LudoNarrare

A Model for Verb Based Interactive Storytelling

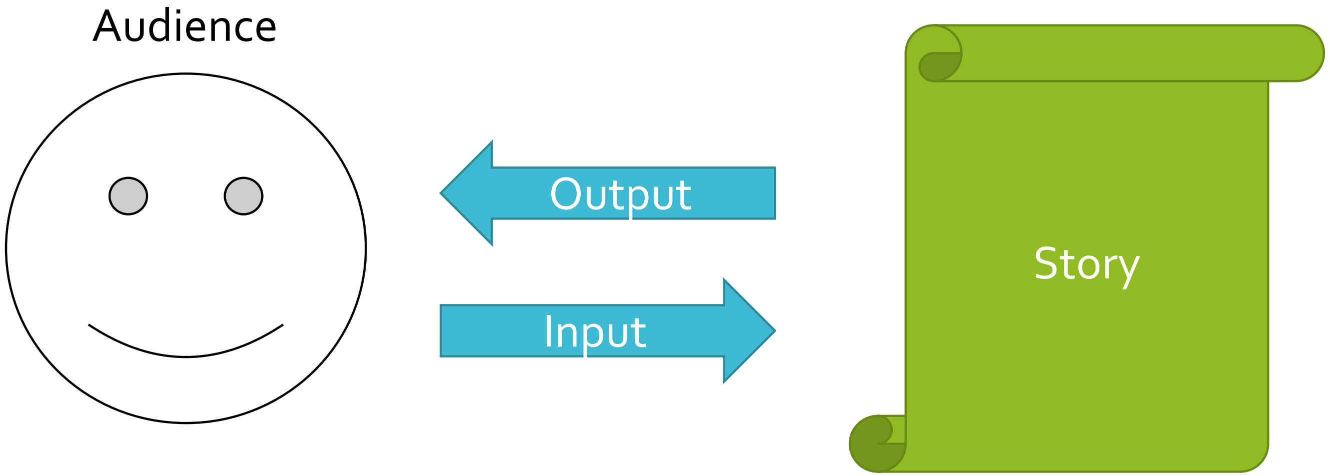
Presentation Outline

- Research Question – The Problem of Interactive Storytelling
- Review of Past Attempts at Solutions
- Concepts Behind LudoNarrare
- The Implementation of LudoNarrare
- Assessment of LudoNarrare as a Solution
- Going Forward

Research Question

The Problem of Interactive Storytelling

What is an Interactive Story?



- A story which is partially determined by real-time audience input.

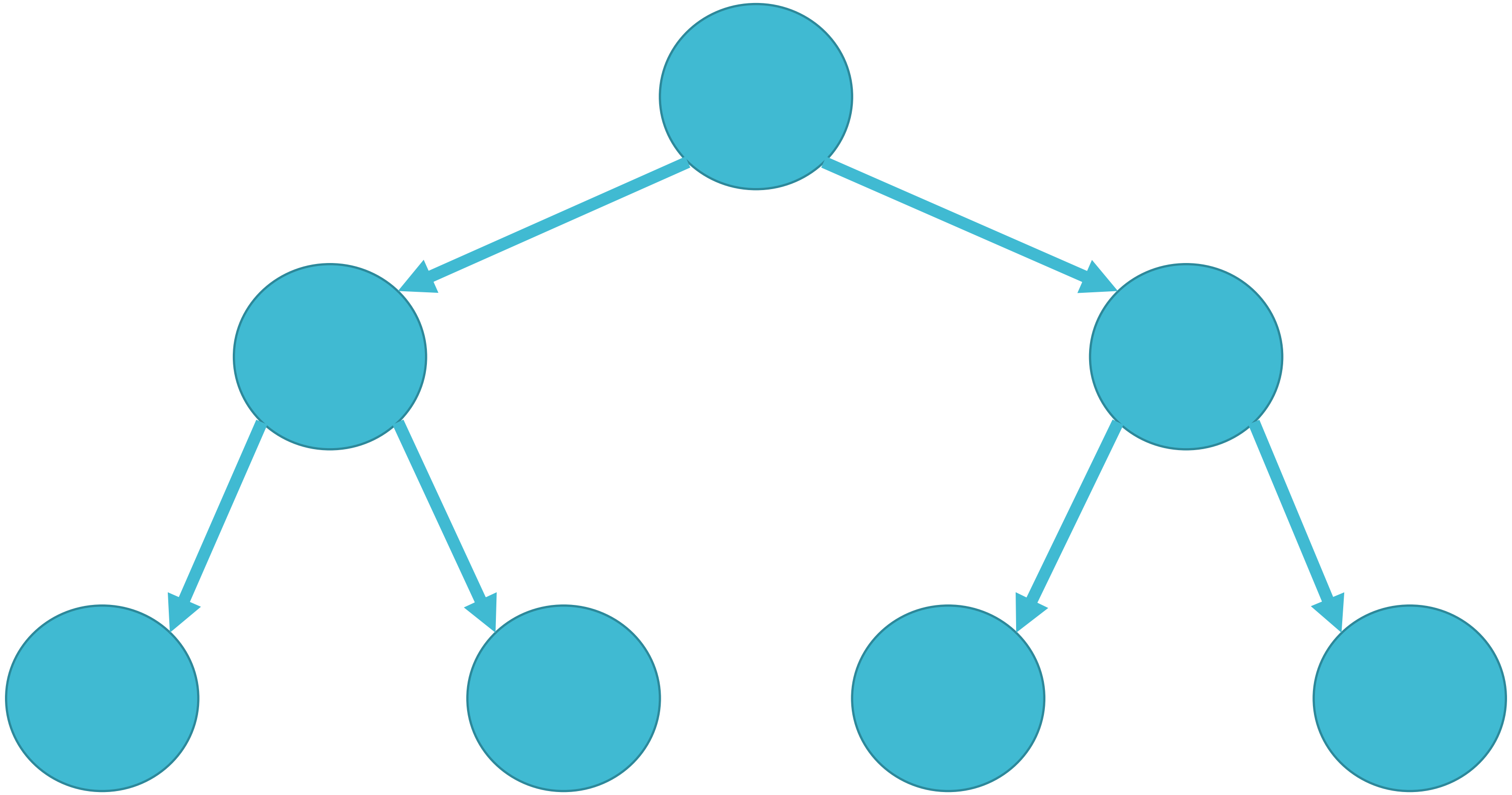
Problem of Computer Mediated Interactive Storytelling

- The very first stories, told by humans, were interactive.
- The challenge is in making a computer act as the storyteller.

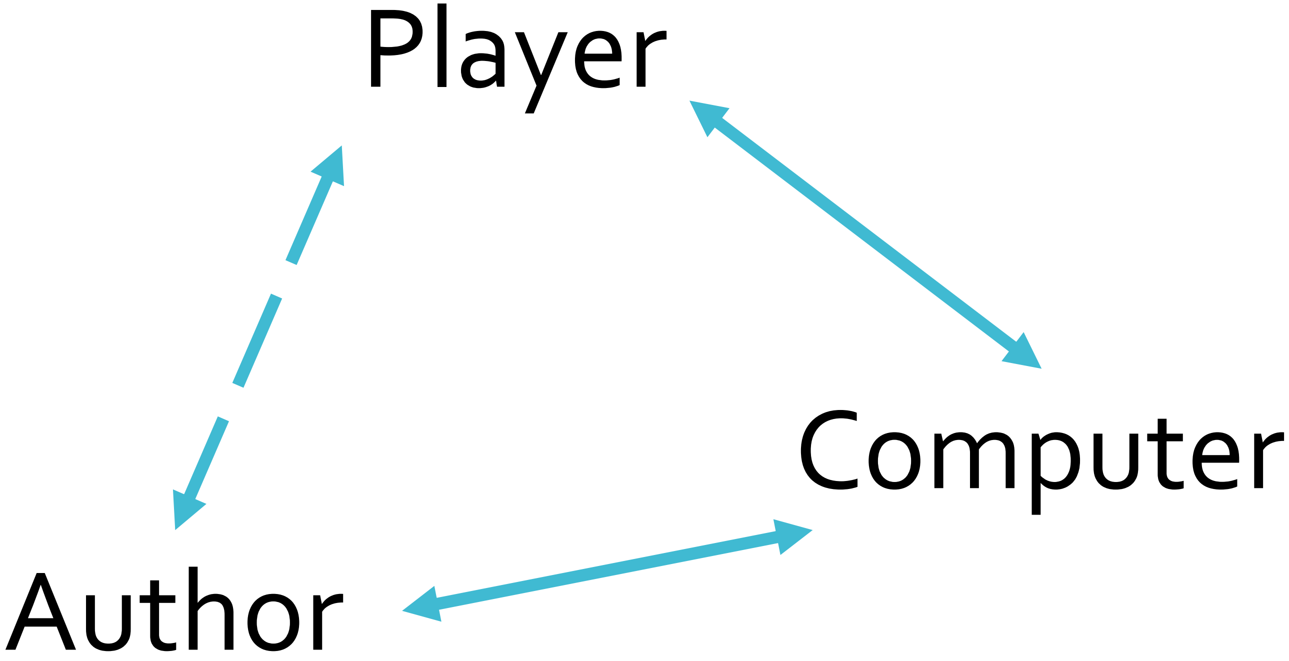
Goals of an Interactive Story

- Players act as a character in the unfolding drama of a story.
- Give players a meaningful sense of agency.
- Allow for a dialogue between the player and the author; permit the unexpected and subversive.
- Surprise the players and surprise the author.

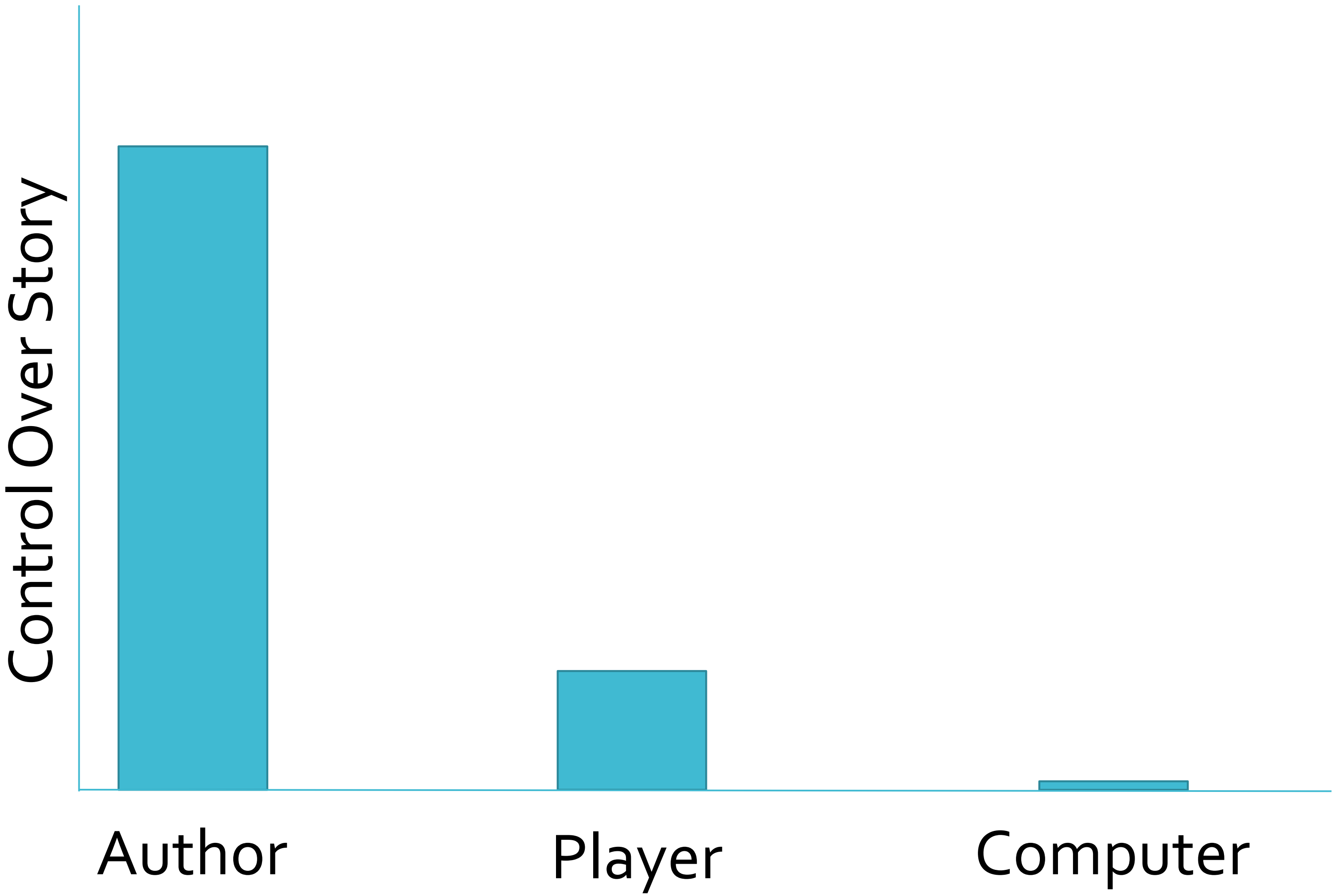
Problem: Static, Predefined Branching Paths as Interactive Story Model



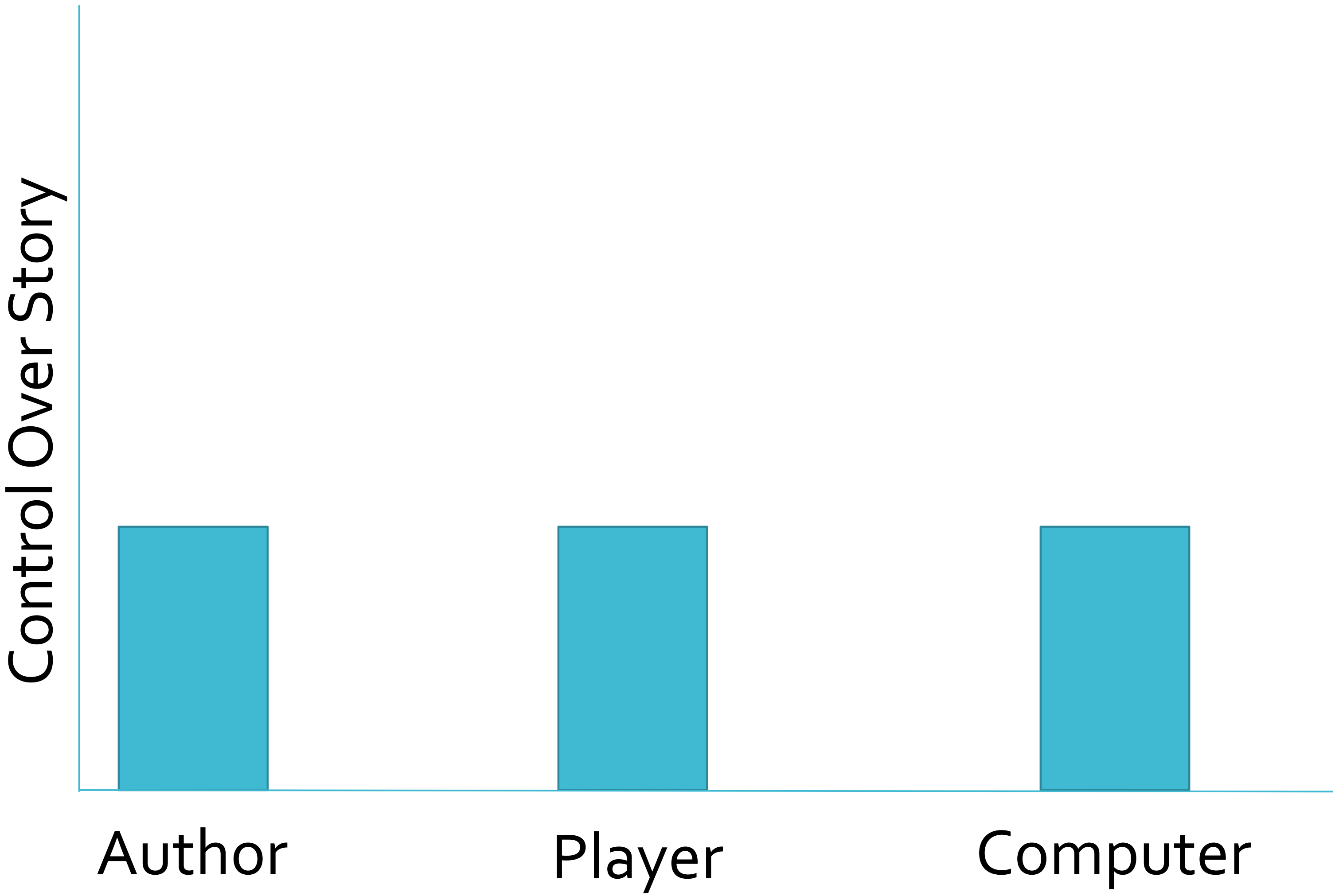
Three Parties Involved in an Interactive Story



Control Over the Story of Each Party with Branching Path Model



Ideal Control Over the Story of Each Party



An Interactive Storytelling Solution Accounts for All Three Parties

The Author's Tools

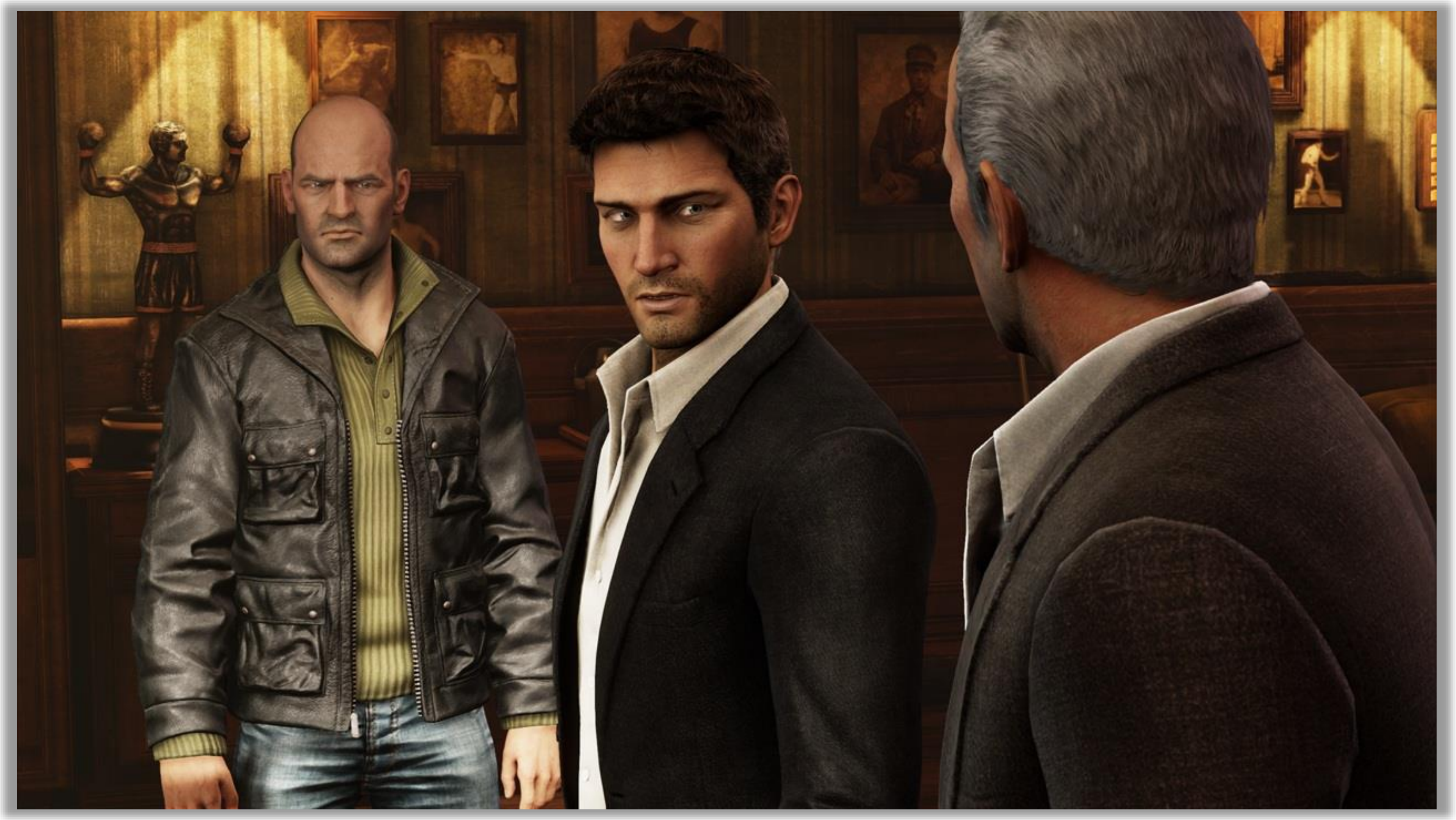
The Computer's Engine

The Player's Interface

Review of Past Attempts at Solutions

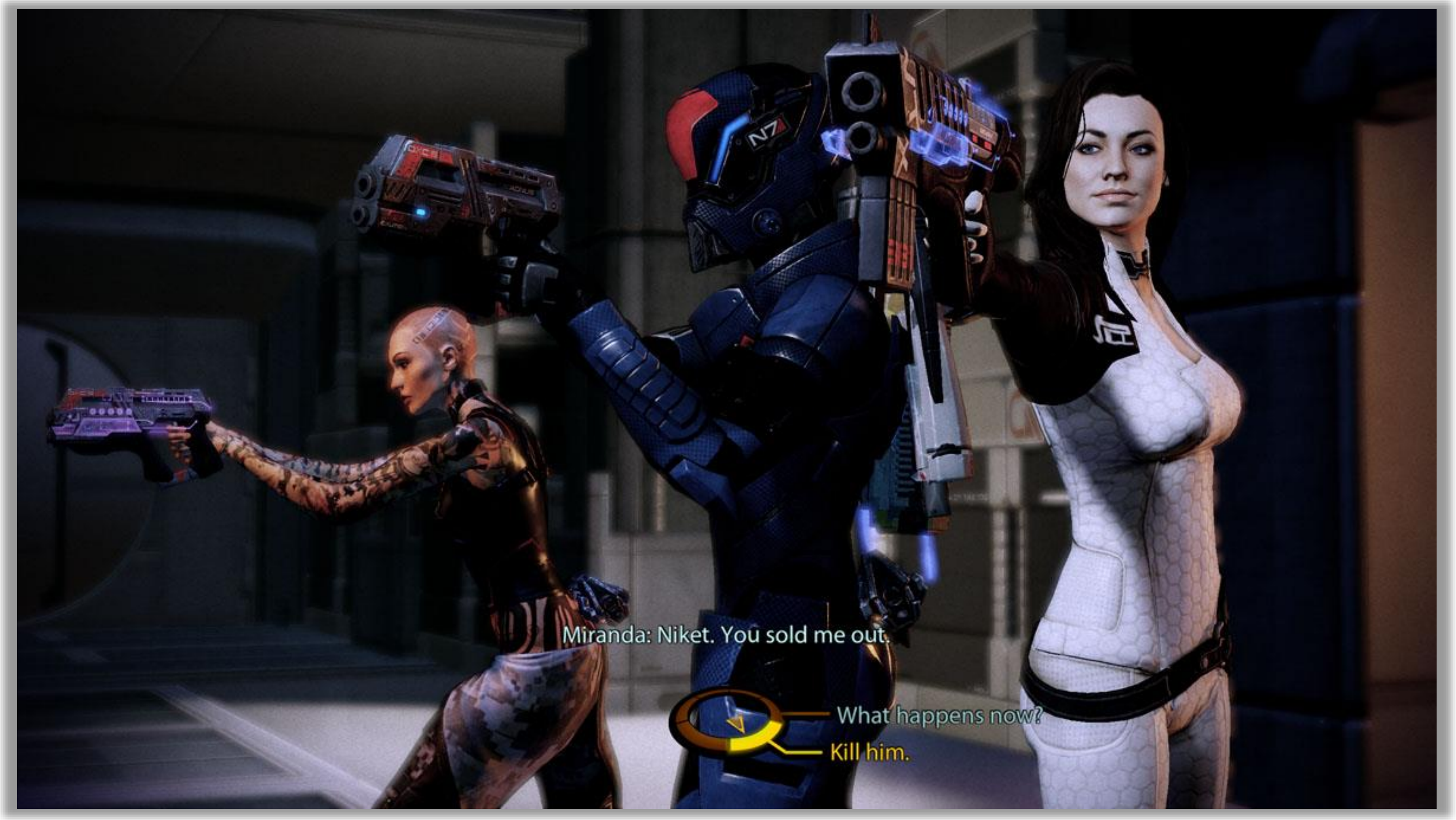
In Research and Practice

Traditional Video Game Cutscene Solution



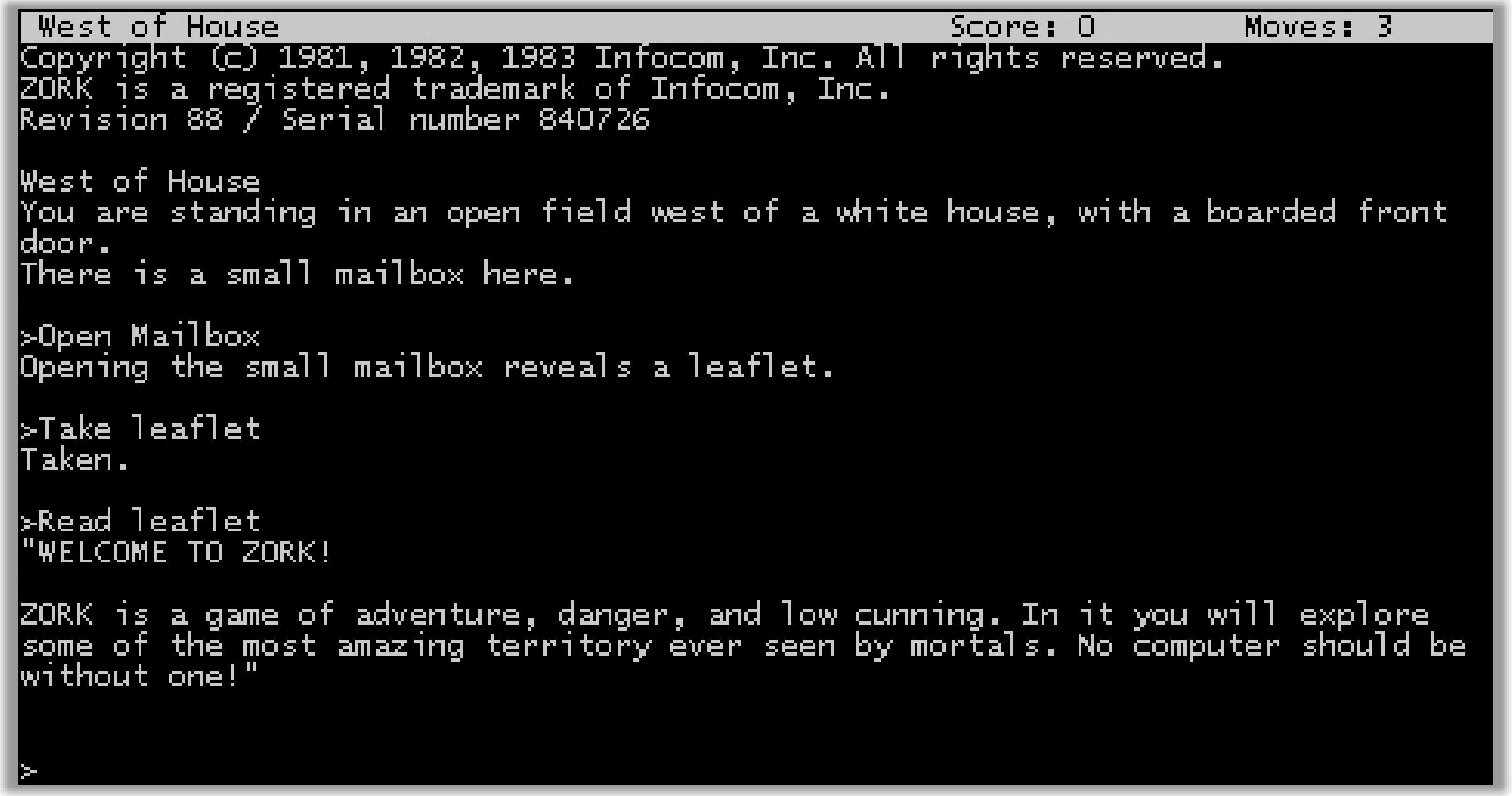
Example: Uncharted 3

Branching Paths Solution


Miranda: Niket. You sold me out.

Example: Mass Effect 2

Interactive Fiction Solution



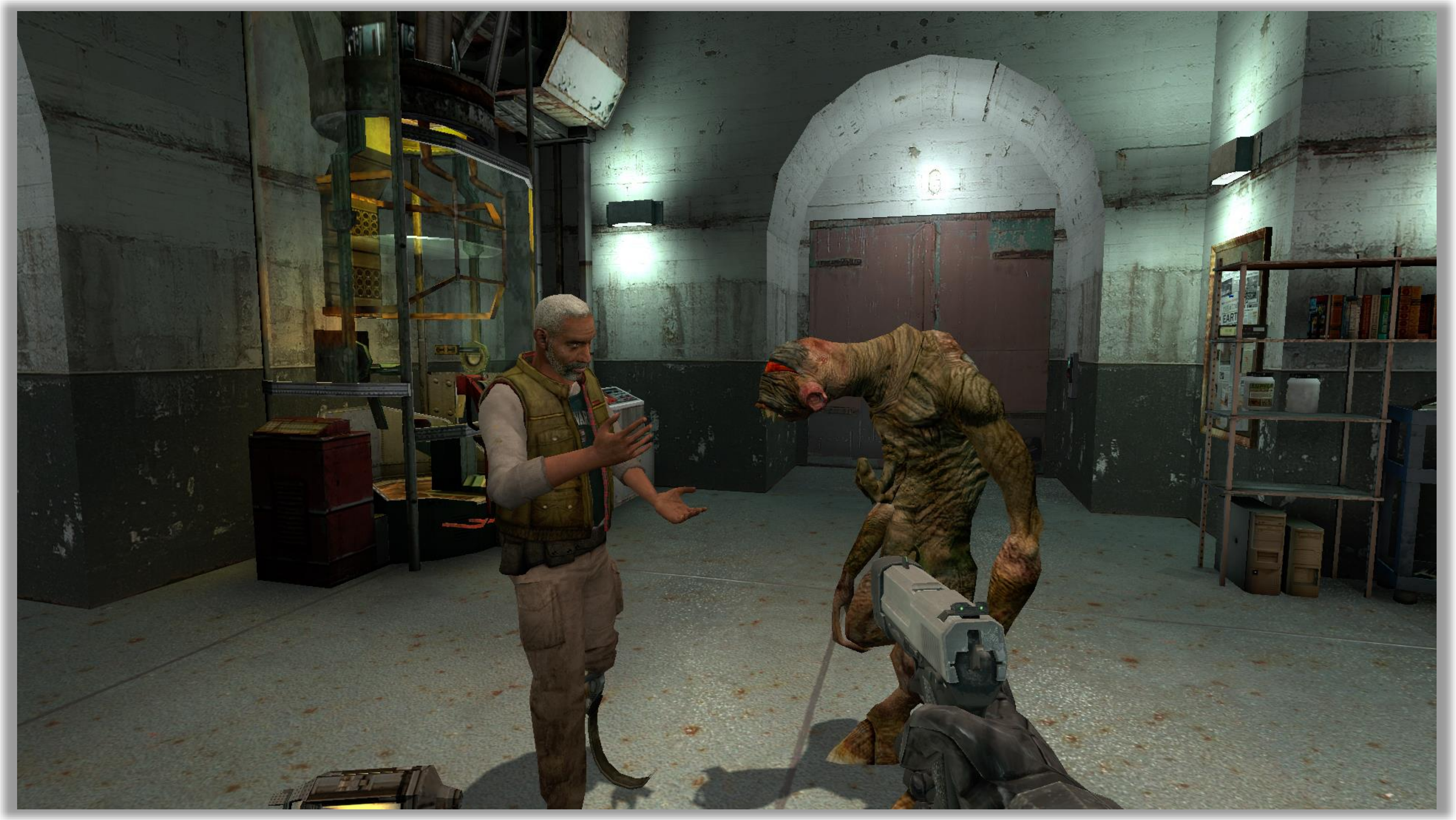
```
 West of House                                          Score: 0          Moves: 3
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with a boarded front
door.
There is a small mailbox here.

>Open Mailbox
Opening the small mailbox reveals a leaflet.

>Take leaflet
Taken.

>Read leaflet
"WELCOME TO ZORK!

ZORK is a game of adventure, danger, and low cunning. In it you will explore
some of the most amazing territory ever seen by mortals. No computer should be
without one!"


>
```

Example: Zork

Human Mediated Roleplaying Solution

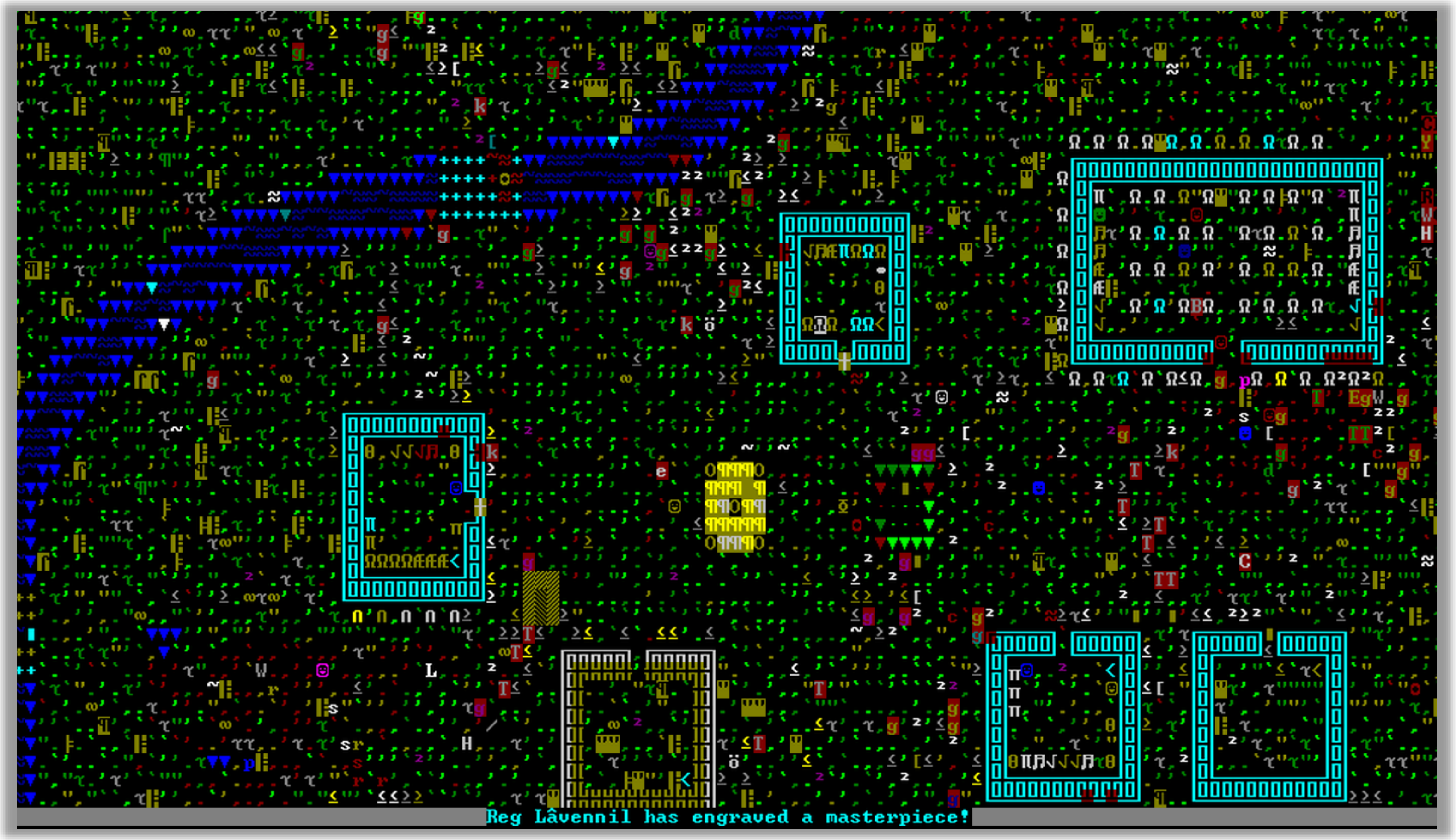
Example: Dungeons and Dragons

Environmental Storytelling Solution



Example: Half-Life 2

Complex Emergent Systems Solution

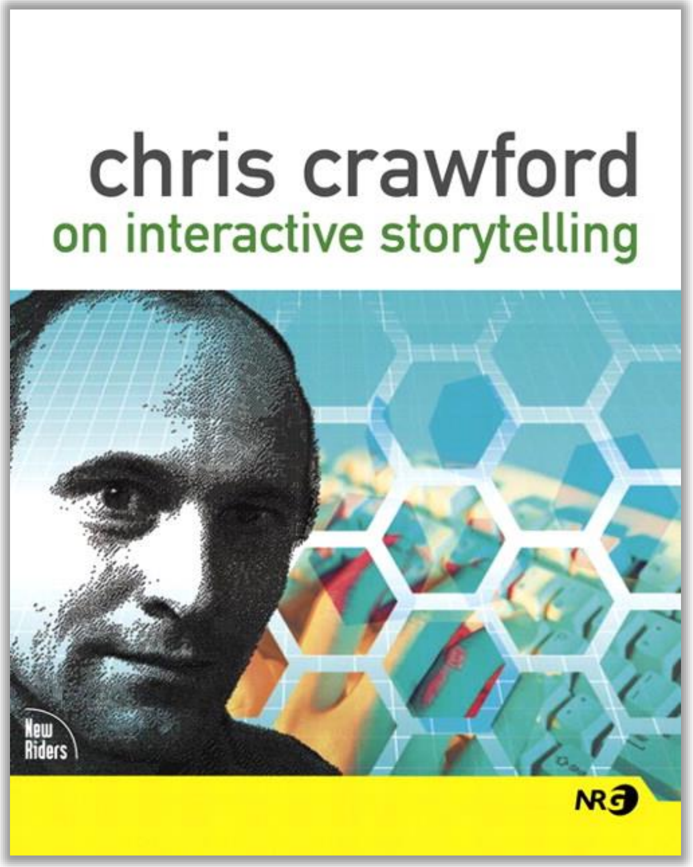


Reg Lâvennil has engraved a masterpiece!

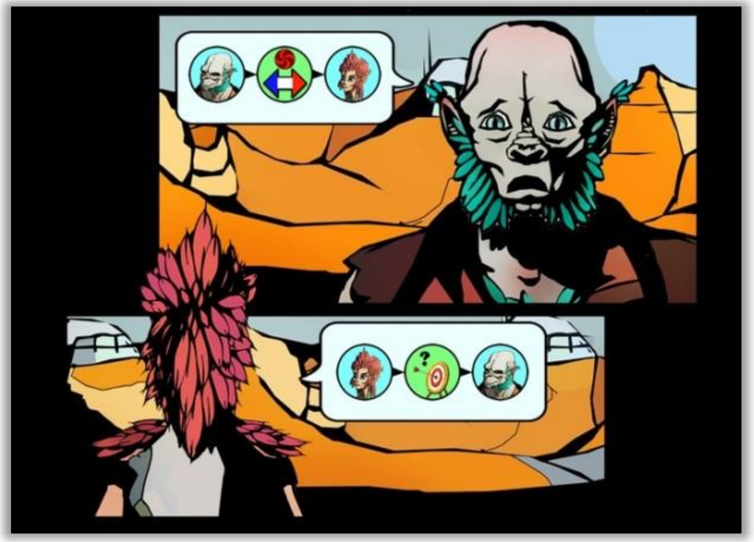
Example: Dwarf Fortress

Chris Crawford's Storytron and Siboot

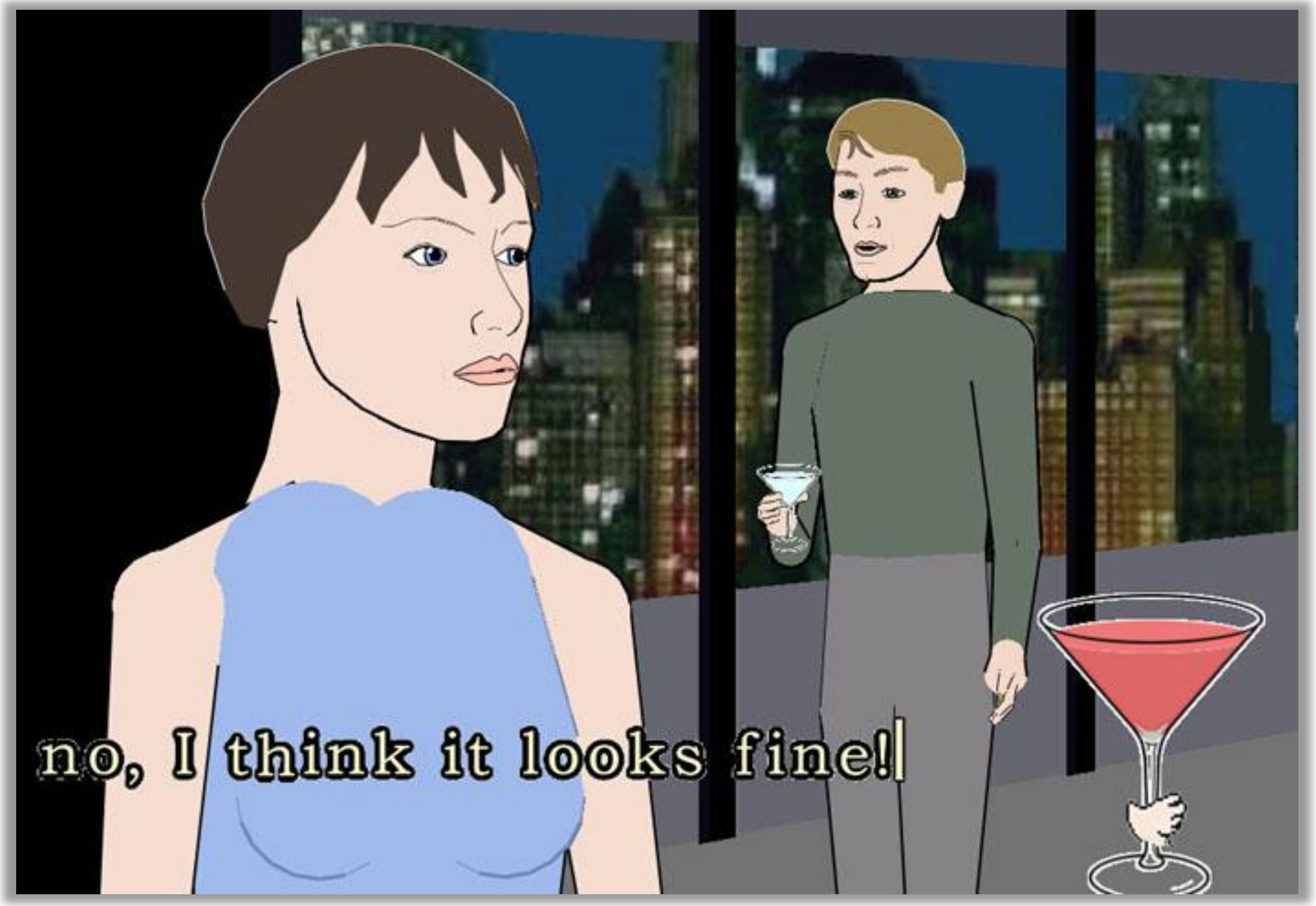
Crawford's Book and Storytron



Siboot



Academic Solutions



Example: Façade

Concepts Behind LudoNarrare

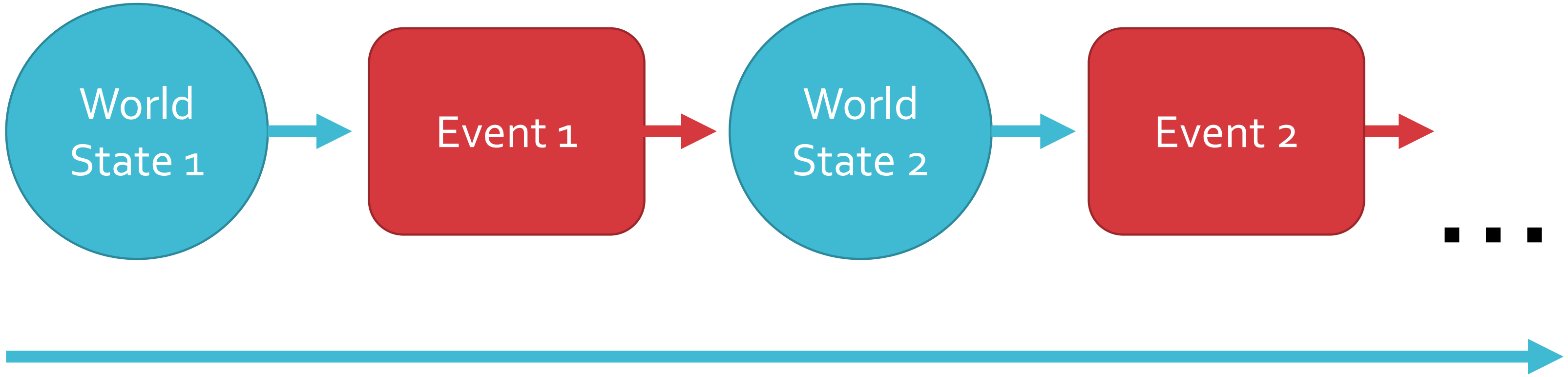
The World Transition and Verb-Event-Exposition Model

Two Part Model of Stories in LudoNarrare

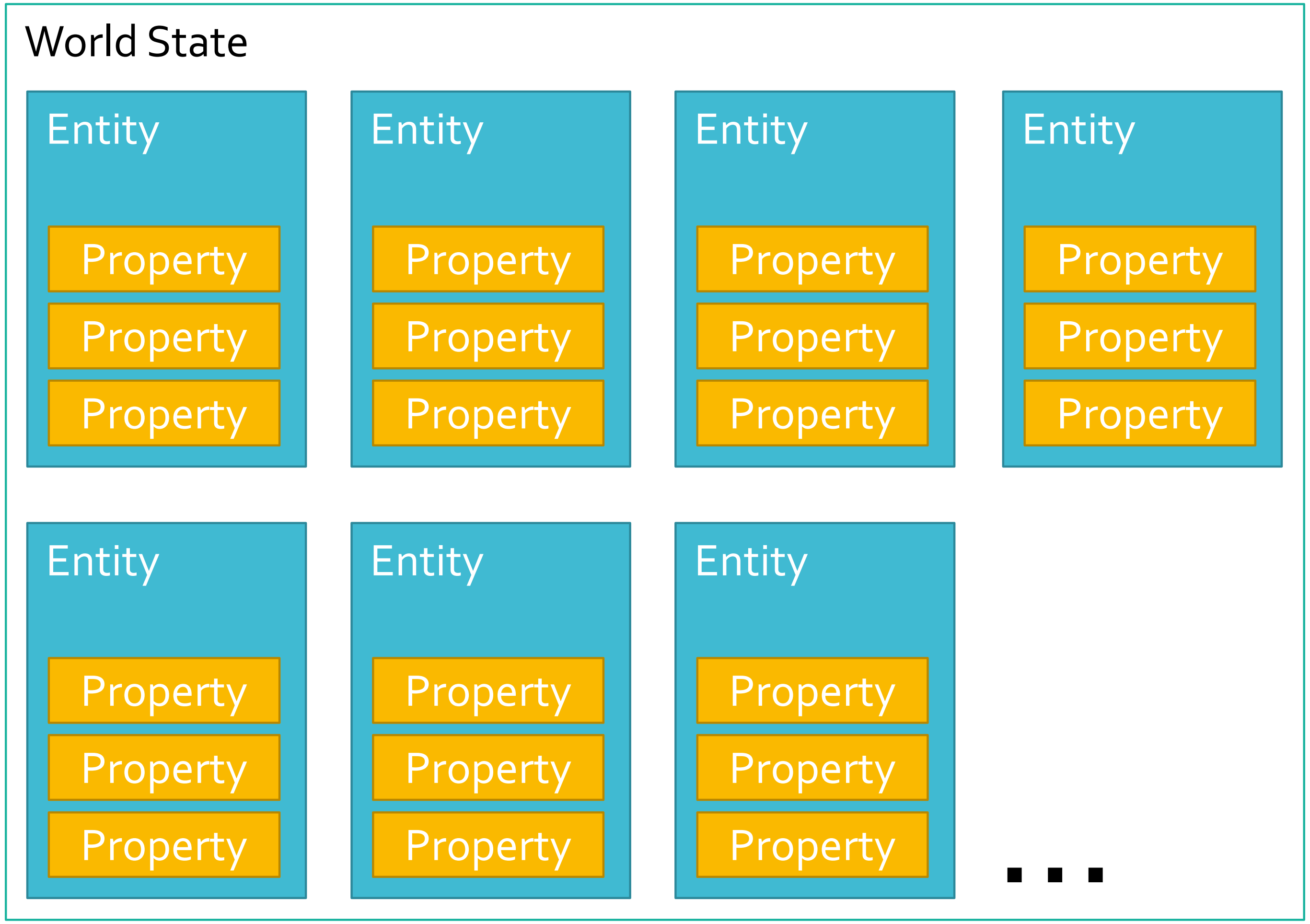
World

Verbs

Story as Series of World States and Events



Anatomy of a World State



World State

Entity
Property
Property
Property

Entity
Property
Property
Property

Entity
Property
Property
Property

Entity
Property
Property
Property

Entity
Property
Property
Property

Entity
Property
Property
Property

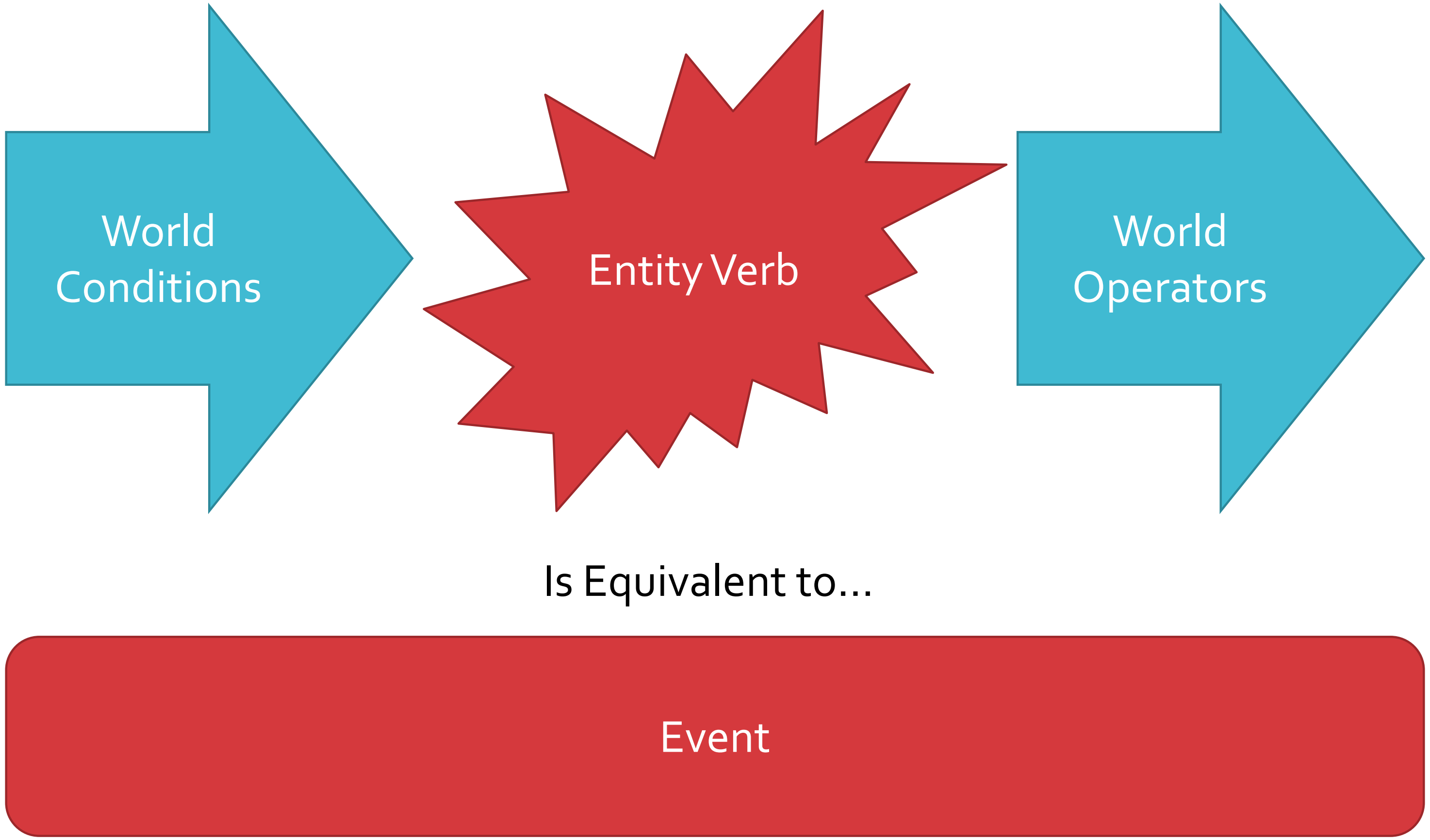
Entity
Property
Property
Property

...

Move away from spatial modeling to logical modeling.

Anatomy of an Event

Conditions and operators act as the logic of the storyworld.



World Conditions

Entity Verb

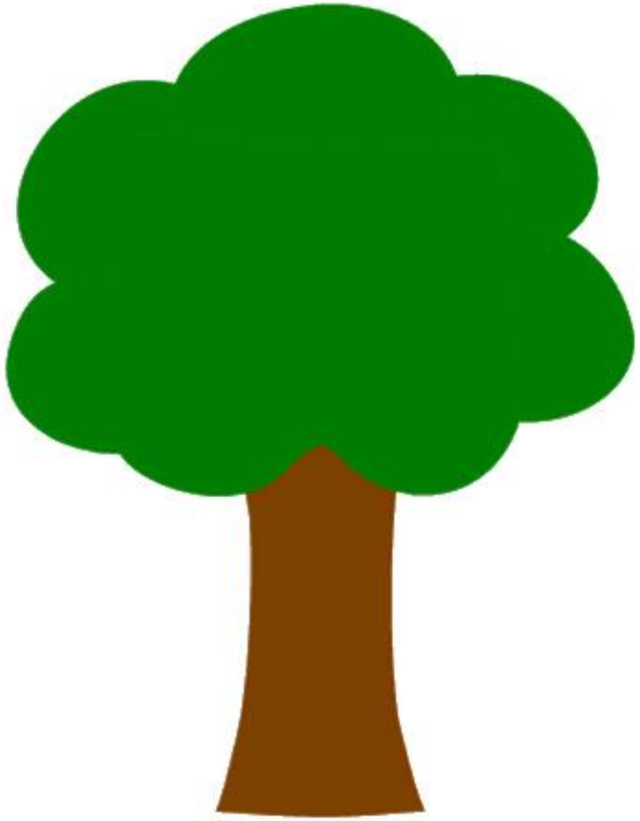
World Operators

Is Equivalent to…

Event

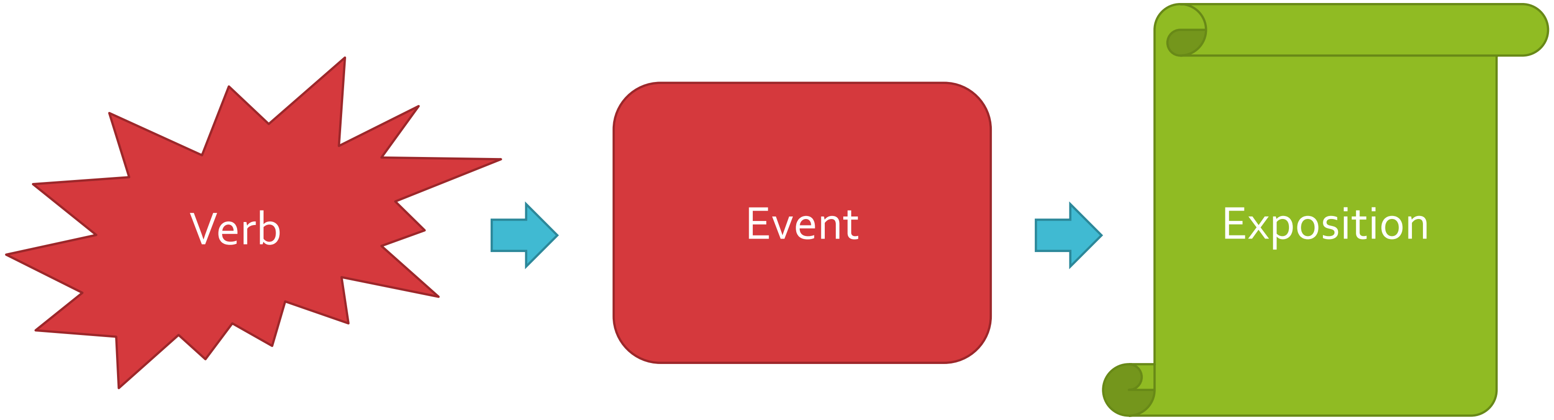
Presentation of an Event as Storytelling

Plot Alone

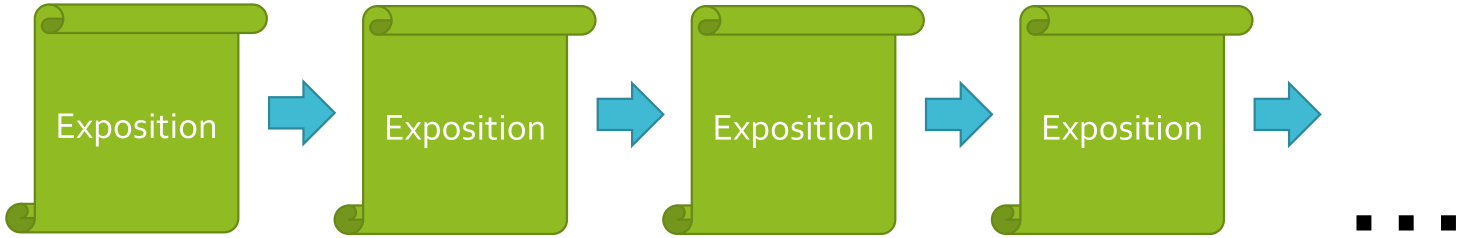


Plot + Telling = Storytelling

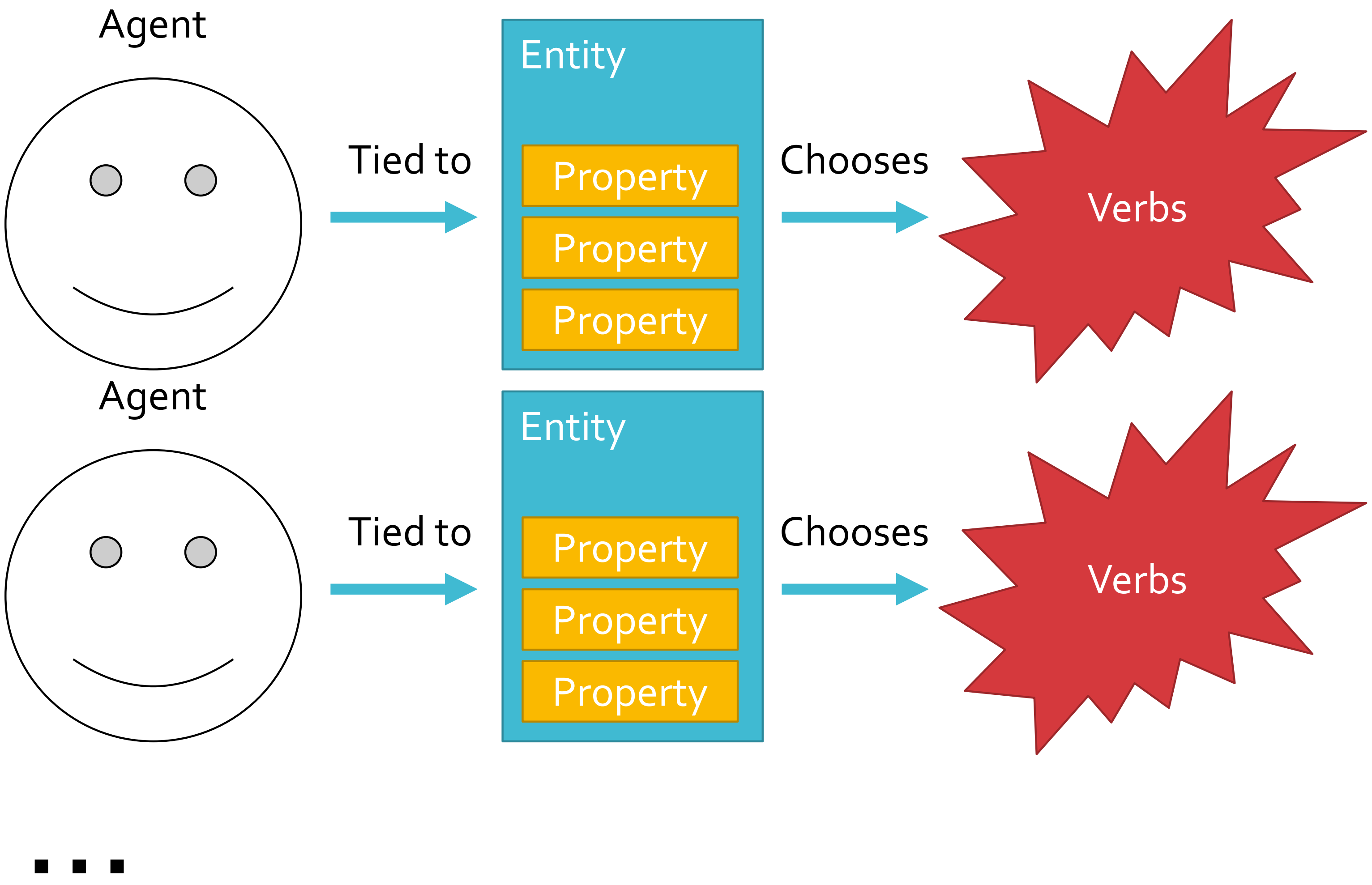
Verb to Event to Exposition



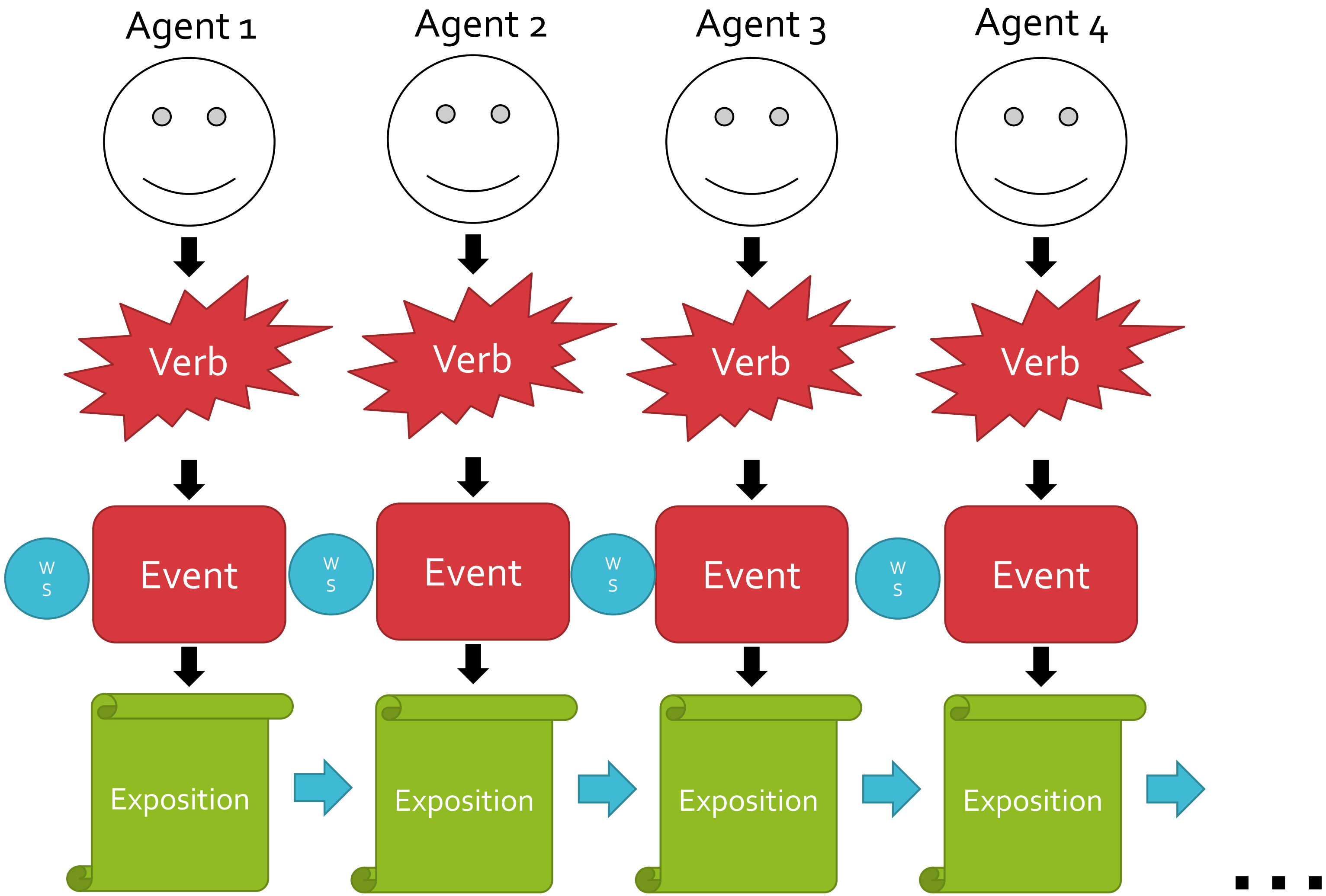
With world state hidden, the interactive story can become a series of expositions on events. This creates a story like traditional linear narratives.



Interaction with the Story



The Interactive Story



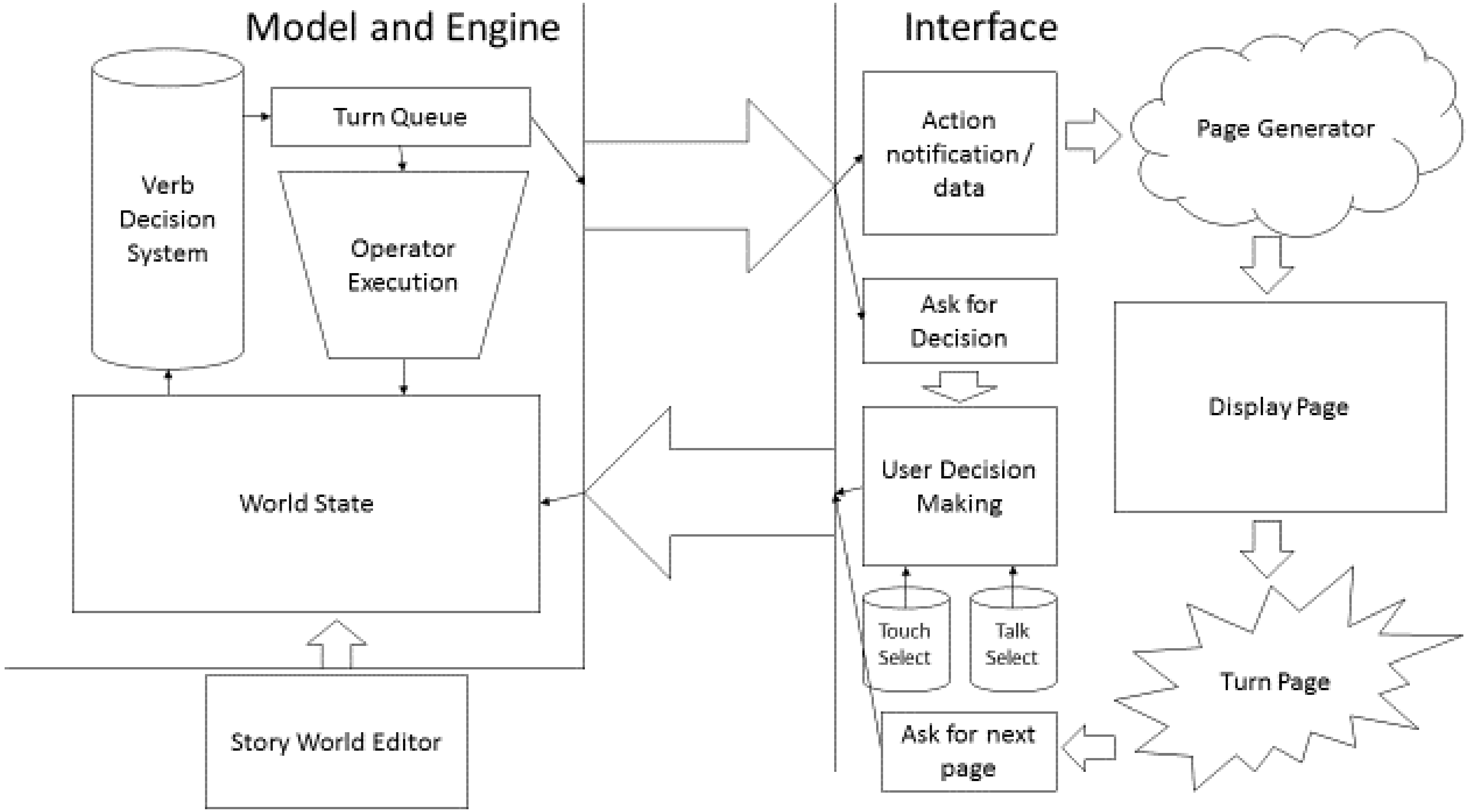
The Implementation of LudoNarrare

Details and Features

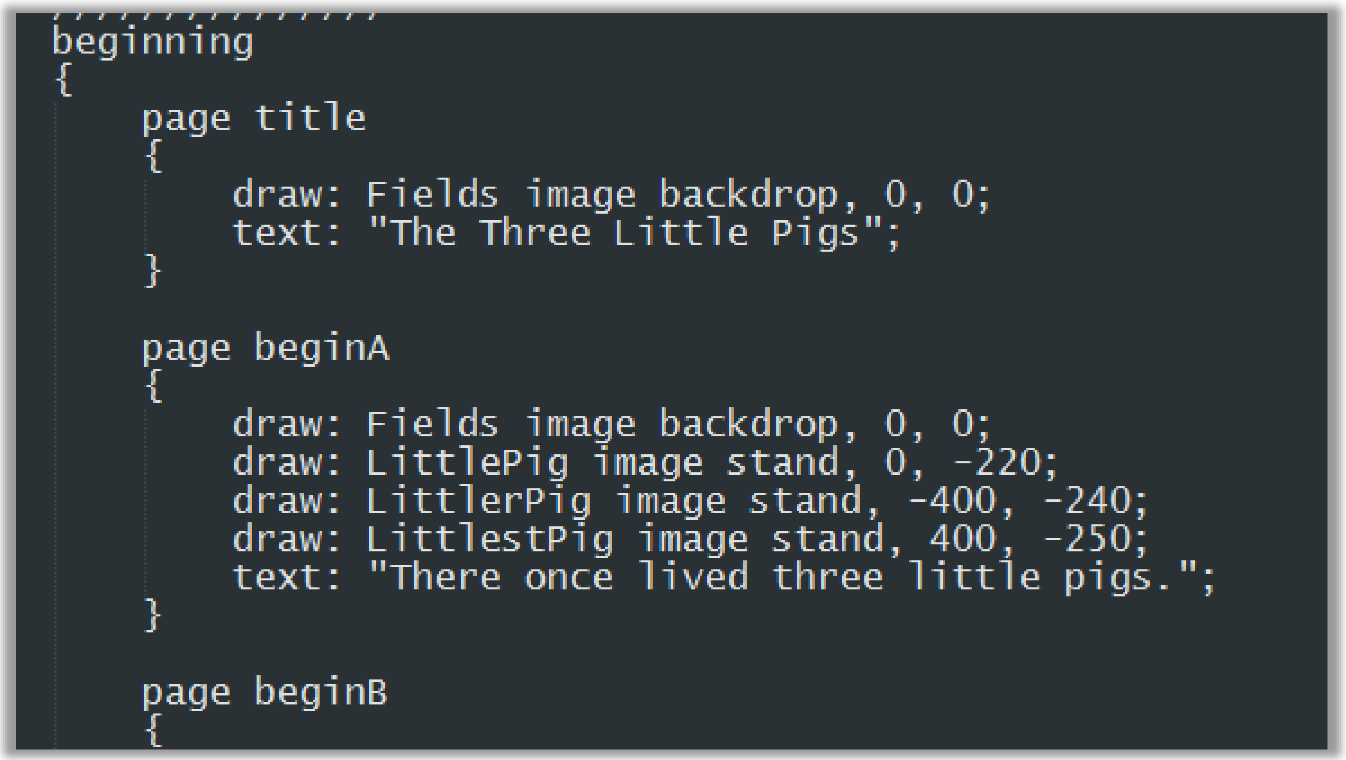
LudoNarrare Implementation

- Uses Unity Game Engine; programmed in C#.
- Has custom language for writing storyworlds; LNScript.
- The interface between the player and the story is a picture book.
- AI agents must be programmed separately as C# classes.

LudoNarrare Engine Loop

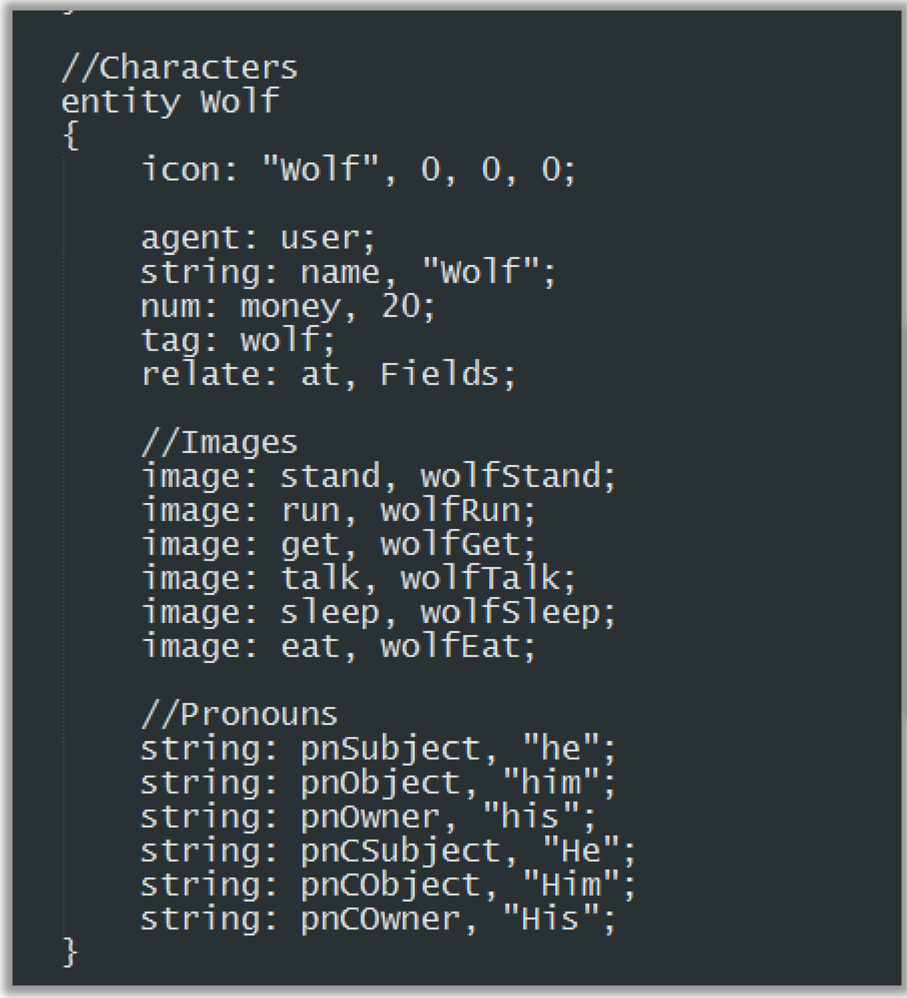


Beginning Definition in LNScript

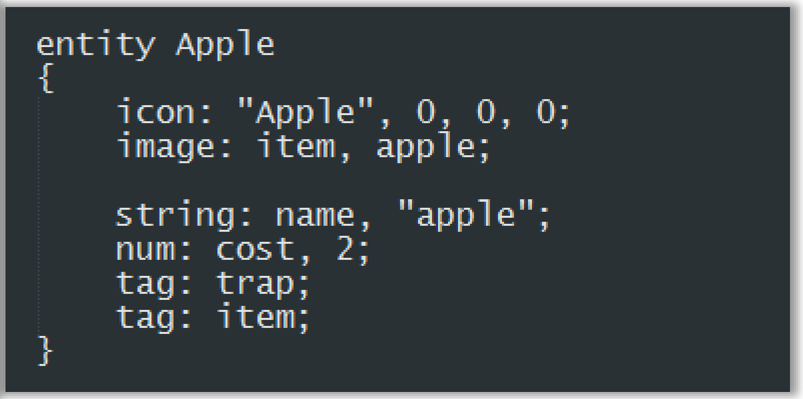


```
///////////////
beginning
{
    page title
    {
        draw: Fields image backdrop, 0, 0;
        text: "The Three Little Pigs";
    }

    page beginA
    {
        draw: Fields image backdrop, 0, 0;
        draw: LittlePig image stand, 0, -220;
        draw: LittlerPig image stand, -400, -240;
        draw: LittlestPig image stand, 400, -250;
        text: "There once lived three little pigs.";
    }

    page beginB
    {
```

Entity Definition in LNScript

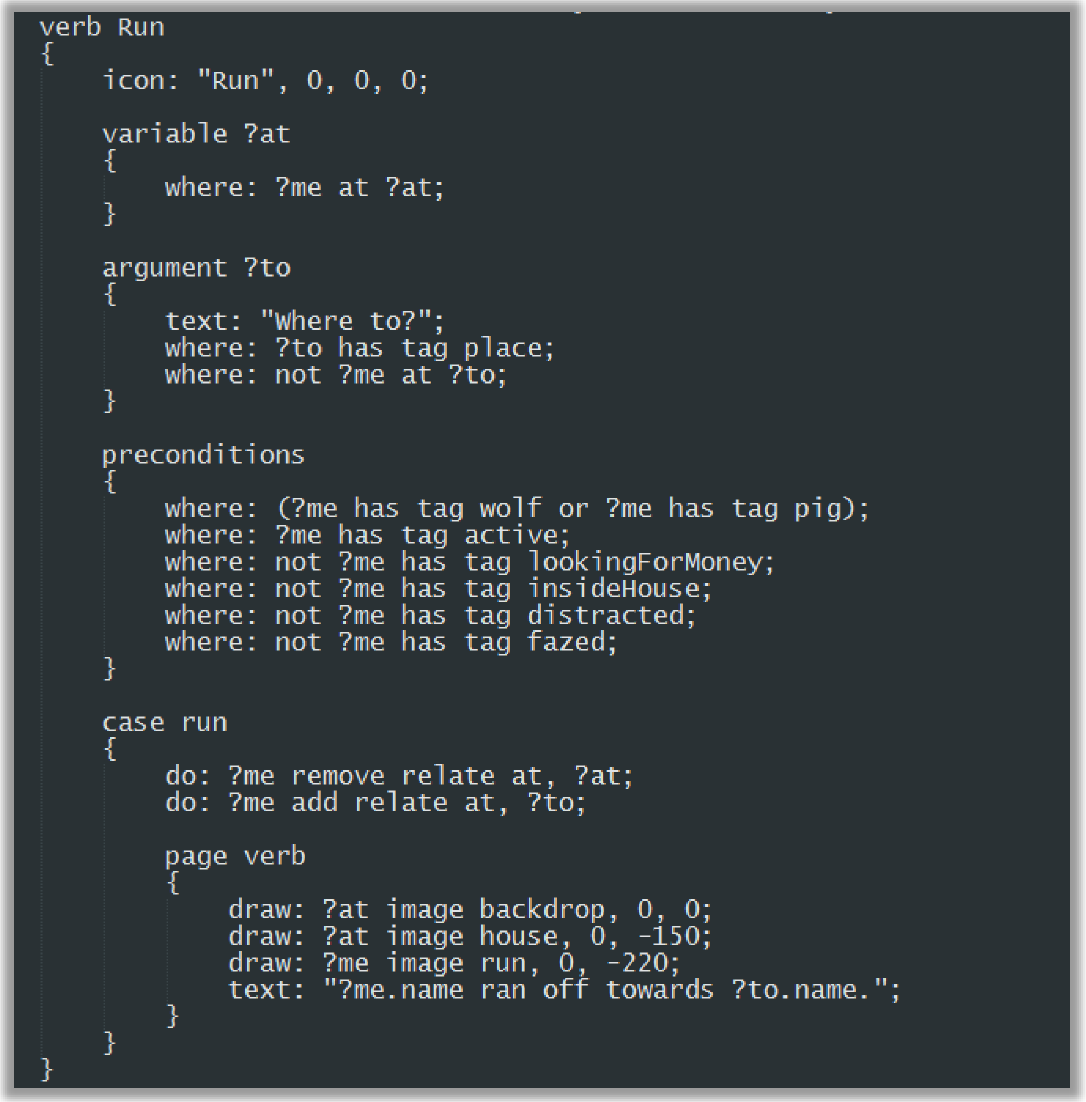


```
//Characters
entity Wolf
{
    icon: "Wolf", 0, 0, 0;

    agent: user;
    string: name, "Wolf";
    num: money, 20;
    tag: wolf;
    relate: at, Fields;

    //Images
    image: stand, wolfStand;
    image: run, wolfRun;
    image: get, wolfGet;
    image: talk, wolfTalk;
    image: sleep, wolfSleep;
    image: eat, wolfEat;

    //Pronouns
    string: pnSubject, "he";
    string: pnObject, "him";
    string: pnOwner, "his";
    string: pnCSubject, "He";
    string: pnCObject, "Him";
    string: pnCOwner, "His";
}
```



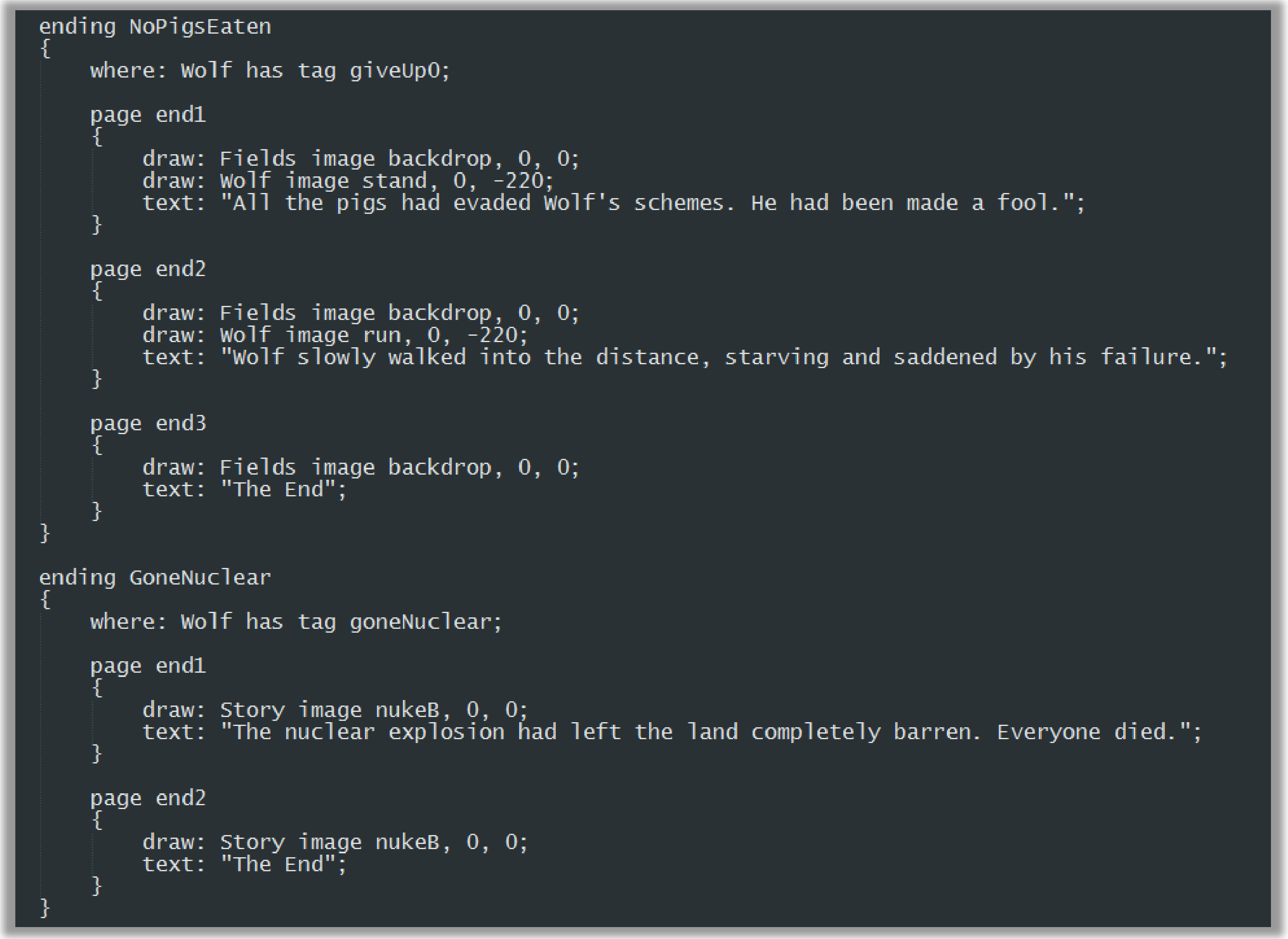
```
entity Apple
{
    icon: "Apple", 0, 0, 0;
    image: item, apple;

    string: name, "apple";
    num: cost, 2;
    tag: trap;
    tag: item;
}
```

Verb Definition in LNScript



```
verb Run
{
    icon: "Run", 0, 0, 0;

    variable ?at
    {
        where: ?me at ?at;
    }

    argument ?to
    {
        text: "Where to?";
        where: ?to has tag place;
        where: not ?me at ?to;
    }

    preconditions
    {
        where: (?me has tag wolf or ?me has tag pig);
        where: ?me has tag active;
        where: not ?me has tag lookingForMoney;
        where: not ?me has tag insideHouse;
        where: not ?me has tag distracted;
        where: not ?me has tag fazed;
    }

    case run
    {
        do: ?me remove relate at, ?at;
        do: ?me add relate at, ?to;

        page verb
        {
            draw: ?at image backdrop, 0, 0;
            draw: ?at image house, 0, -150;
            draw: ?me image run, 0, -220;
            text: "?me.name ran off towards ?to.name.";
        }
    }
}
```

Ending Definition in LNScript

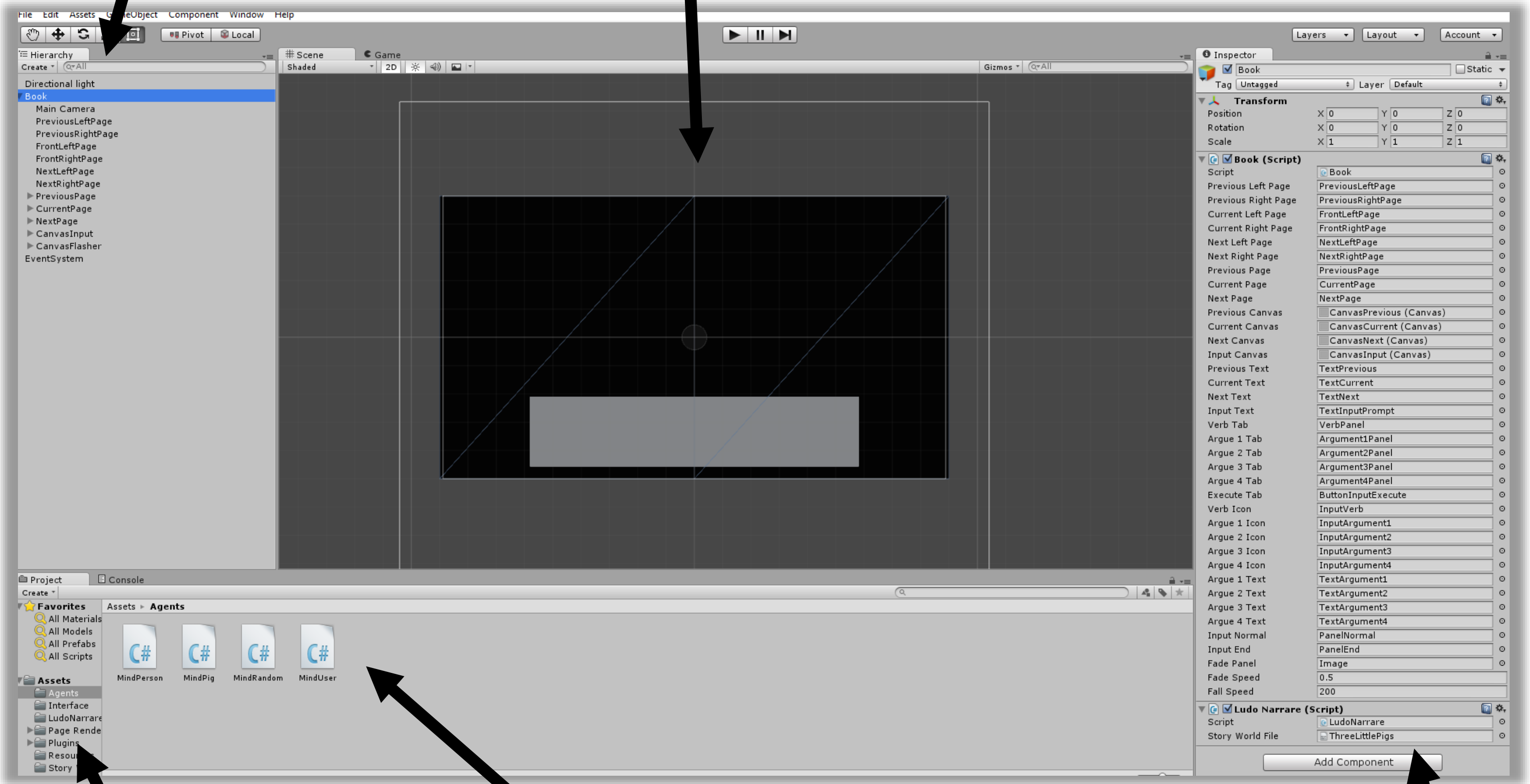


```
ending NoPigsEaten
{
    where: Wolf has tag giveUp0;

    page end1
    {
        draw: Fields image backdrop, 0, 0;
        draw: Wolf image stand, 0, -220;
        text: "All the pigs had evaded Wolf's schemes. He had been made a fool.";
    }

    page end2
    {
        draw: Fields image backdrop, 0, 0;
        draw: Wolf image run, 0, -220;
        text: "Wolf slowly walked into the distance, starving and saddened by his failure.";
    }

    page end3
    {
        draw: Fields image backdrop, 0, 0;
        text: "The End";
    }
}

ending GoneNuclear
{
    where: Wolf has tag goneNuclear;

    page end1
    {
        draw: Story image nukeB, 0, 0;
        text: "The nuclear explosion had left the land completely barren. Everyone died.";
    }

    page end2
    {
        draw: Story image nukeB, 0, 0;
        text: "The End";
    }
}
```

Unity for LudoNarrare

Components of the book and the interface.

3D model of story book.



Book ☑ · □ Static

Tag: Untagged · Layer: Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Book (Script)

Script	Book
Previous Left Page	PreviousLeftPage
Previous Right Page	PreviousRightPage
Current Left Page	FrontLeftPage
Current Right Page	FrontRightPage
Next Left Page	NextLeftPage
Next Right Page	NextRightPage
Previous Page	PreviousPage
Current Page	CurrentPage
Next Page	NextPage
Previous Canvas	CanvasPrevious (Canvas)
Current Canvas	CanvasCurrent (Canvas)
Next Canvas	CanvasNext (Canvas)
Input Canvas	CanvasInput (Canvas)
Previous Text	TextPrevious
Current Text	TextCurrent
Next Text	TextNext
Input Text	TextInputPrompt
Verb Tab	VerbPanel
Argue 1 Tab	Argument1Panel
Argue 2 Tab	Argument2Panel
Argue 3 Tab	Argument3Panel
Argue 4 Tab	Argument4Panel
Execute Tab	ButtonInputExecute
Verb Icon	InputVerb
Argue 1 Icon	InputArgument1
Argue 2 Icon	InputArgument2
Argue 3 Icon	InputArgument3
Argue 4 Icon	InputArgument4
Argue 1 Text	TextArgument1
Argue 2 Text	TextArgument2
Argue 3 Text	TextArgument3
Argue 4 Text	TextArgument4
Input Normal	PanelNormal
Input End	PanelEnd
Fade Panel	Image
Fade Speed	0.5
Fall Speed	200

Ludo Narrare (Script)

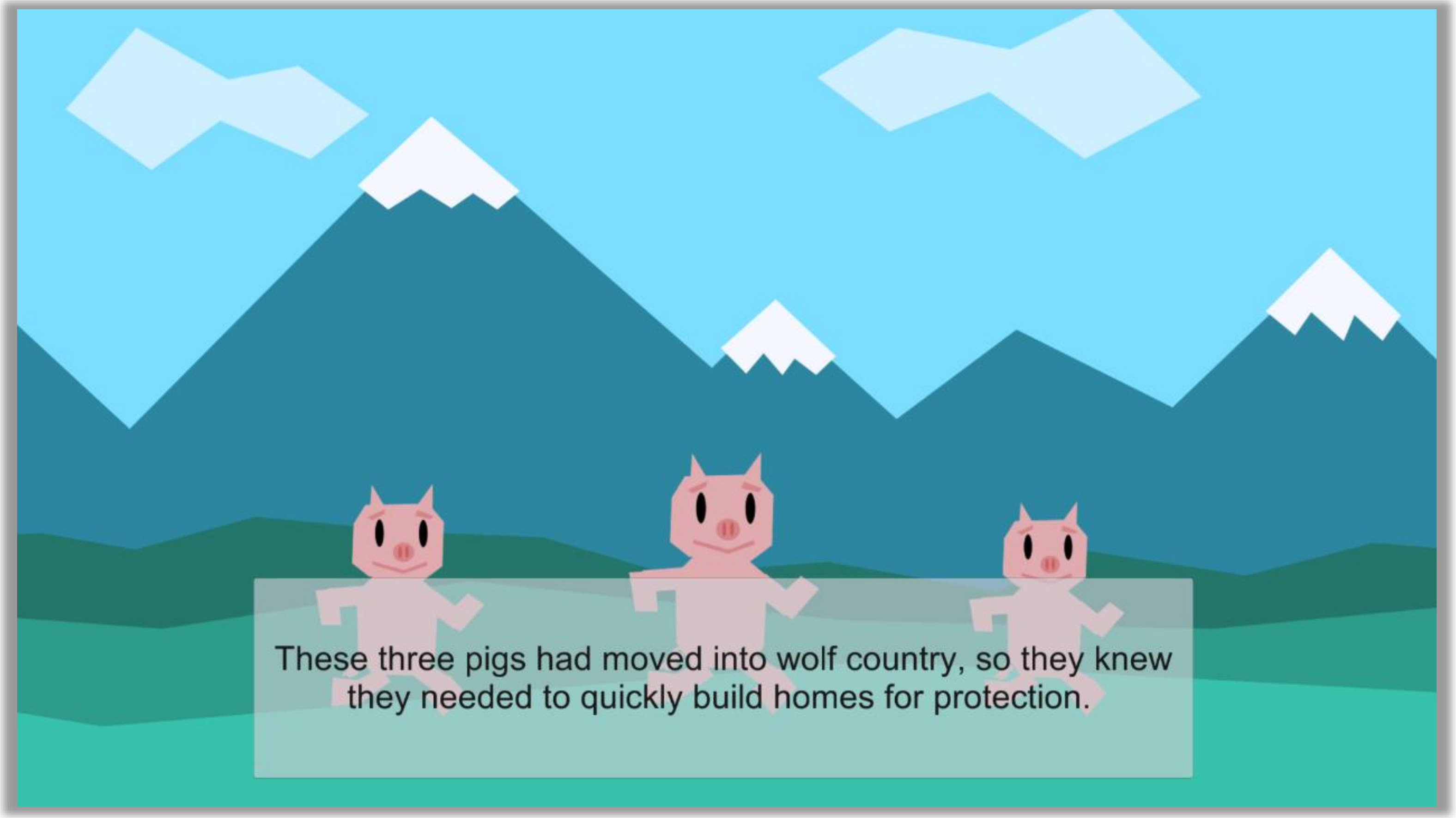
Script	LudoNarrare
Story World File	ThreeLittlePigs

Resource folders store storyworld definition file and images.

Agents coded as C# classes.

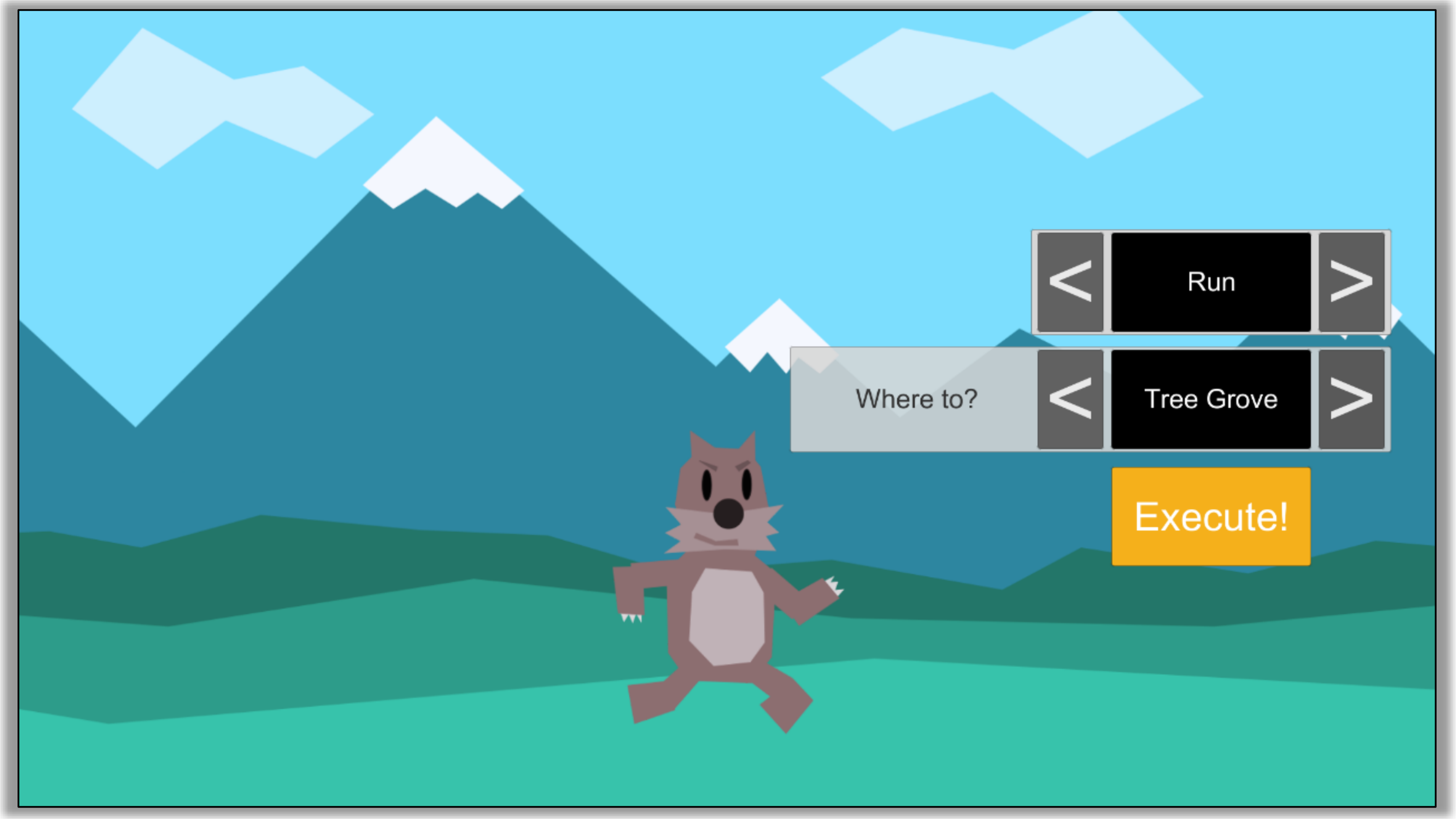
Storyworld file loaded as an argument to the LudoNarrare Engine.

LudoNarrare Interactive Story Exposition Interface



These three pigs had moved into wolf country, so they knew they needed to quickly build homes for protection.

LudoNarrare Interactive Story Decision Interface



Assessment of LudoNarrare as a Solution

LudoNarrare’s Effectiveness from Multiple Perspectives

Does LudoNarrare Solve the Interactive Storytelling Problem?

- + Generates stories with player input
- + Verbs have meaningful effect on the world
- + Events can be told expressively, leading to storytelling
- - No system for ensuring that events follow good narrative flow

Does LudoNarrare Distinguish Itself from Past Solutions?

- + Move away from a precise spatial world model separates LudoNarrare from video games and virtual worlds
- + LudoNarrare is designed specifically with explicit interactive storytelling in mind
- + LudoNarrare focuses on decisions and not reflexes or puzzle solving
- - Both LudoNarrare and video games tell stories using the same verb-event-exposition model

Does LudoNarrare Create Compelling Experiences?

- This can only be answered in practice. The *Three Little Pigs* demo was created to start answering this question.

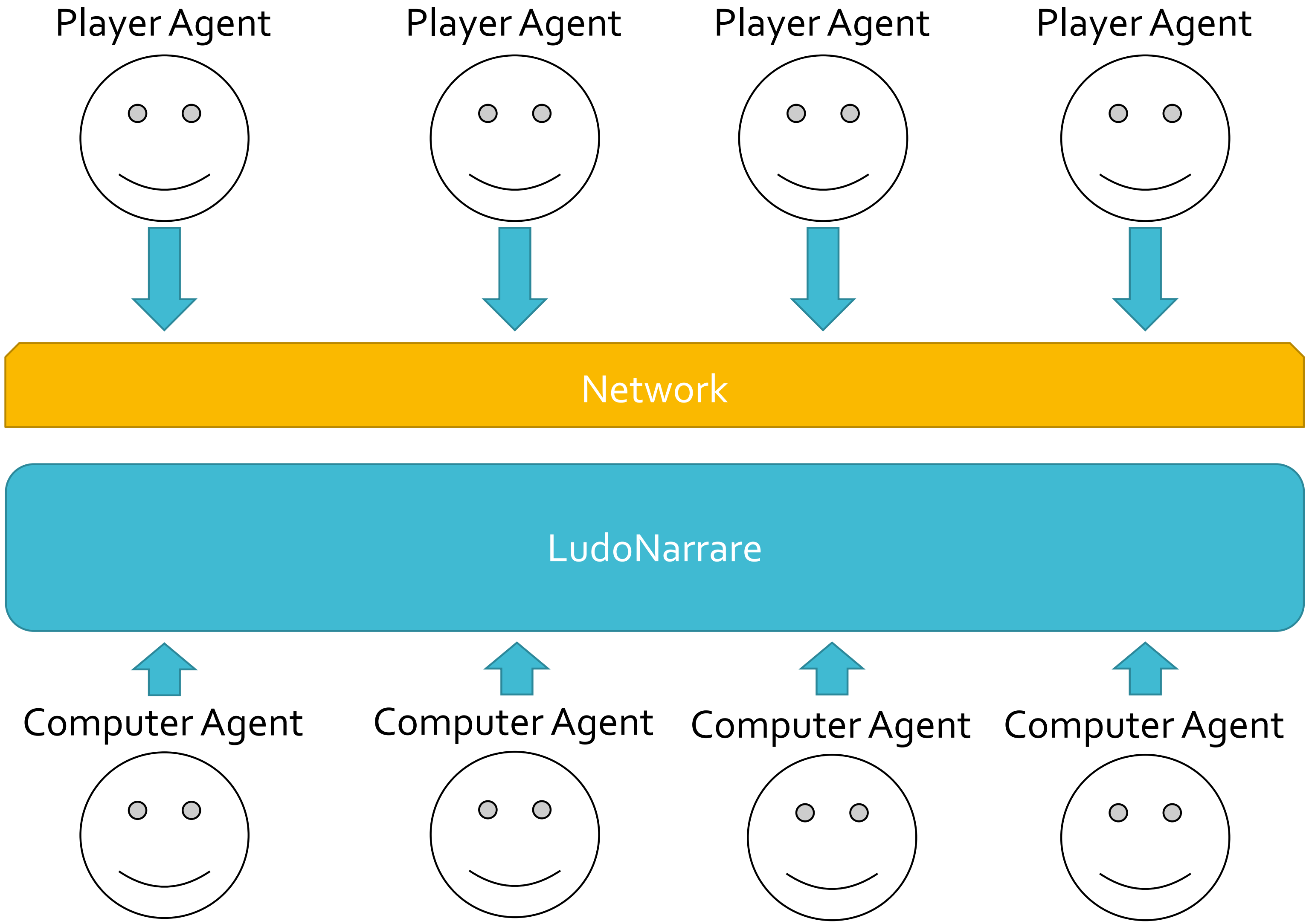
Going Forward

How to Build Upon the Knowledge Gained from LudoNarrare

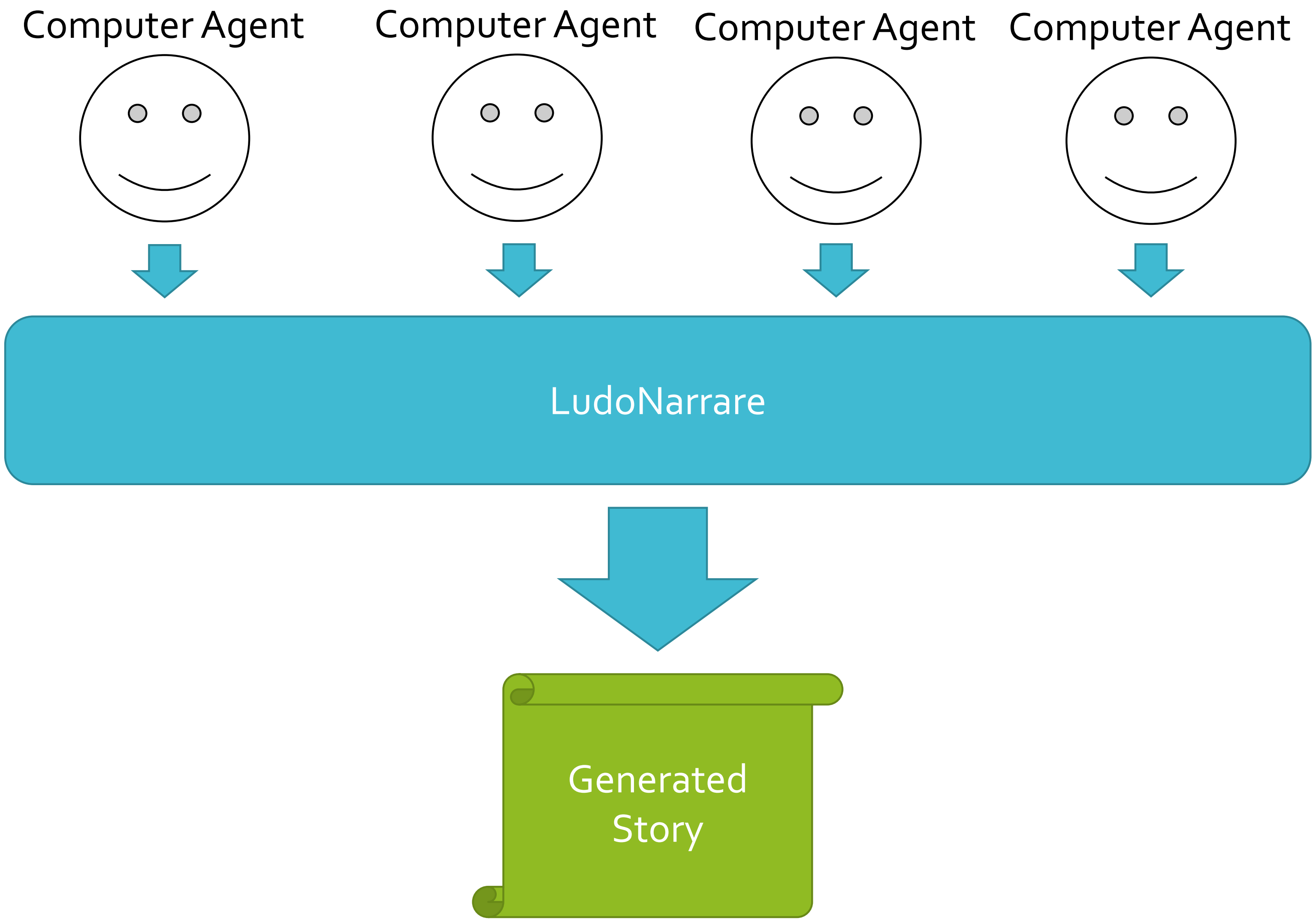
Expanding from LudoNarrare

- Improve the tools; move from LNScript to a GUI software tool to make it easier for non-programmers to author storyworlds.
- Create a more robust system for defining conditions and operators of verbs.
- Significantly improve performance.
- Write code for more interesting computer agents.
- Design system that ensures events are ordered in a narratively compelling way.
- Expand the method of turning events into exposition to be less static and more dynamic.

Multiplayer Story Worlds

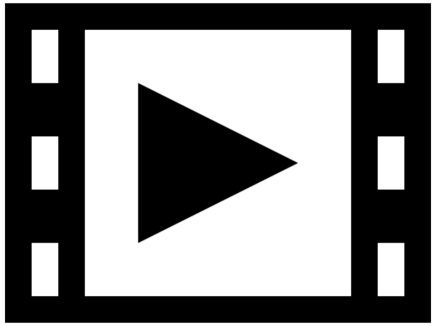


Story Generators



Interfaces Beyond the Picture Book

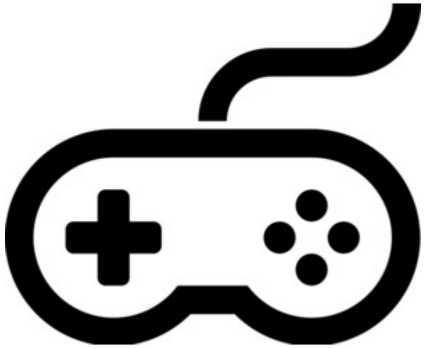
Film



Theater

Music

Video Games



Demonstration

The Three Little Pigs