Comparing First Generation Drama Engines

What is a “Drama Engine”?

- An engine capable of interpreting domain language code to support interaction with dramatically interesting characters
- A way to creatively author character AI
- A system that supports the self-organization of social contexts in the player’s mind, reflected to some degree in the game
- The artistic and commercial future of play

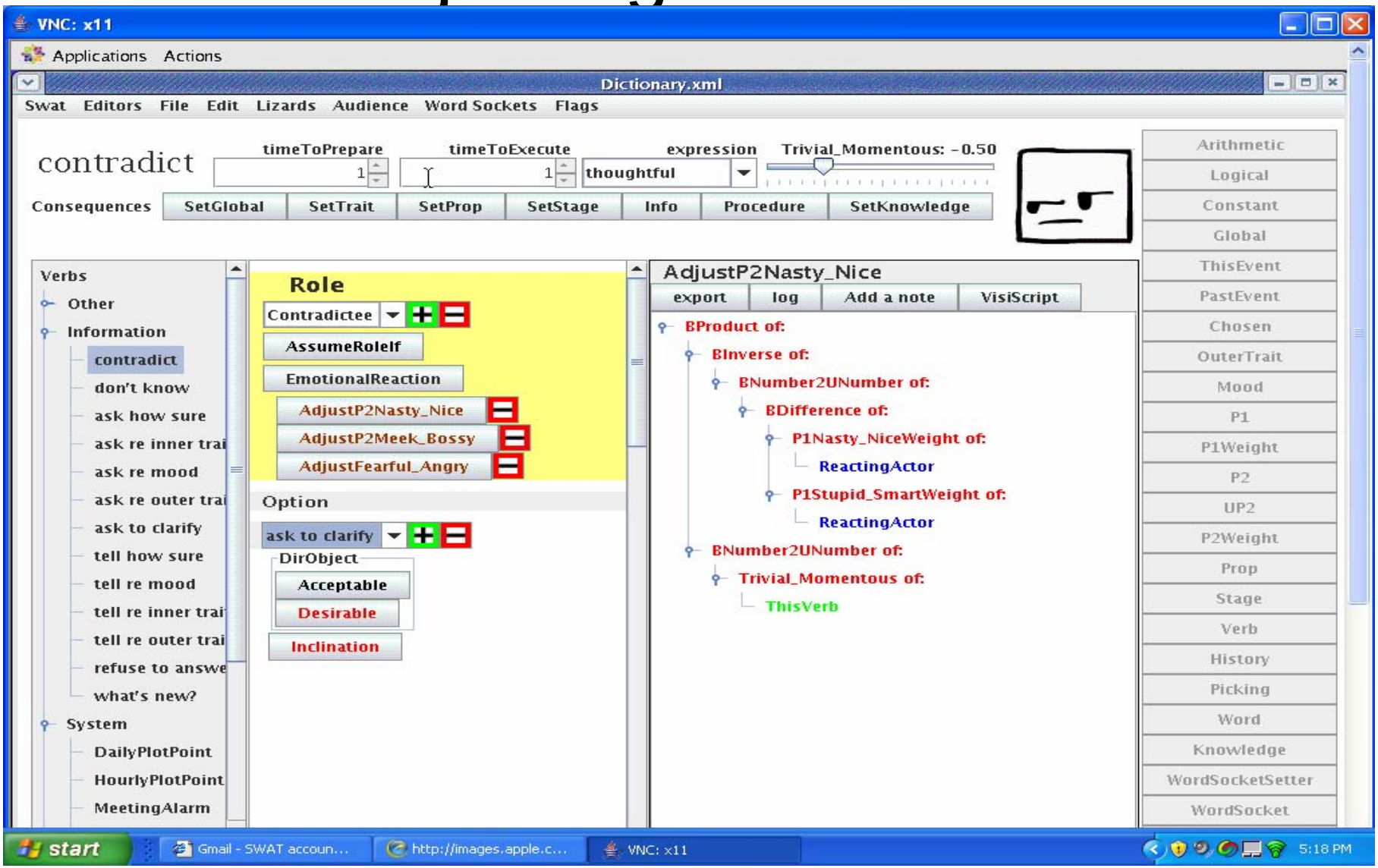
Approaches To A Wicked Problem

Engine Designs	Homogenous Characters	Heterogeneous Characters
Generative System	*Storytron*	*Façade's Architecture*
Emergent System	*Drama Princess*	*Rocket Heart*

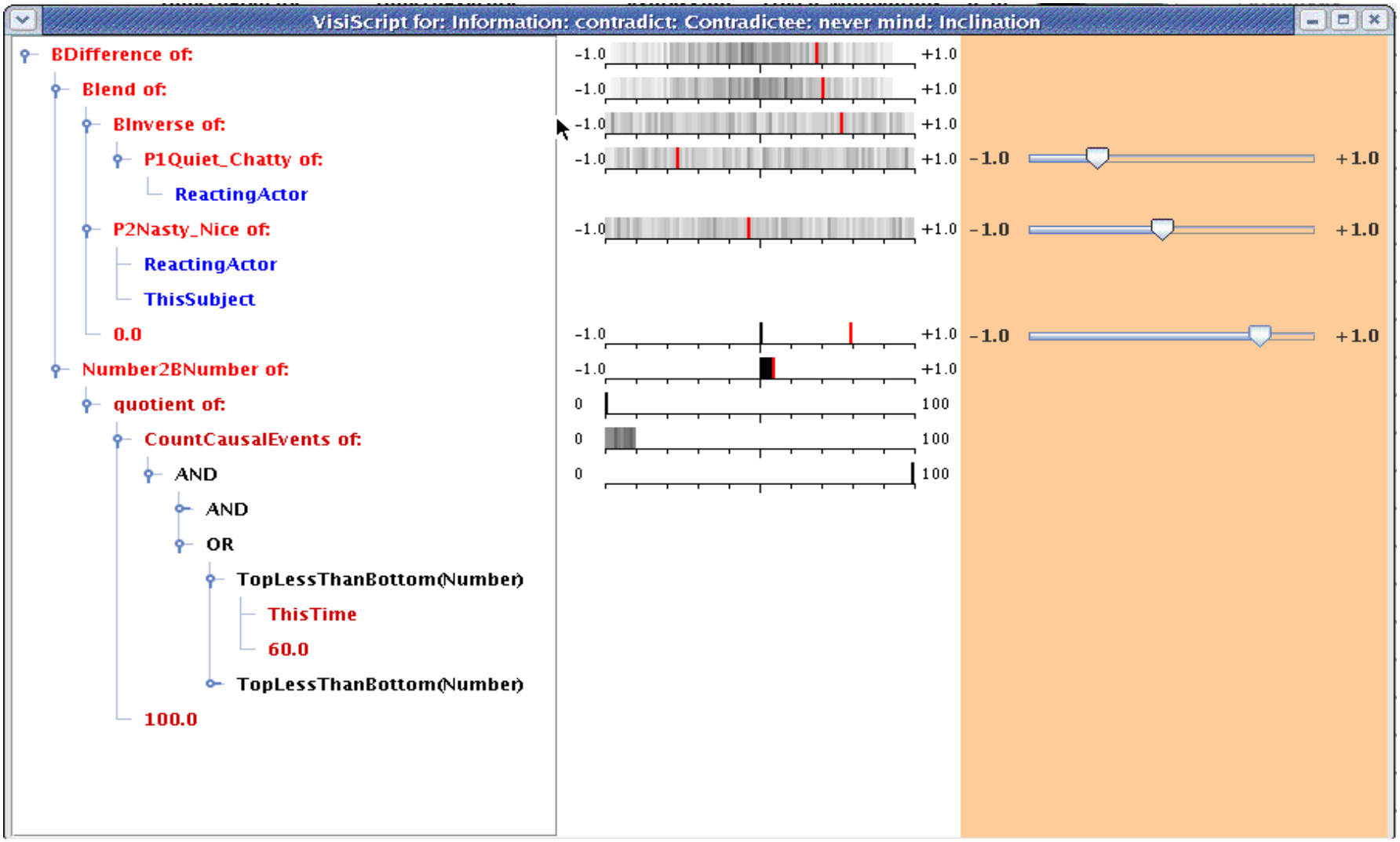
Storytron

- World is composed of sentences in toy language, together they “generate” a story
- Linguistic “Deikto” Interface – Turn-Based
- Business model based on subscriptions
- No animation, no spatial relationships
- Static facial feedback
- Special actor “Fate” manages discourse
- Characters are defined by bounded (-1,1) floats, traits are constant, perception of traits is variable

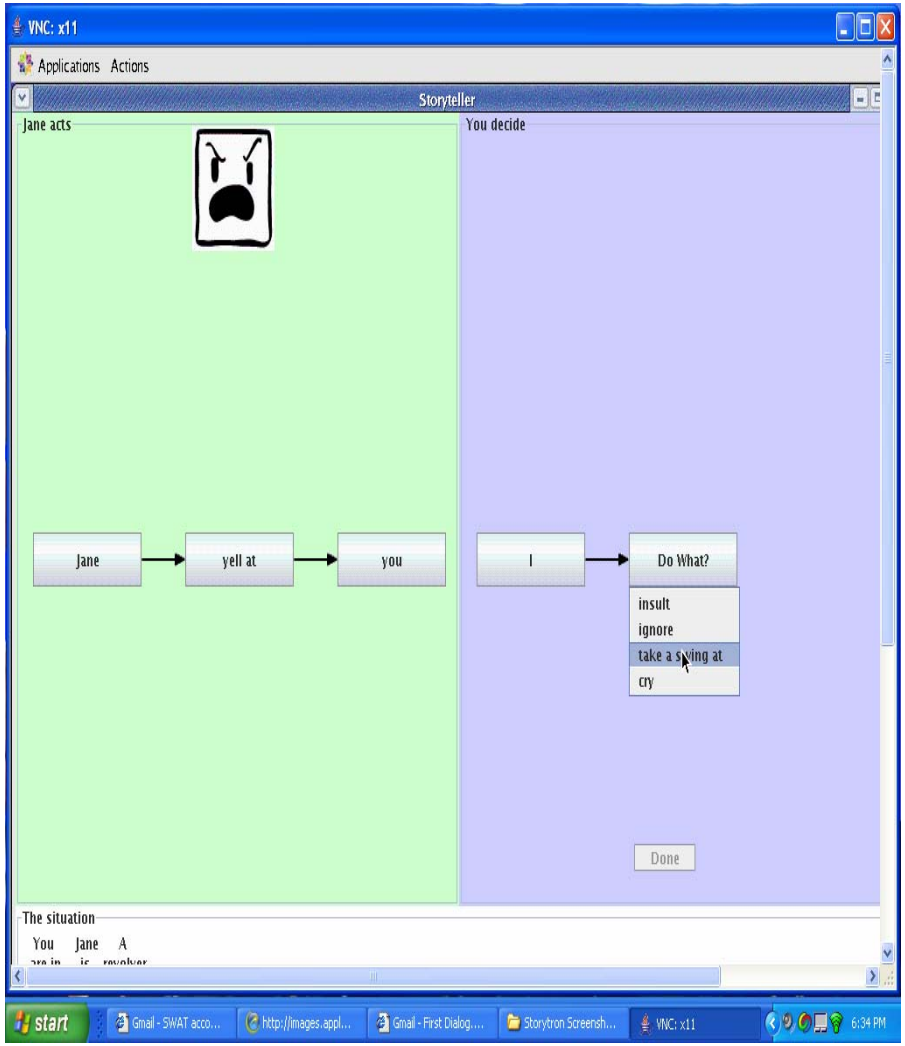
Reacting To Verbs Involves Adjusting Variables

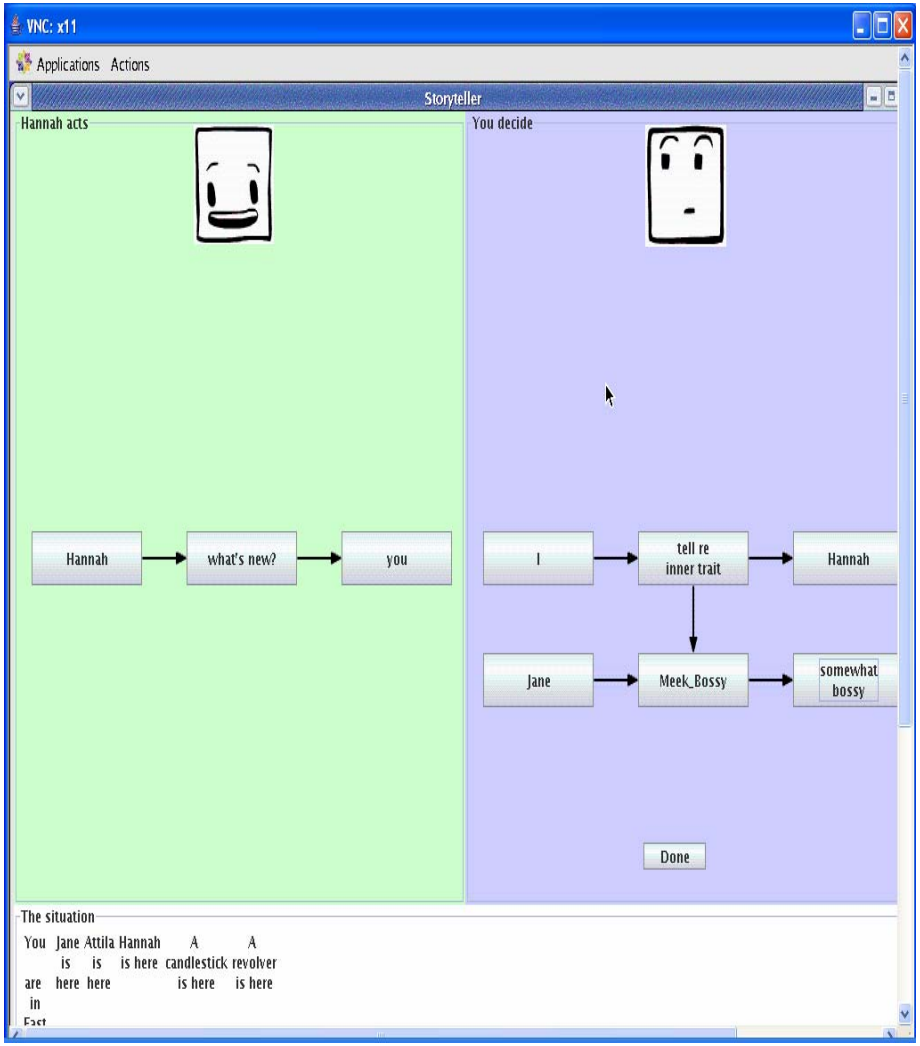


Every Way Of Reacting To A Given Verb Is Weighed By Inclination Equations The Author Designs, a Character's Personality Traits are Taken as Input

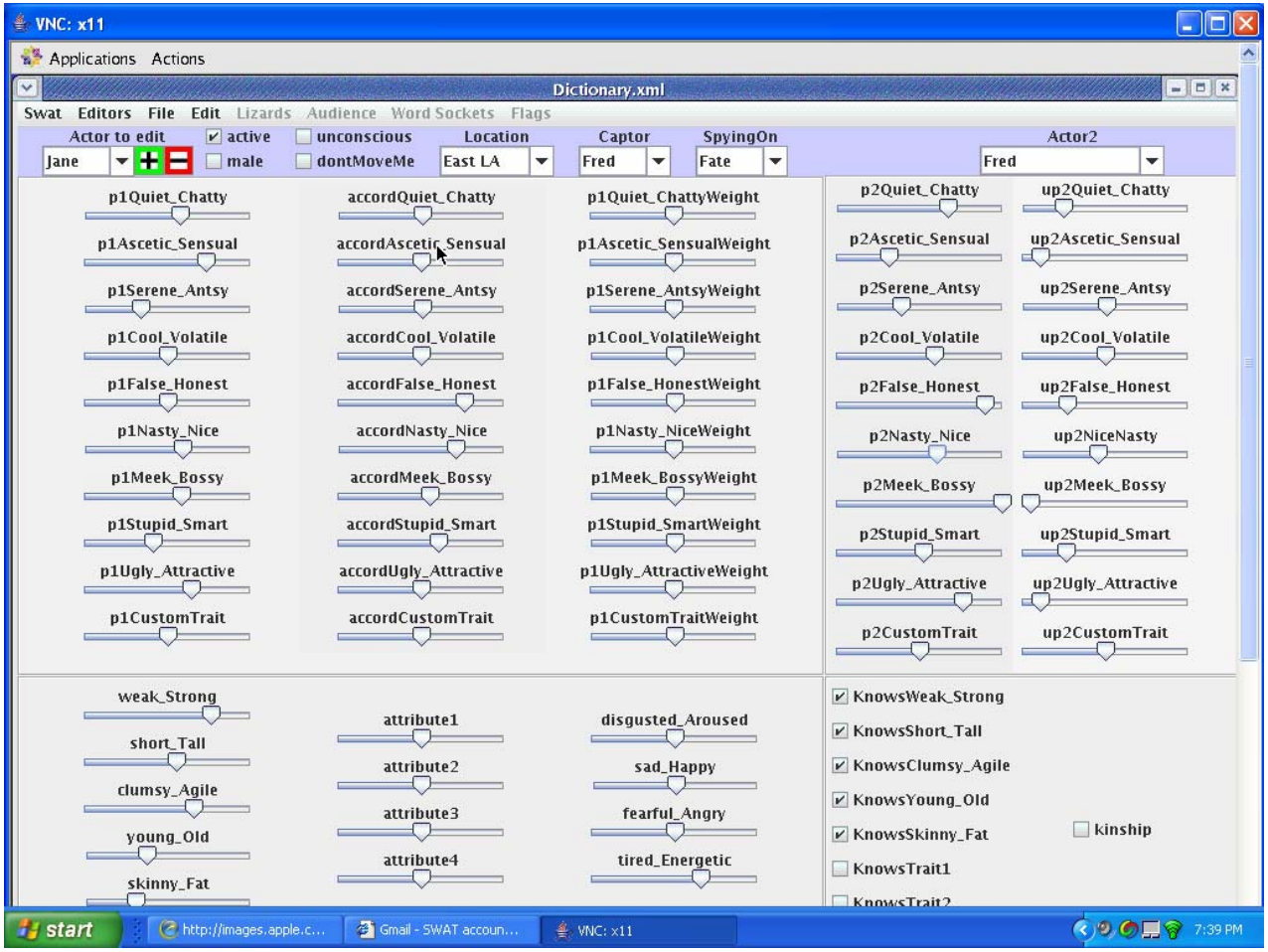


Jane Rejects Fred's Advances, Violence and Gossip Result

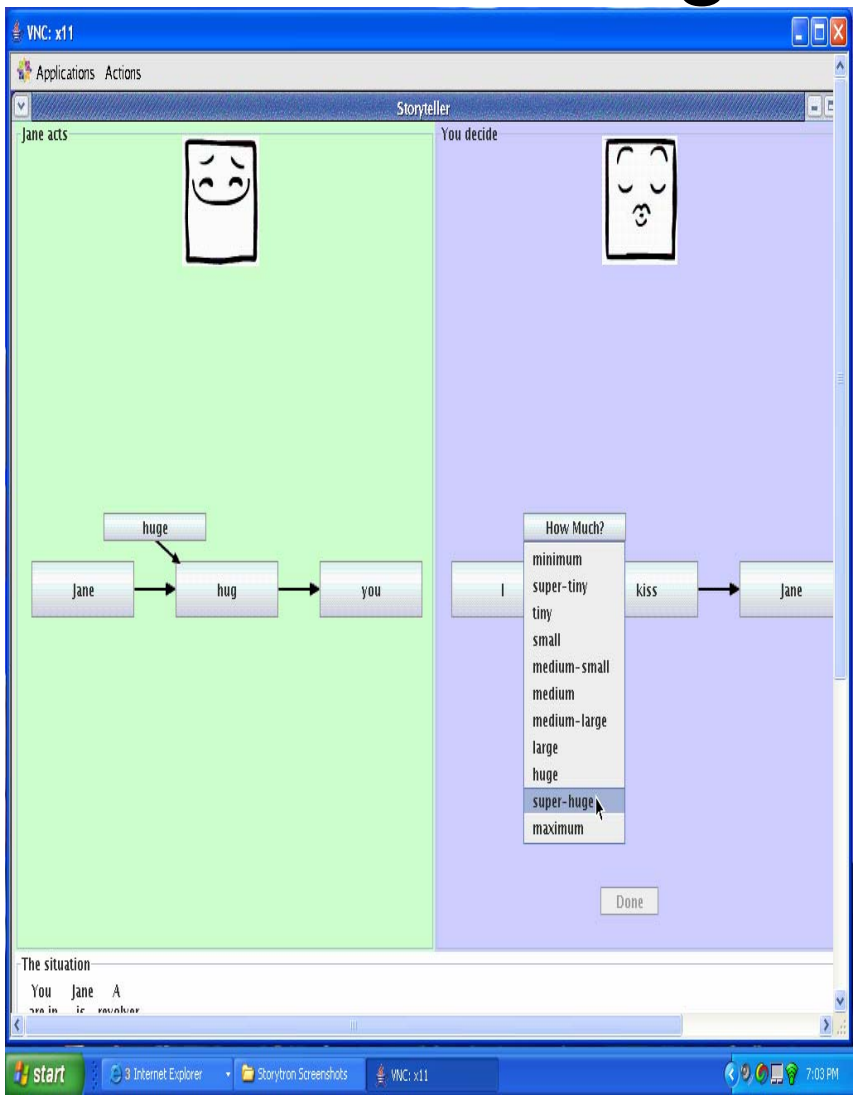


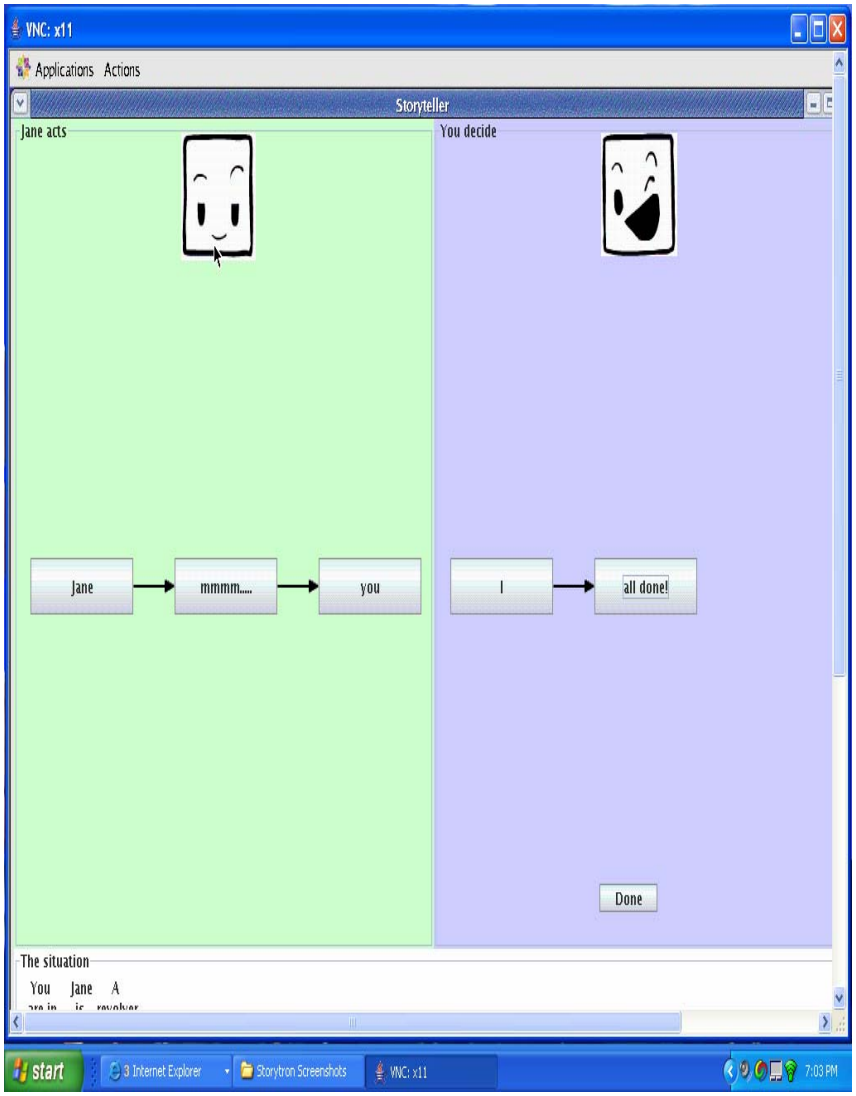


I Want To Get In Jane's Head, So I Open Her Up In The Editor



After Adjusting Her Opinion Of Fred Things Go Better





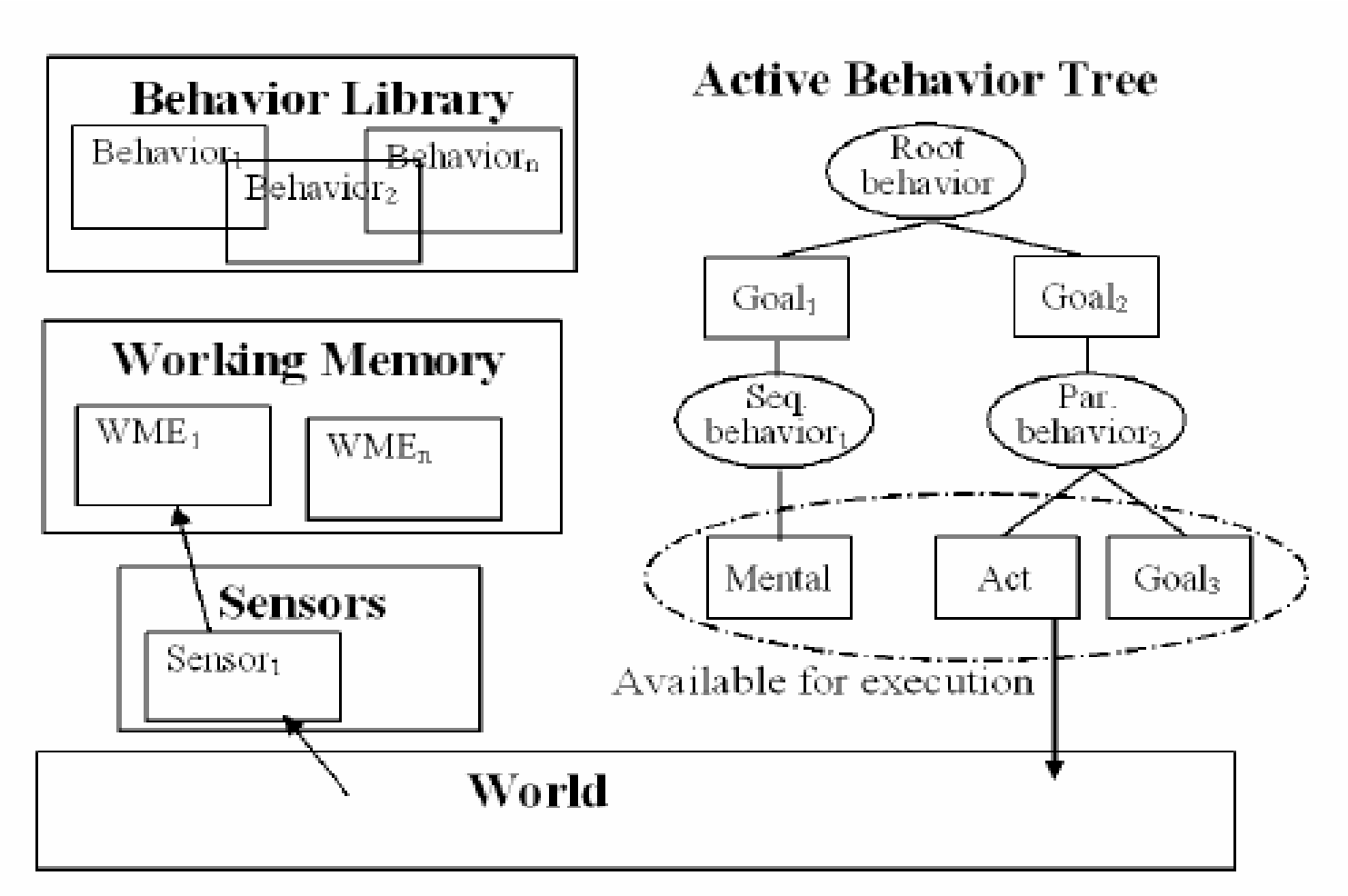
Storytron Limitations

- Each verb requires a minimum of five scripts per Boolean role, plus one script per variable adjusted and three or five scripts per reaction option
- Content demands tend toward limited Local Agency, but powerful Global Agency due to generative recombination
- Balancing and debugging potentially staggering due to highly coupled nature of content
- Graphically spare front-end limits feedback

Façade's Architecture

- A suite of modules and languages
- A high-level "Beat" authoring language
- A language for global "Mix-in" behaviors
- A language for local "Joint Dialogue" behaviors
- A language parser build on top of JESS
- A Drama Manager that uses probability at a high level to sequence "Beats"

A Behavioral Language: Core Content Creation Tool




```
sequential behavior OpenDoor() {
  precondition {
    (KnockWME doorID :: door)
    (PosWME spriteID == door pos :: doorPos)
    (PosWME spriteID == me pos :: myPos)
    (Util.computeDistance(doorPos, myPos) > 100)
  }
  specificity 2;
  // Too far to walk, yell for knocker to come in
  subgoal YellAndW....ForGuestToEnter(doorID);
}

sequential behavior OpenDoor() {
  precondition { (KnockWME doorID :: door) }
  specificity 1;
  // Default behavior - walk to door and open
}
```

Relative Proximity And Facial Expressions Give Feedback On Joint Dialogue Behaviors

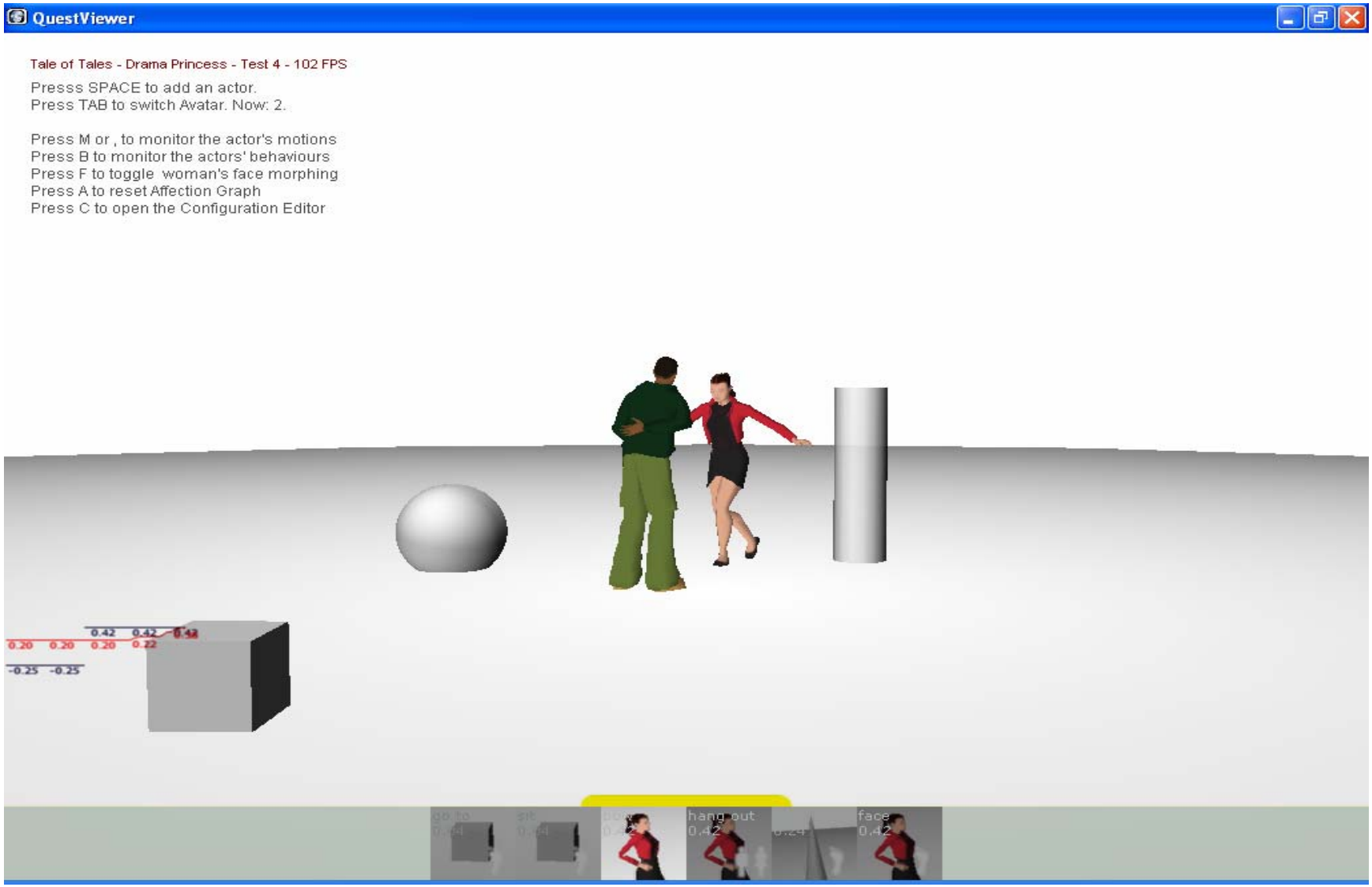
Façade’s Architecture's Limitations

- Learning curve for ABL is steep, authoring environment may improve this
- Content is oriented toward rich Local Agency but limited Global Agency, more Beats may balance this by increasing number of generative recombinations
- Input and feedback are not always clear, due to limitations of language parsing and system ambiguity
- Actors attempts to play off out-of-character inputs rely on a complaint player

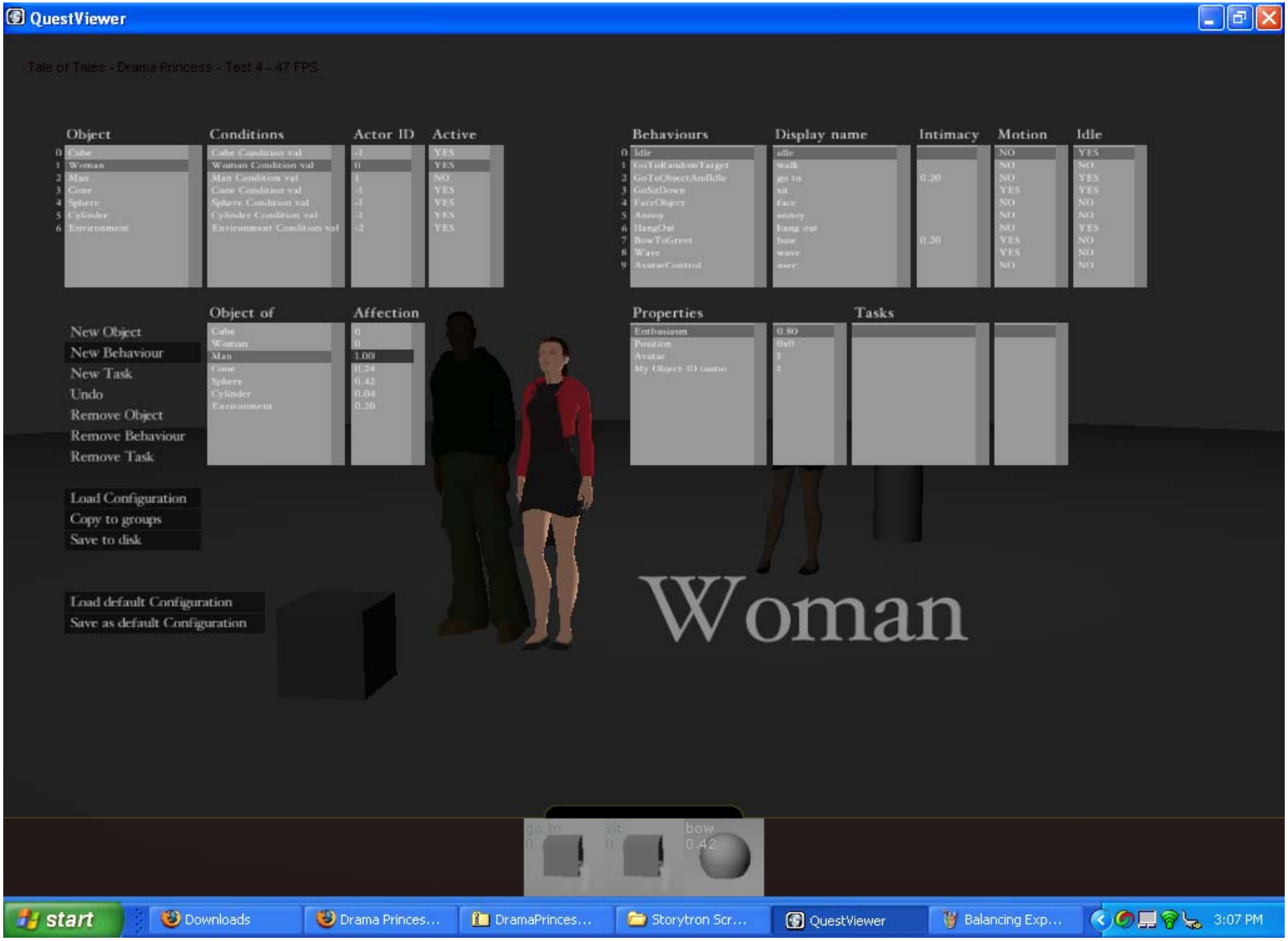
Drama Princess

- Behavior and Animation Module for *144,* a retelling of Little Red Riding Hood
- Actors are “empty shells” driven by objects in environment and characterized by a single variable, *enthusiasm*, which weighs changes in *affection* for objects
- Lack of feedback intentional to emphasize player interpretation
- Filters on available behaviors (distance, condition, intimacy, affection) limit actions to a particular context
- Uses probability at a low level to mix behaviors

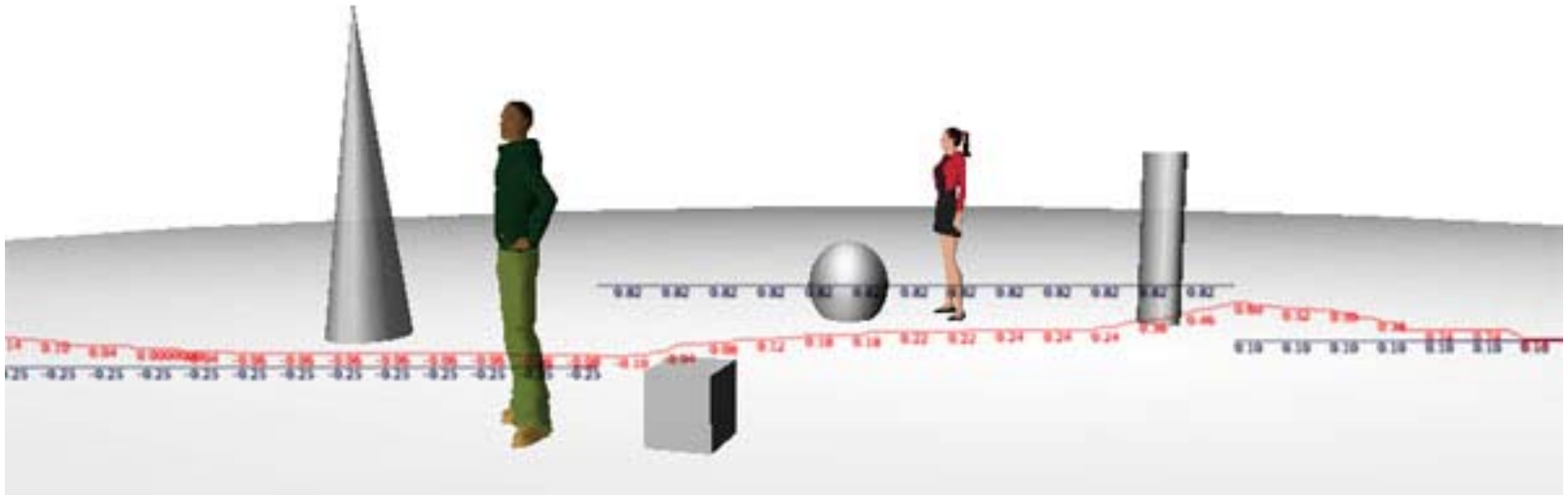
Affection Rises



Adjusting Enthusiasm



Relationships Evolve Over Time



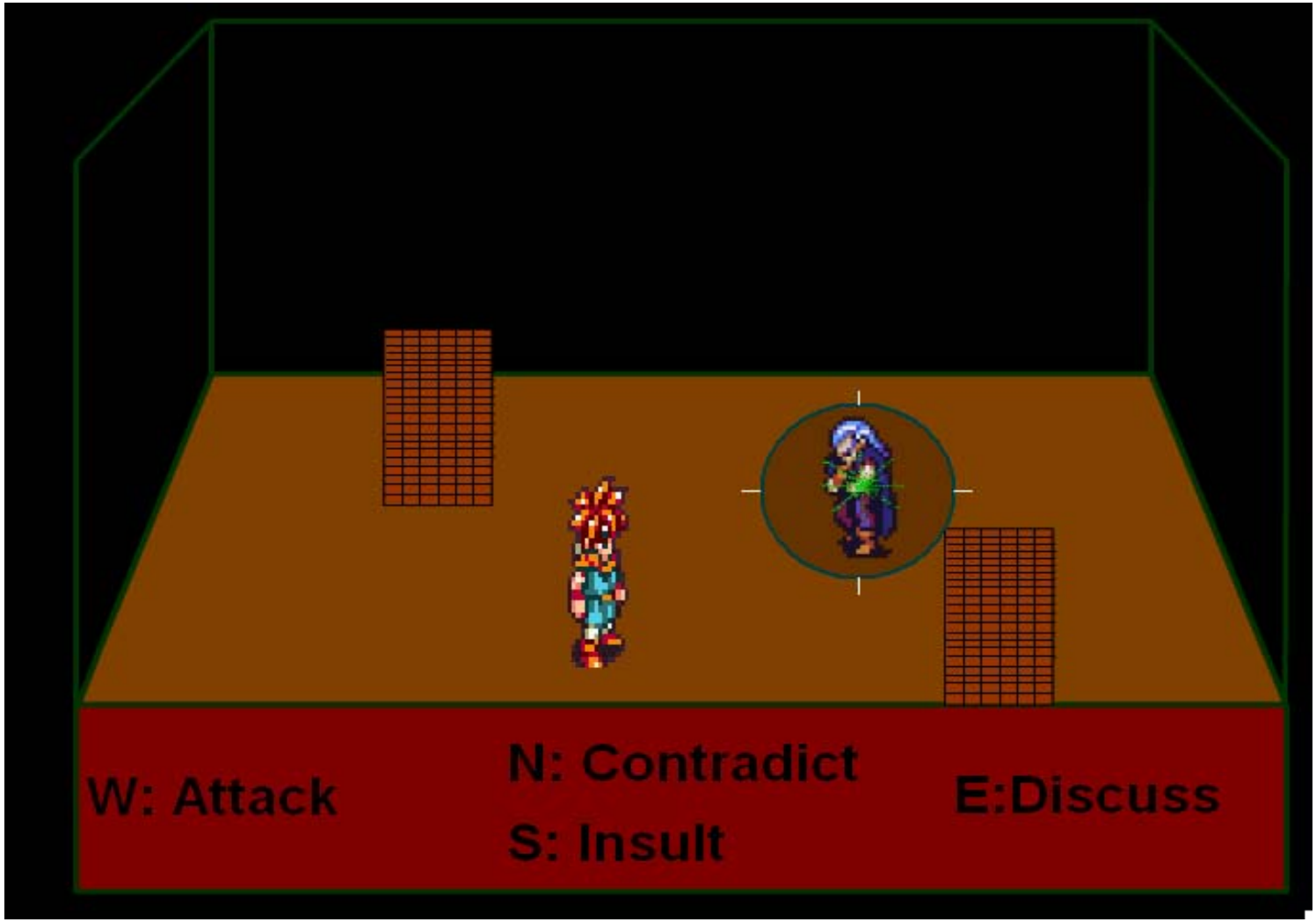
Limitations of Drama Princess

- Oriented toward free-form play, can’t support goals or strategy
- Randomness implies a lack of control, which implies a short-live novelty and little re-play value, though casual players may enjoy it occasionally to relax
- Lack of behavioral complexity and language implies shallow social play
- Little to no Global Agency

Rocket Heart

- Characters behave in procedurally unique ways and have unique relationships in a goal context
- Designed to nest inside more traditional forms of gameplay and interface with game engine
- Game actions can alter code for character's relationship actions
- Actions are conducted in context of relationship, but can mismatch to dramatic effect
- Hinges on carefully arranged, cascading feedback, based on periodic shifts in relationships and reacting actions
- Tactile interface provides context-sensitive options

An Argument Between Rivals



Anna: A Shy Romantic

```
relationship Romance "Bob" + 25
sparkmod Romance * 0.5
watch Romance 3h
{
        action "romance"
}
watch Relations "command" "Bob"
{
        mood + 25
        do event
}
schedule
{
        ~learn "Mr. Tengumi"
        @T 1000 – 1400
}
schedule
{
        ~lunch "Bob"
        @T 1230 – 1300
}
schedule $romance
{
        ~romance "Bob"
        @P "Bob"
        # 5
}
```

```
relationship Friend "Coco" + 20
watch Work
{
        if (romancing)
                continue
        if (! this.exists)
                relationship Work + 10
}

watch Romance > 50
{
        kill this
        schedule $romance
        {
                ~romance target
                @P target
                # 25
        }
}
```

Rocket Heart Limitations

- Requires supporting gameplay for involving play complexity
- Characters tend toward over-the-top behavior, limiting potential aesthetics to Shogo-sequel theatrics, and potential audiences to the casual web and mobile spaces
- Balancing behaviors with game context and other actors is difficult
- Tends toward more Local than Global Agency, more strongly than Facade

Conclusions On The Outset Of The First Generation

- Emergent systems are much faster to create than Generative systems (five months versus five years)
- Procedurally unique characters richen Local Agency, while balancing and content limits constrain Global Agency, visa versa for homogenous, data-differentiated characters
- Generative systems can provide social play that stands on its own, while emergent systems are augmentative to more traditional forms of play
- Once a system has been established and tested, the time involved in producing content becomes much lower than spatial level design in traditional games
- Procedural and/or modular art assets make drama game development significantly cheaper than lineated games

What Would Entail A Second Generation Drama Engine?

- Clearly and thematically designed input schemes that are “consistently inconsistent” – matching feedback that can be inferred from character cues to a degree of ambiguity that is balanced with the game
- Leveraging of available forms of social feedback (language, gestures, relative proximity, facial expressions, posture) in a complementary suite
- Leveraging new forms of input to make context-sensitive verb choices more intuitive and less mechanical (e.g. Wii and PS3 motion sensors)
- Successful experimentation with background social simulation to cue characters based on a greater culture, or to allow the player to influence said culture indirectly (candidates include Boid’s algorithm, Memetic algorithm’s driven by authored heuristics)