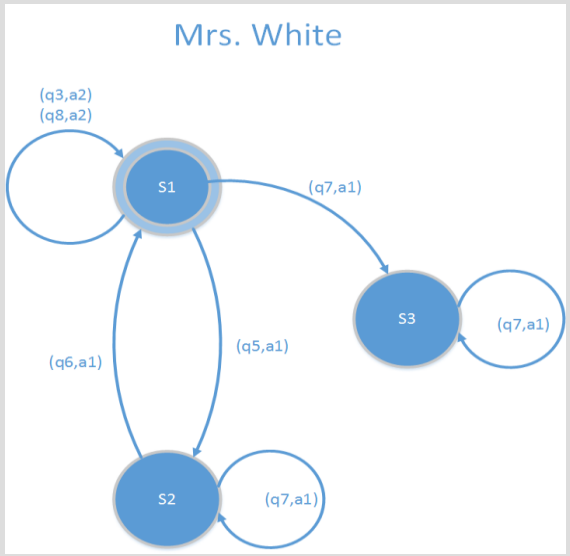
Input

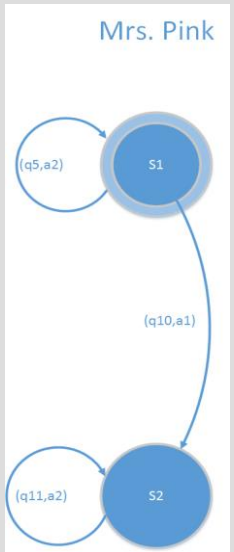
- The **role of each character** in the game expressed as a **transition system** based on **question-answer pairs** to switch states
- This provides also a way to model character mood, memory of past events, etc.







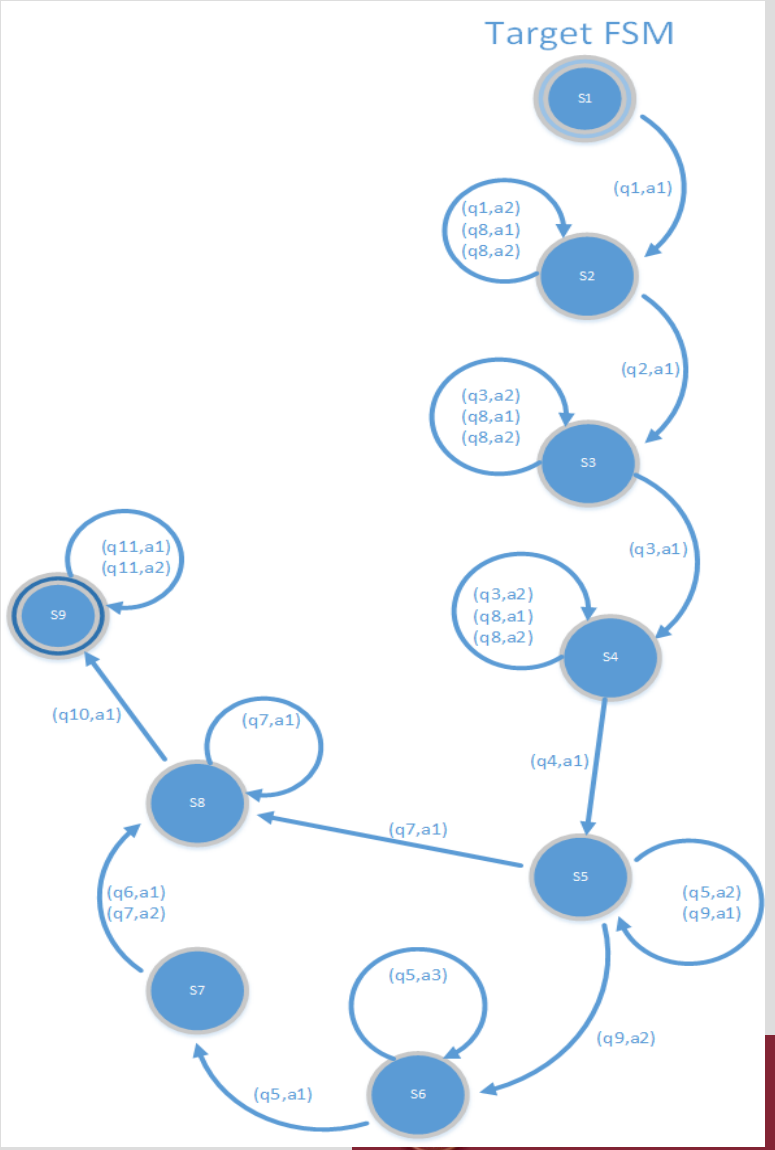




Dialogue Systems in Videogames 2.0 Behavior Composition

Input

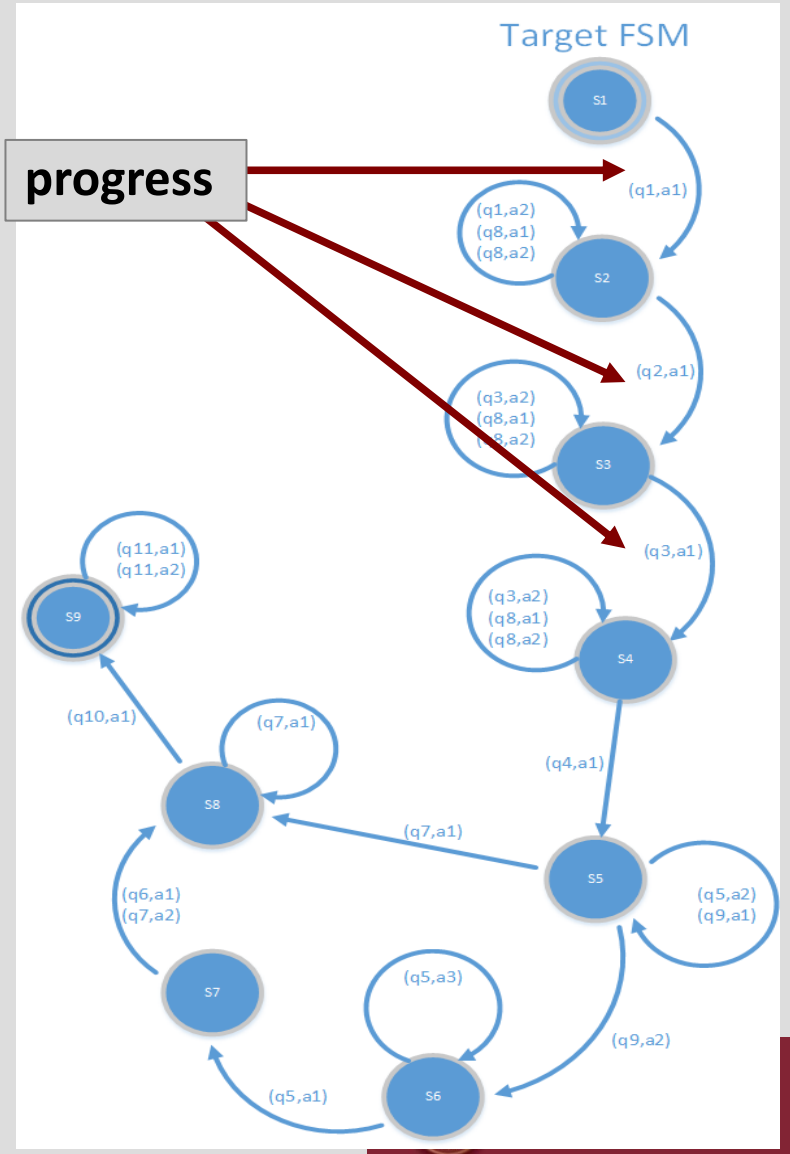
- A target **intended storyline** expressed also as a **transition system** of possible **question-answer** interactions
- Dialogue storyline specification has **loops**



Dialogue Systems in Videogames 2.0 Behavior Composition

Input

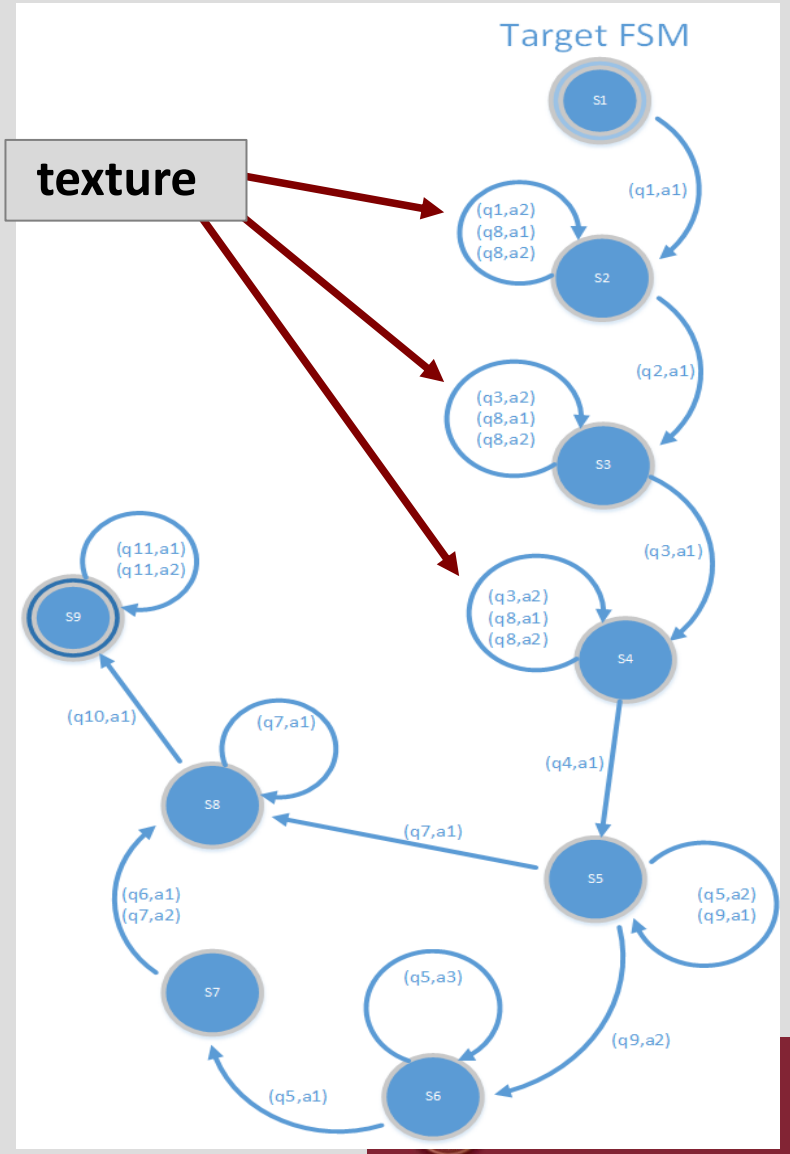
- A target **intended storyline** expressed also as a **transition system** of possible **question-answer** interactions
- Dialogue storyline specification has **loops**
- Some **(q,a)** pairs **“progress”** the story forward from **S1** to **S2,** from **S2** to **S3**, etc.



Dialogue Systems in Videogames 2.0 Behavior Composition

Input

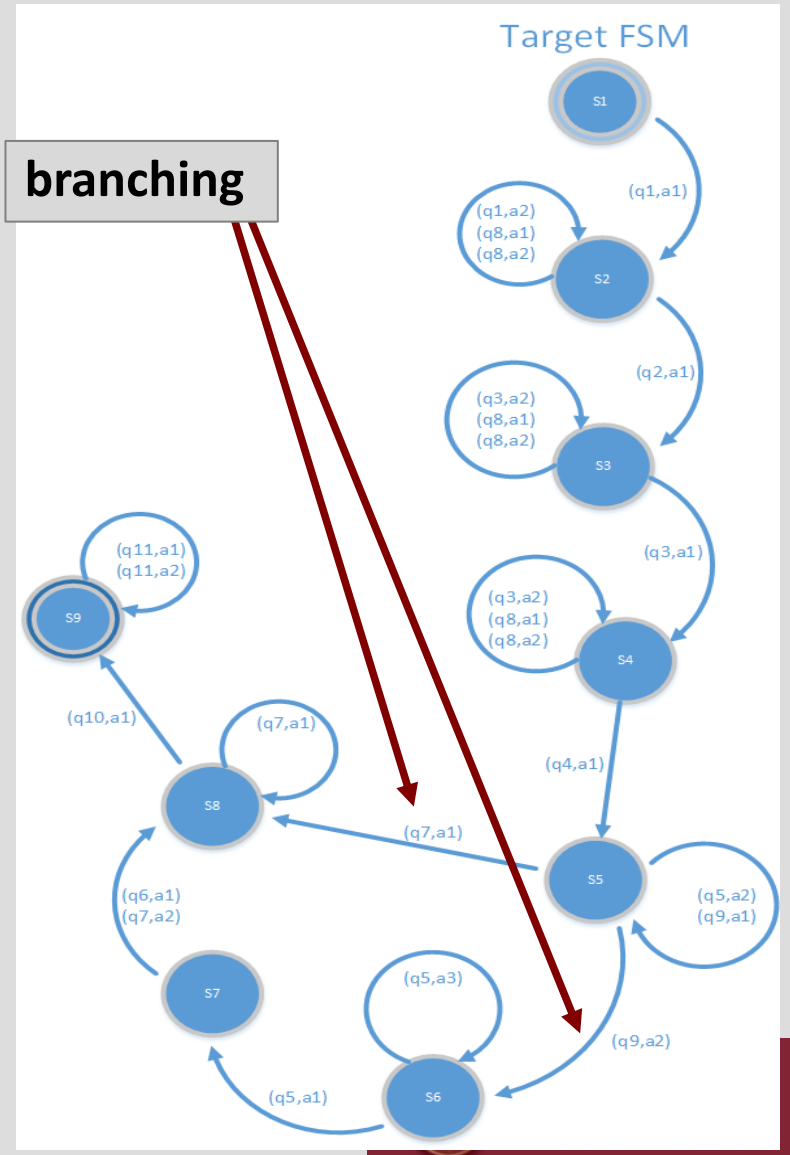
- A target **intended storyline** expressed also as a **transition system** of possible **question-answer** interactions
- Dialogue storyline specification has **loops**
- Some **(q,a)** pairs **"progress"** the story forward from **S1** to **S2,** from **S2** to **S3**, etc.
- Some **(q,a)** pairs are **"texture"** for the story which remains in the same state



Dialogue Systems in Videogames 2.0 Behavior Composition

Input

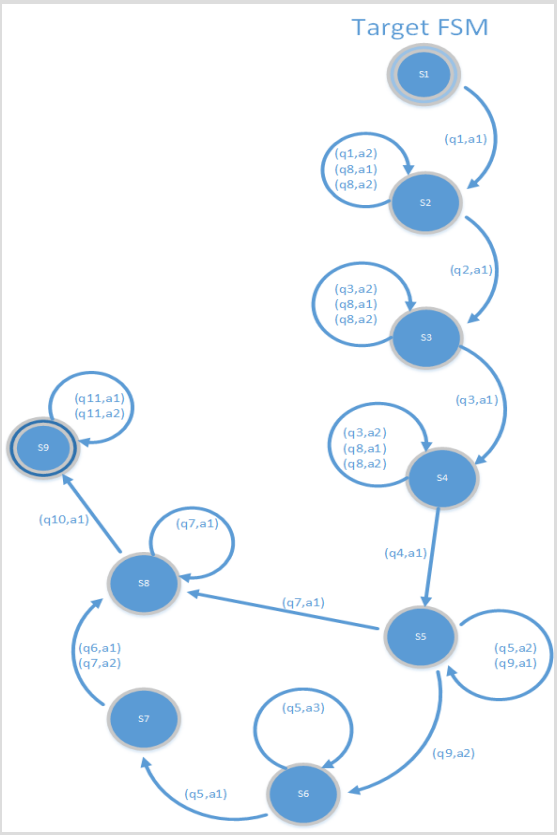
- A target **intended storyline** expressed also as a **transition system** of possible **question-answer** interactions
- Dialogue storyline specification has **loops**
- Some **(q,a)** pairs **"progress"** the story forward from **S1** to **S2,** from **S2** to **S3**, etc.
- Some **(q,a)** pairs are **"texture"** for the story which remains in the same state
- Some **(q,a)** pairs lead to **different states** of the story, essentially **branching** to more than one alternatives

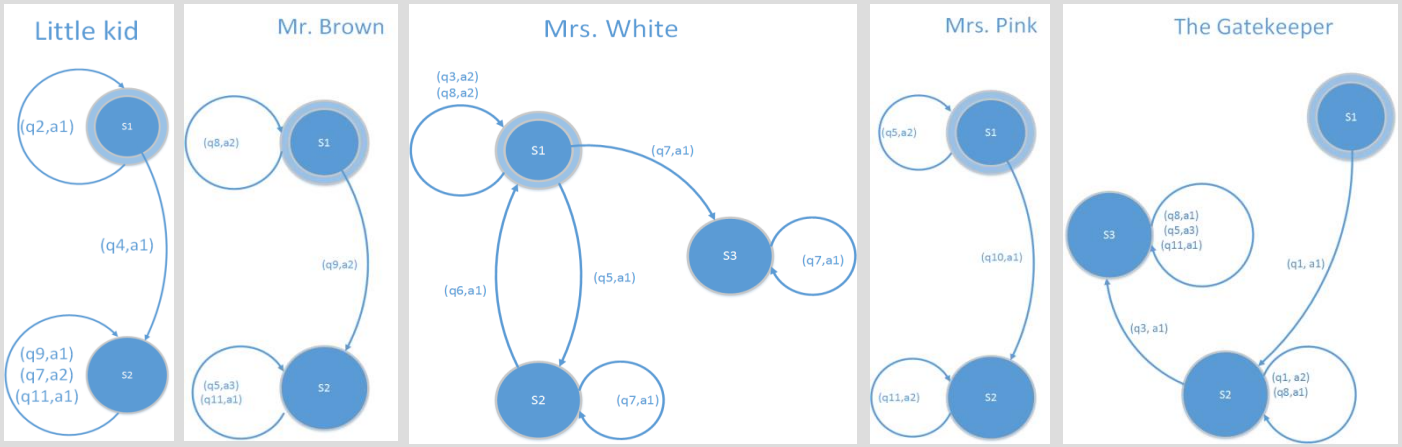


Dialogue Systems in Videogames 2.0
Behavior Composition

Output

- A computed strategy expressed as a **lookup-table** that says **which character** should engage with a possible **question-answer interaction**
- **Ensures** that for **every possible unfolding** there will **always be some character to realize** the question-answer pairs of **the intended storyline**



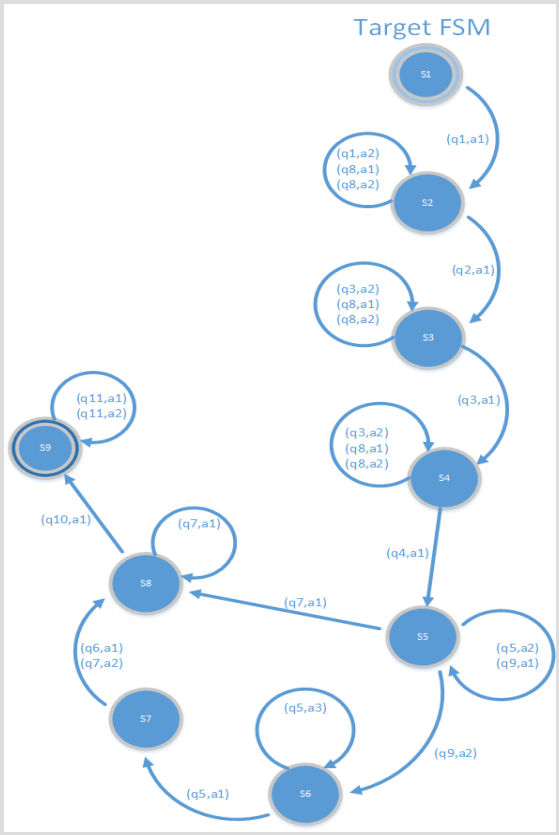


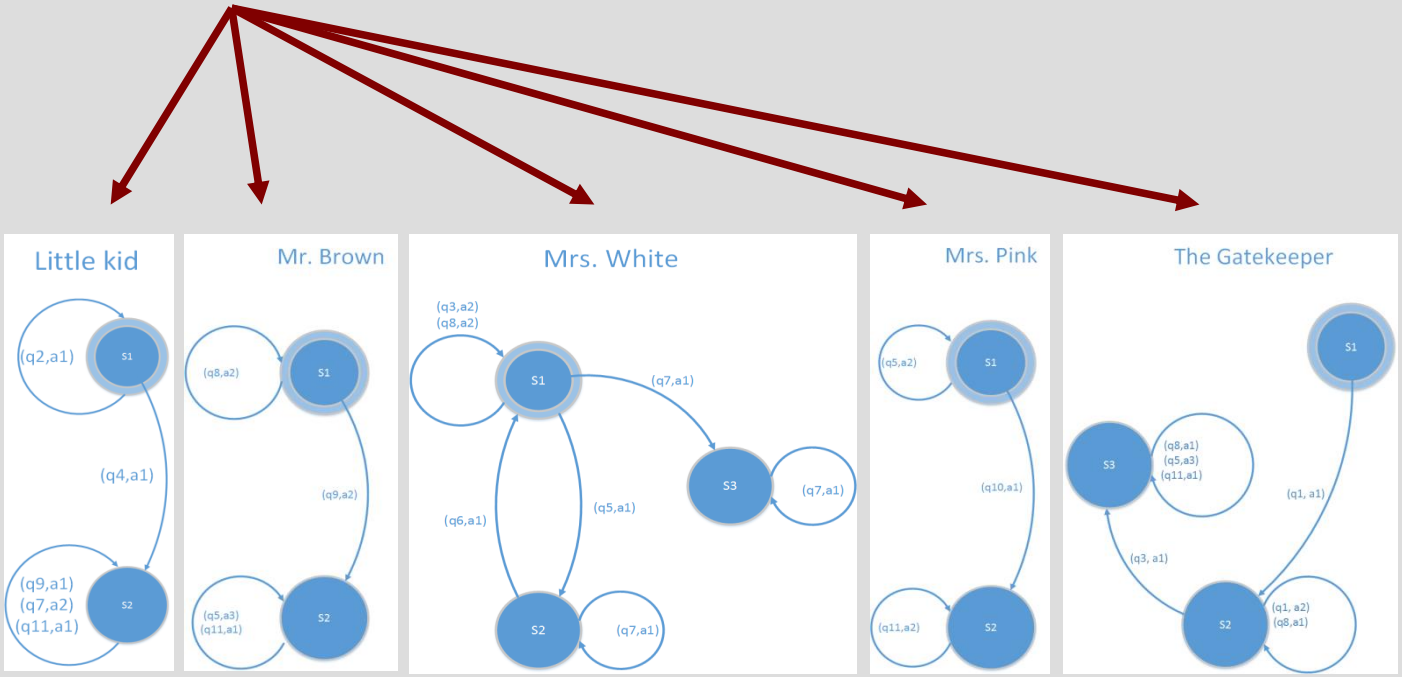
SAPIENZA UNIVERSITÀ DI ROMA

Dialogue Systems in Videogames 2.0
Behavior Composition

AI Behavior Composition

- Intended storyline of Dialogue System:
  Target Behavior
- Characters in the game:
  Available Behaviors





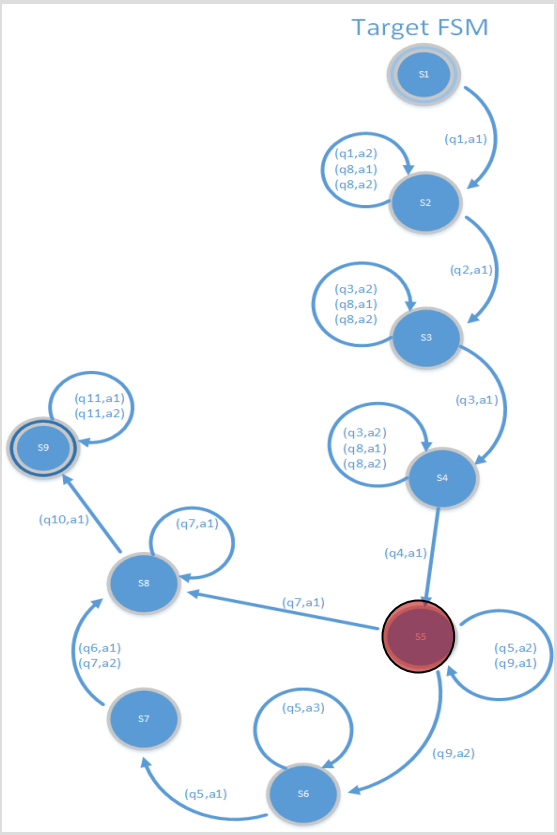
SAPIENZA
UNIVERSITÀ DI ROMA

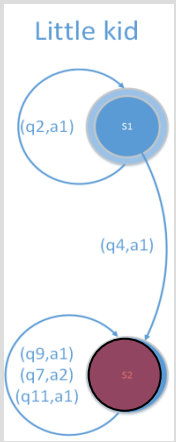
Dialogue Systems in Videogames 2.0 Behavior Composition

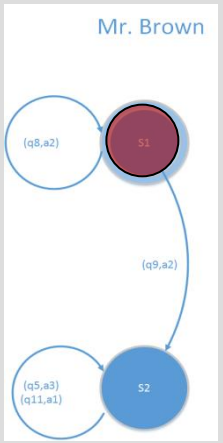
AI Behavior Composition

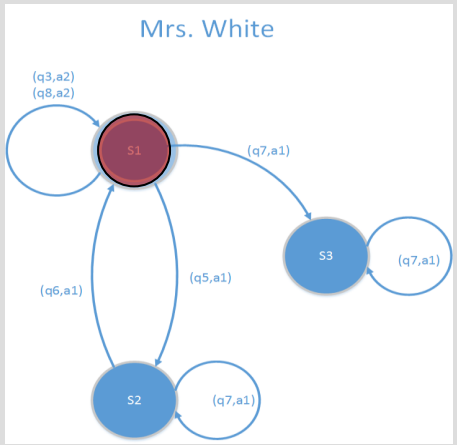
- Output strategy as a look-up table: **Controller Generator (CG)**

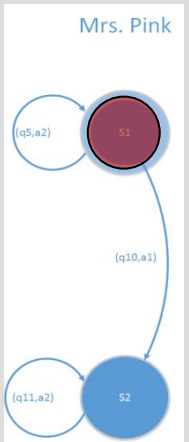
LK	MrB	MrsW	MrsP	GK	Target	QA	Assign
S2	S1	S1	S1	S3	S5	(q5,a2)	MrsP
S2	S1	S1	S1	S3	S5	(q7,a1)	MrsW
S2	S1	S1	S1	S3	S5	(q9,a1)	LK
S2	S1	S1	S1	S3	S5	(q9,a2)	MrB

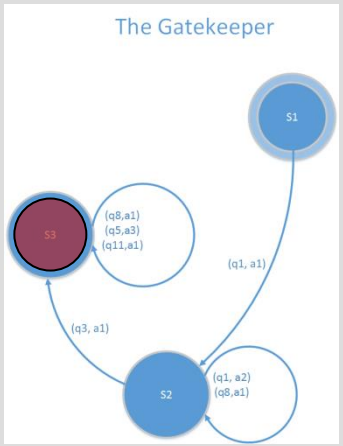












Dialogue Systems in Videogames 2.0
Behavior Composition

AI Behavior Composition

- Output strategy as a look-up table:
  Controller Generator (CG)
- Characters are **resources** with which a Drama Manager realizes the desired process described by the Target transition system
- A **CG** is a table that indicates the **available dialogue interactions** and the **corresponding characters** that can **facilitate** them
- A **Drama Manager** can use such CG in order to make decisions about how the story should continue (e.g. which options to offer to the player), while keeping it consistent with the flow specified by the Target Behavior

Dialogue Systems in Videogames 2.0 Behavior Composition

Benefits

- Intuitive way to formalize the desired unfolding of the dialogue system storyline as a transition system
- In videogame development it is common to model character behavior as transition systems (in variants of Finite State Machines)
- Flexible and modular approach for storyline generation: the Target Behavior is not directly linked to the Available Behaviors, so they can be easily plugged in/out at desing level.
- A successful computation of the output strategy CG ensures no deadlocks for any possible unfolding of the storyline expressed as a Target Behavior

Dialogue Systems in Videogames 2.0 Architecture

JaCO: Behavior Composition RESTful web service

- Input: Available Behaviors and Target Behavior in XML
- Output: Controller Generator lookup table in XML
- (Also available command-line)

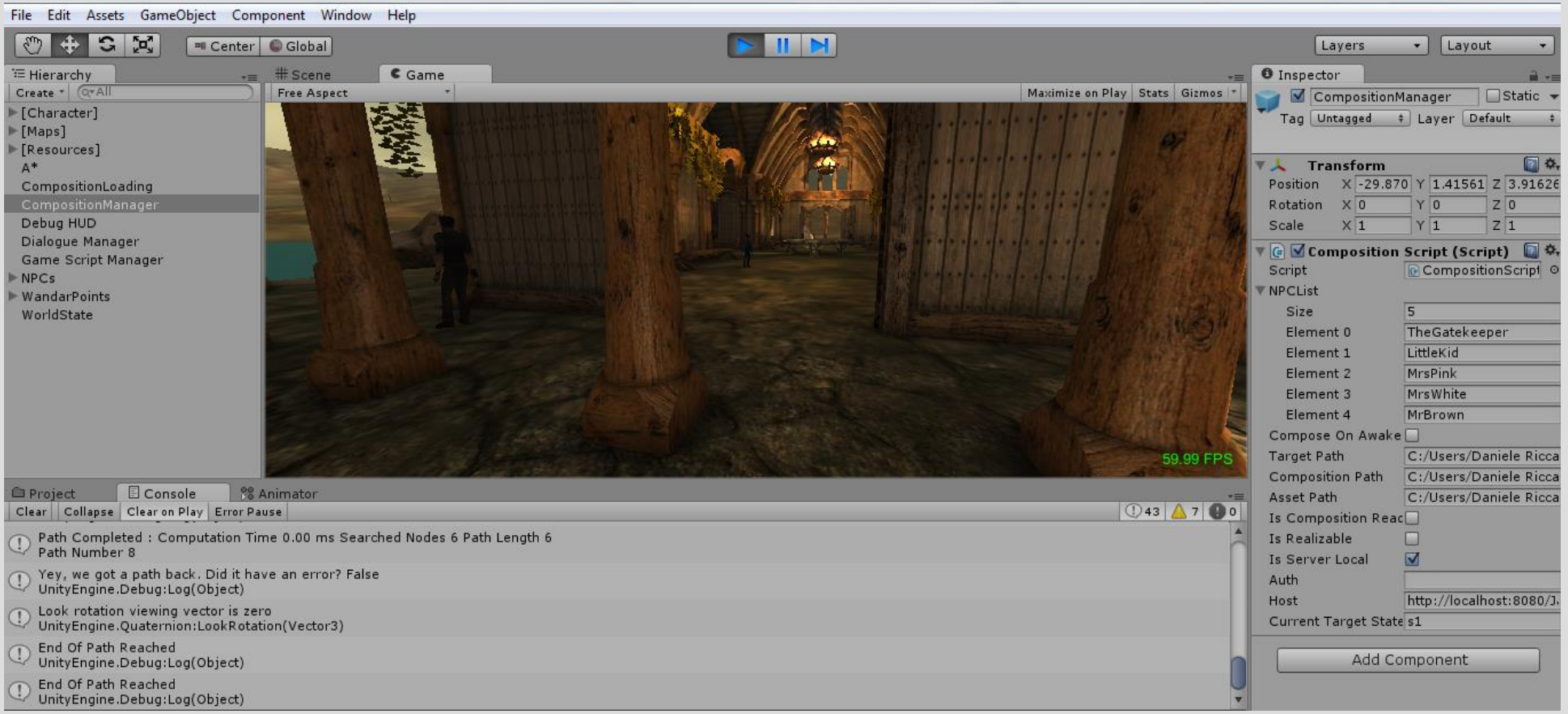
```
java -jar jaco.jar
Next NPC to be invoked: LittleKid
=====================================
Target state: s9
Available behaviors:
  LittleKid: s2
  MrBrown: s2
  MrsPink: s2
  MrsWhite: s2
  TheGatekeeper: s3
Next action: q11a1
Next NPC to be invoked: TheGatekeeper
=====================================
Target state: s9
Available behaviors:
  LittleKid: s2
  MrBrown: s2
  MrsPink: s2
  MrsWhite: s2
  TheGatekeeper: s3
Next action: q11a2
Next NPC to be invoked: MrsPink
=====================================
9 states found.
Did we have a composition? true
```

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<behavior>
    <name>Target</name>
    <finiteStateMachine>
        <state node="s1">
            <transition action="q1a1">
                <target>s2</target>
            </transition>
        </state>
        <state node="s2">
            <transition action="q1a2">
                <target>s2</target>
            </transition>
            <transition action="q8a1">
                <target>s2</target>
            </transition>
            <transition action="q8a2">
                <target>s2</target>
            </transition>
            <transition action="q2a1">
                <target>s3</target>
            </transition>
        </state>
        <!-- other lines hidden for brevity -->
    </finiteStateMachine>
</behavior>
```

SAPIENZA UNIVERSITÀ DI ROMA

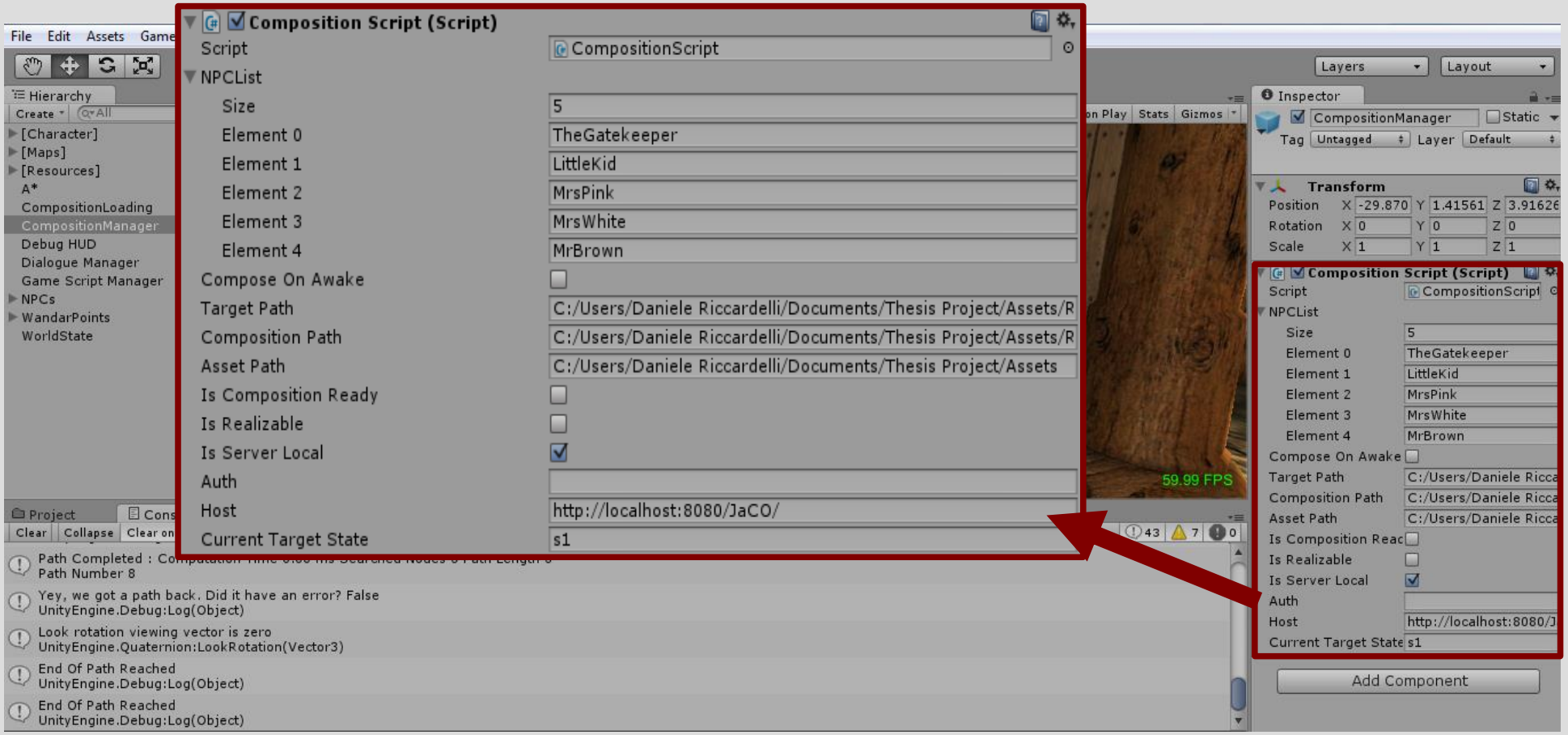
Dialogue Systems in Videogames 2.0 Architecture

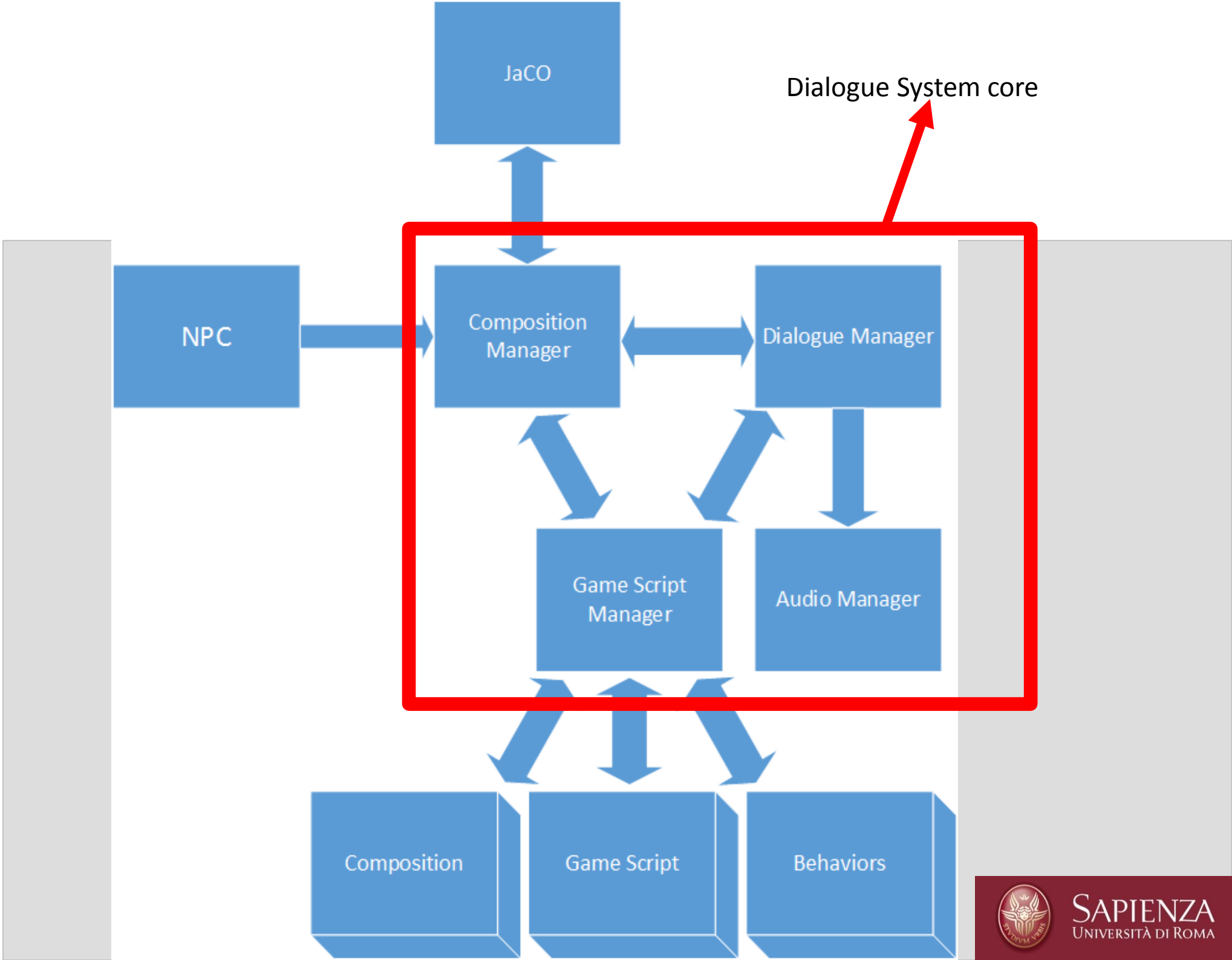
Unity game engine (www.unity3d.com)

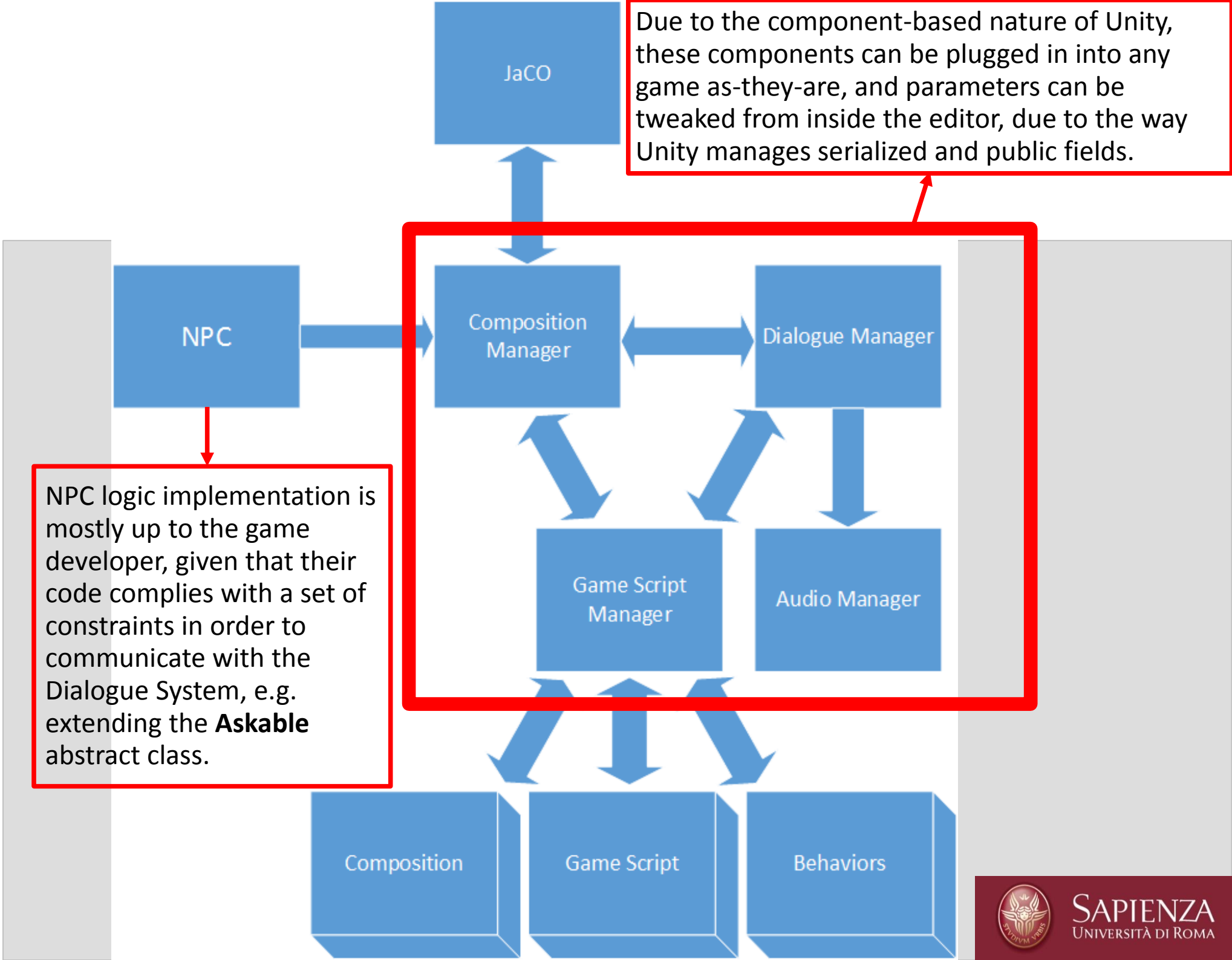


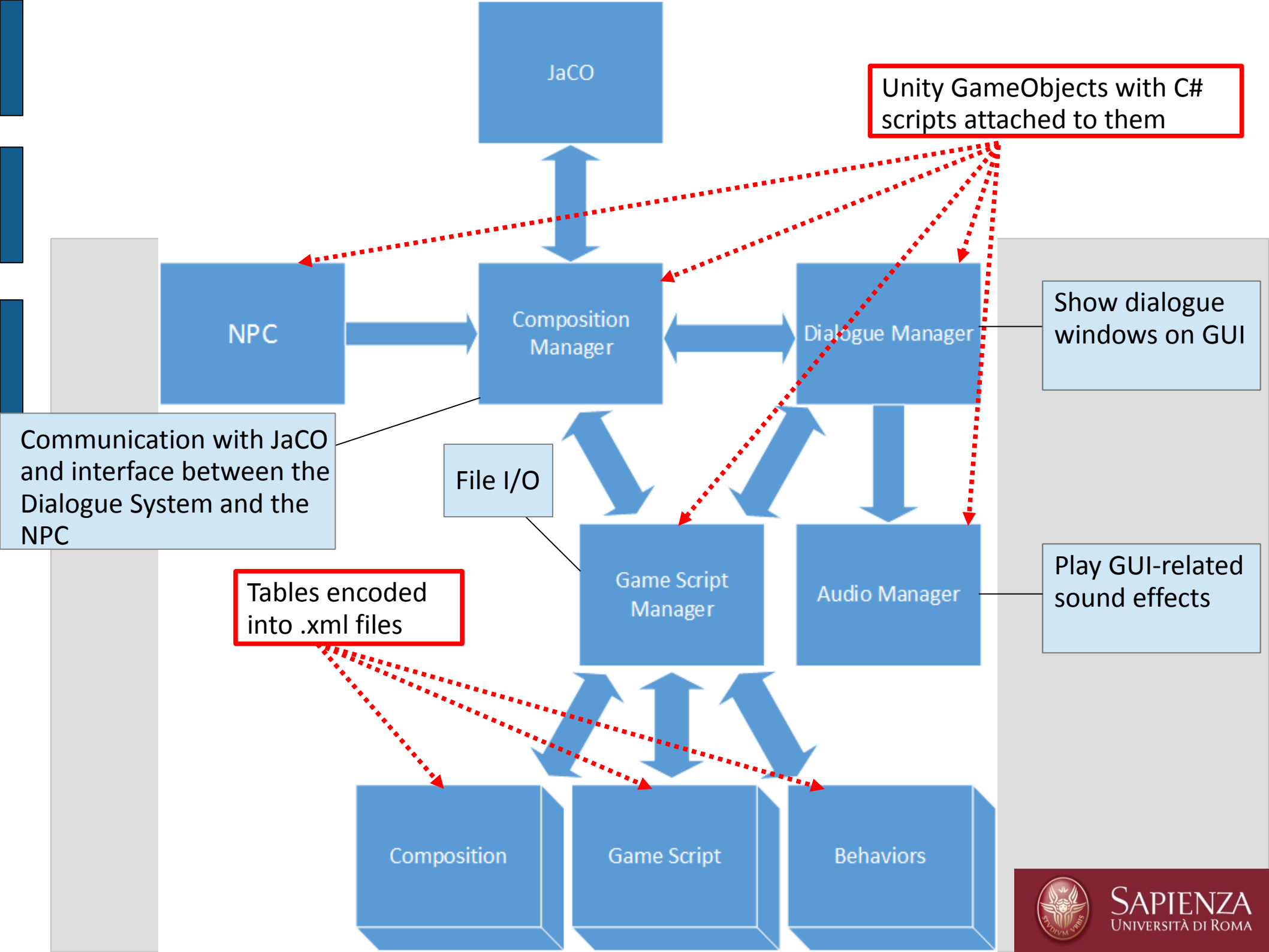
Dialogue Systems in Videogames 2.0 Architecture

Unity game engine (www.unity3d.com)





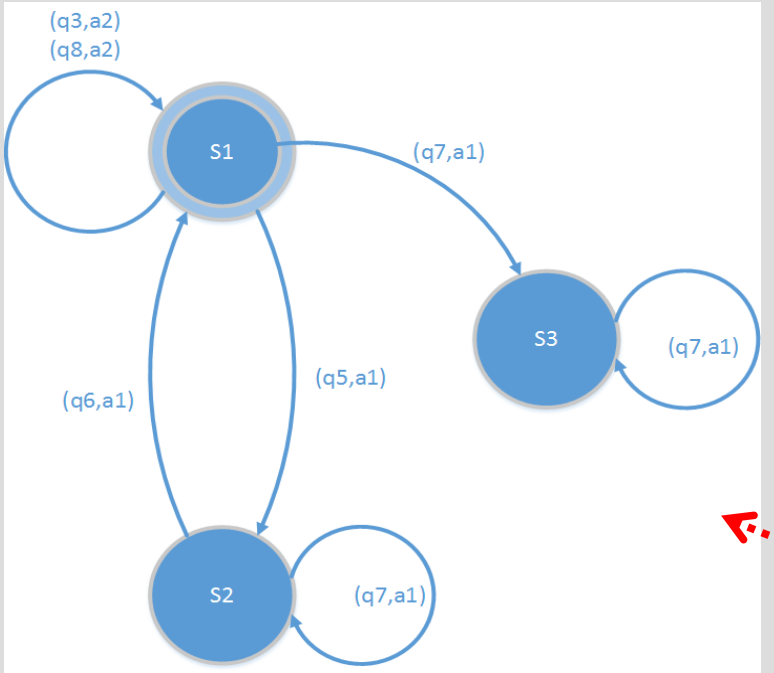

JaCO
Due to the component-based nature of Unity, these components can be plugged in into any game as-they-are, and parameters can be tweaked from inside the editor, due to the way Unity manages serialized and public fields.
NPC
Composition Manager
Dialogue Manager
Game Script Manager
Audio Manager
NPC logic implementation is mostly up to the game developer, given that their code complies with a set of constraints in order to communicate with the Dialogue System, e.g. extending the **Askable** abstract class.
Composition
Game Script
Behaviors


JaCO
Unity GameObjects with C# scripts attached to them
NPC
Composition Manager
Dialogue Manager
Show dialogue windows on GUI
Communication with JaCO and interface between the Dialogue System and the NPC
File I/O
Game Script Manager
Audio Manager
Play GUI-related sound effects
Tables encoded into .xml files
Composition
Game Script
Behaviors

Dialogue Systems in Videogames 2.0 Architecture

Two levels for the **behavior** of characters in the game

- A **high-level behavior** capturing their **“role”** in the **dialogue system**
- A **low-level behavior** capturing their **“physical” interactions** in the game-world (here, just random walker)

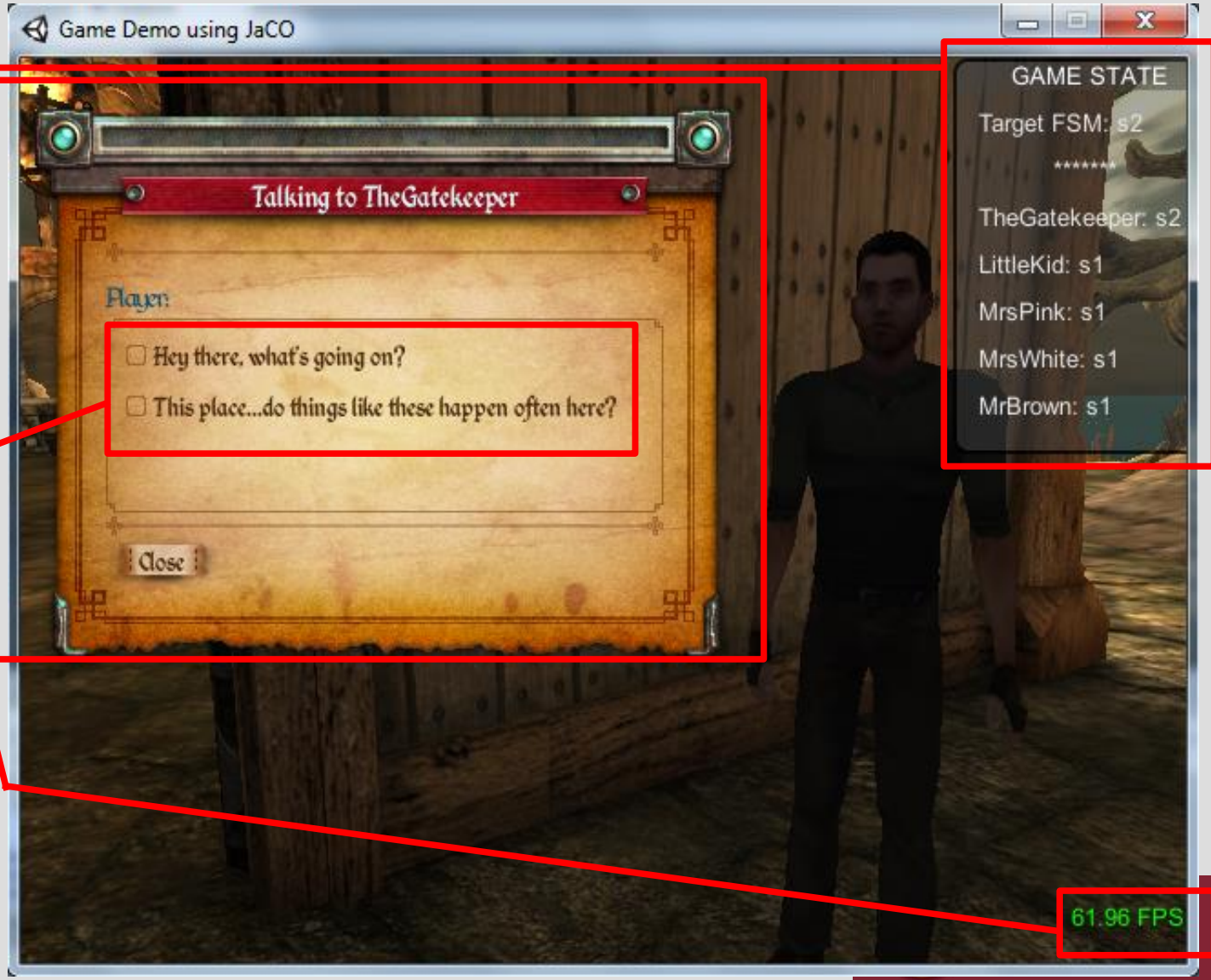


“Uncommon Crime Scene” Demo

Debug HUD

Dialogue GUI

Player-NPC allowed interactions according to the NPC and Target state



“Uncommon Crime Scene” Demo

- Simple mini-game where the player's task is to **interrogate** the **witnesses** of a crime and **find** the **criminal**
- **5 characters** populating the scene with which the player can **talk**
- An **interaction** consists of a question asked by the player paired with an answer by the character
- The **same question** can have **different answers**, according to **when** (Target state) and **who** (Available Behaviors state) we ask

A possible script unfolding

(Q1,A1) **Player:** *Hey there, what's going on?*
The Gatekeeper: *A terrible thing just happened here! Go inside and investigate! Quick!*

(Q8,A2) **Player:** *This place...do things like these happen often here?*
Mr. Brown: *Not at all, I wouldn't say so.*

(Q2,A1) **Player:** *You, little kid: do you know anything about this crime?*
Little kid: *I could tell you...if only you could give me something in return...first.*

(Q3,A2) **Player:** *The little kid is looking for something he's lost. Do you know what that is?*
Mrs. White: *Hmm, I have no idea. I barely see that kid around.*

(Q3,A1) **Player:** *The little kid is looking for something he's lost. Do you know what that is?*
The Gatekeeper: *Oh, I guess I do! I found this in the yard, this morning.*

(Q4,A1) **Player:** *Here you are. Can you tell me now?*
Little kid: *If I were you, I would ask Mrs. White over there...*

(Q7,A1) **Player:** *What can you tell me about the woman in the pink dress?*
Mrs. White: *I saw her running away as I heard that scream...luckily Mr. Brown stopped her.*

(Q10,A1) **Player:** *It was you! I asked witnesses! CONFESS!!!*
Mrs. Pink: *Oh no! Yes, it was me...but I swear, that was just a cookie!*