# Invisible Dynamic Mechanic Adjustment in Virtual Reality Games

Justus Robertson[1], Rogelio E. Cardona-Rivera[2], and R. Michael Young[2]
[1]*Department of Theatre, Film, Television and Interactive Media, University of York, York, UK*
[2]*School of Computing and the Entertainment Arts and Engineering Program, University of Utah, UT, USA*
justus.robertson@york.ac.uk, {rogelio | young}@eae.utah.edu

***Abstract*—A key part of managing a player's virtual reality experience is ensuring that the environment behaves consistently to the player's interaction. In some instances, however, it is important to change how the world behaves—*i.e.* the world's simulation rules or *mechanics*—because doing so preserves the virtual environment's intended quality. Mechanics changes must be done carefully; if too overt, they may be perceivable and potentially thwart a player's sense of presence or agency.**

**This paper reports the result of a study, which demonstrates the widely-held but heretofore-untested belief that changing an environment's mechanics *without considering what the player knows* is visible to the player. The study's findings motivate the paper's second contribution: an automated method to perform *invisible* dynamic mechanics adjustment, which affords shifting a game's previously-established mechanics in a manner that is not perceivably inconsistent to players. This method depends on a knowledge-tracking strategy and two such strategies are presented: (1) a conservative one, relevant to a wide variety of virtual environments, and (2) a more nuanced one, relevant to environments that will be experienced via head-mounted virtual reality displays. The paper concludes with a variety of design-centered considerations for the use of this artificial intelligence system within virtual reality.**

## 1. Introduction

Virtual environments require significant human expertise to design and build [1] and as a result, they are brittle: they often provide the same, limited range of experience to all users. Because virtual worlds are synthetic and their interactions are entirely computer-mediated, there is potential for artificial intelligence (AI) systems to create and manage these virtual worlds to afford richer and more-personalized user experiences. Such AI systems—termed **experience managers** (EM)—structure interaction in order to achieve particular rich experiences per a designer's specification of what players should or should not encounter during all potential playthroughs of the virtual world [2].

A key part of managing a player's experience is ensuring that the virtual world remains *mechanics-wise consistent*: "[The] Game should react in a consistent, challenging, and exciting way to the player's actions" (3, p. 1511). Mechanics-wise consistency is arguably easy to establish and uphold. However, industry practice has developed (at least) one motivating case where it is important to shift previously-established mechanics because it serves a more-important experiential purpose. The EM within *The Elder Scrolls V: Skyrim* [4] shifts its previously-established combat mechanics when the player attempts to attack an *essential* non-player character (NPC), themselves needed to progress plot-critical events: if an essential NPC's health is depleted, they fall to one knee (Figure 1) as opposed to dying like non-essential NPCs. This mechanics-shift is potentially *visible*: players may notice that the outcomes of the same action are not equivalent in all game contexts for unexplained reasons. Visible dynamic mechanics adjustment (DMA) is considered problematic: scholars argue it potentially thwarts the player's sense of presence and agency [5, 6], and practitioners stress the importance of making DMA invisible [7, 8].

Figure 1. Essential NPCs in *The Elder Scrolls V: Skyrim* fall to one knee when their health is depleted, as opposed to dying. This is a perceivable shift in established mechanics, in service of some experience goal.

However, this "obvious" problem has neither been robustly demonstrated nor generally solved. Thus, we do not know to what extent visible DMA is a problem; if it is, we lack a method to generally solve it. This may result in wasted effort, ad-hoc solutions, or re-inventing the wheel. In this paper, we report two contributions to fill that gap.

We first report the results of a human-subjects study we conducted to assess the need for DMA to remain invisible. The study used a hand-crafted interactive narrative that can be generated by an automated planning-based experience manager [e.g. 9, 10] to test whether players notice inconsistencies brought about by a (non-dynamic) mechanics adjustment; no care is taken to make the adjustment invisible in the experiment. Our analysis demonstrates that players do notice these mechanics-shifts in real time, providing an empirical basis for our second contribution: a formal model of invisible dynamic game adjustment that depends on *knowledge tracking*, with two methods to track player knowledge—a conservative method relevant to a wide variety of virtual environments (VE), and a more nuanced one relevant to VEs intended to be experienced via head-mounted virtual reality (VR) displays.

## 2. Are Mechanics Adjustments Perceivable?

We first tested whether adjustments that contradict previously established mechanics are visible to the player. Players are said to notice unconstrained game adjustments, but this phenomenon has not been formally validated. In fact, past experiments have called into question a player's ability to notice inconsistencies in Dynamic Difficulty Adjustment (DDA, a more-specific kind of DMA) [8] and VR [11]. If players do not notice contradicting adjustments, then there is no need to restrict mechanic changes; all adjustments would be invisible and not impact gameplay.

### 2.1. Study Design

To test whether inconsistent adjustments are noticed in real-time by players, we build upon cognitive psychology work in reading comprehension. Specifically, we study the perceivability of changes to mechanics in a text-based Choose Your Own Adventure [12, CYOA]. We pursued a text-based interactive narrative experiment—as opposed to one within a virtual environment—for three key reasons.

First, cognitive psychologists who study reading comprehension have demonstrated that readers are sensitive to logical inconsistencies in the context of short stories [13, 14]. These inconsistencies impact reading comprehension by breaking coherence, which is measured through *reading time*. Previous work shows an increase in the average reading time for a sentence when it introduces a logical inconsistency compared to the same sentence when it is consistent, relative to the same surrounding story. In our evaluation, we use reading time as a test for whether mechanics adjustments that can be effected by an EM are invisible to the player.

Second, we wanted to control for potential confounds. If we were to use (for example) a VR head-mounted display (HMD), we would need to somehow ensure that all experiment participants experience the same stimuli in an environment where they have direct agency over their *field-of-view* (FoV), which is practically infeasible.

Third, we wanted to make it difficult for mechanic changes to be detected. A text-based CYOA has a fairly low degree of VR *fidelity*, the perception-independent degree of exactness with which real world experiences are reproduced [15]. Arguably, if DMA is detected with this fidelity, the effect will be greater with in higher-fidelity VEs.

Our study uses two types of short Choose Your Own Adventure stories: those with *consistent* and those with *inconsistent* action outcomes. Consistent outcomes model game worlds where game adjustment ensures game mechanics never contradict what the player has observed. Inconsistent action outcomes model worlds where one is free to shift back and forth between two sets of contradicting action outcomes. We expect participants to read inconsistent outcomes more slowly on average when compared with consistent outcomes. This delay accounts for time participants need to reconcile conflicting world models while reading inconsistent stories and indicates the game mechanic modification was visible to the player.

### 2.2. Setup

We recruited participants through Amazon's Mechanical Turk to play a series of ten stories. Participants were told they would read a series of 10 short Wild West stories, make a choice that affects the outcome, and then answer a question about each experience. The study was advertised as taking 10 to 20 minutes and participants were paid a reward on completion. No mechanism prevented participants from skipping through the stories to get the reward. Each session was timed and participants were asked a simple comprehension question after each story.

Participants were presented with a game via a short tutorial that instructed them to place their thumbs on the spacebar and index fingers on the 'f' and 'j' keys. The tutorial and stories were presented to the participants one line at a time. Each spacebar press erased the current line and presented the next. When presented with a two-option choice, participants pressed the 'f' key to select the first option or the 'j' key to select the second. Participants were randomly assigned to one of two sequences of consistent and inconsistent test stories. The first three stories were used for training and the following seven were test stories. Reading times for each sentence were timed in milliseconds and stored on a database along with comprehension responses.

### 2.3. Materials

Each CYOA has five parts. Here, we present an example for each, drawn from one CYOA used in the study. The first part introduces the main character.

Introduction *You sell snake oil liniment to the men and women of the pioneer. You travel near and far in your wagon selling what you pitch as a magical, cure-all elixir.*

An expectation is then created about the main character.

Differentiating Information *Of course, your elixirs don't actually cure anything. You make the concoction from mineral oil, red pepper, and turpentine. No matter what you claim, your mixtures can't heal anything. In fact, they usually make people sick.*

This creates an expectation about future action outcomes. In each story, there are two models of world mechanics: World A and B. In this World A, snake oil elixirs have magical properties that cure characters. In World B, snake oil elixirs are inert. We create the expectation that the participant exists in World B. Here, they expect snake oil to have no healing properties. Once a World B expectation has been created, the participant is given a situation.

Choice Frame *You arrive at a small town called Slate. You hear gunshots as you pull in to town. A young woman runs over to your wagon and implores you to bring elixir to the local tavern. As you enter the tavern with a bottle of elixir, you see the local sheriff and a young man laying on the floor. Each man has a fatal bullet wound in their stomach. The young woman asks you to save them with your elixir.*

At this point, the participant makes a choice.

Choice *Save the sheriff. Save the young man.*

Choices are designed so that they always show an outcome consistent or inconsistent with the World B expectation. A consistent outcome uses the expected mechanics from World B. In this case, that the snake oil heals neither the sheriff nor the young man.

Consistent 1 *[You help the sheriff drink the elixir but he dies from the fatal wound.] The young man also succumbs to his bullet wound. You tell the town people that the elixir did not have time to fully restore their wounds. You sell several crates of elixir to protect the people from possible bullet wounds.*

Consistent 2 *[You help the young man drink the elixir but he dies from the fatal wound.] The sheriff also succumbs to his bullet wound. You tell the town people that the elixir did not have time to fully restore their wounds. You sell several crates of elixir to protect the people from possible bullet wounds.*

An inconsistent outcome uses the unexpected mechanics from World A. In this case, for the snake oil to heal either the sheriff or young man.

Inconsistent 1 *[You help the sheriff drink the elixir and it soon heals the fatal wound.] You run out to your wagon and bring back another bottle for the young man. He is fully healed as well. You sell several crates of elixir to protect the people from any additional bullet wounds.*

Inconsistent 2 *[You help the young man drink the elixir and it soon heals the fatal wound.] You run out to your wagon and bring back another bottle for the sheriff. He is fully healed as well. You sell several crates of elixir to protect the people from any additional bullet wounds.*

No matter what the player decides, the sheriff and young man are healed. The first sentence of each outcome, shown surrounded with brackets, is the target sentence. Target sentence reading time is measured and analyzed. All target sentences are 18 syllables long and as structurally similar as possible to control for text-level factors [16]. Finally, every story is followed by a simple comprehension question.

Question *Did you visit a small town called Slate? Yes*

The comprehension questions are all yes/no, as easy as possible, and concern some major story event or detail.

**Hypothesis.** We hypothesize that consistent target sentences will be read faster than inconsistent sentences. The null hypothesis is thus: there is no difference in reading times between consistent/inconsistent target sentences.

### 2.4. Results

We recruited 115 participants. The participants had to have a HIT approval rating of at least 95% and at least 500 HITs approved. We used comprehension questions to screen for inattentive participants by only accepting those who answered at least 5 out of 10 questions correctly. Of the 115 participants, 86 answered at least 50% of the comprehension questions correctly (acceptance rate: 75%). With seven stories per participant, that leaves 602 reading time data points. Of those, 11 were rejected for being outside three standard deviations from the mean (acceptance rate: 98.2%). To test our hypothesis we performed a t-test between the two groups.

We found a significant difference between the consistent ($M$=2058, $SD$=1143) and inconsistent ($M$=2302, $SD$=1315) groups, $t(578) = 2.40$, $p = 0.017$ and it took participants on average 244ms longer to read the inconsistent target sentences than the consistent target sentences, illustrated via the box plot in Figure 2. Our results are consistent with prior work, which found differences from $\sim$200ms [13] to $\sim$300ms [14] for textually distant inconsistencies. We thus reject the null in favor of our alternate.

### 2.5. Discussion

Our results suggest that invisibility is broken when established world rules are violated. This is empirical evidence that common wisdom is in fact true: we have—for the first time—demonstrated that fully observable mechanic adjustments are actively noticed by players during gameplay. This evidences the need for restricting DMAs to those which the player has not observed in order to maintain invisibility.

The question then becomes: how might an EM perform invisible DMA in an automated manner? This requires formally modeling invisible DMA within the EM, which is what we describe in the subsequent sections.

## 3. Prior Work on DMA

Dynamically adjusting game mechanics relative to experience requirements is not a new idea. Thue and Bulitko [17, p. 44] proposed using an EM to perform what they termed *procedural game adaptation*: "a designer-controlled way to change a game's dynamics during end-user play." They formulate the problem centered on the player's *preferences*. In contrast, we formulate the problem centered on the player's *perception*. Whereas they propose to shift mechanics toward ones players *prefer* (based on a reinforcement learning-based model of player preferences), we propose to shift mechanics to ensure the broadest set of possible worlds consistent with player perception in a data-lean way.

AI systems that manipulate game elements in perception-consistent ways are also not new. Sunshine-Hill and Badler [18] developed a system for "alibi generation," which is responsible for incrementally giving procedurally generated NPCs information meant to give the impression that their behavior is goal-driven. This *level-of-detail* (LoD)

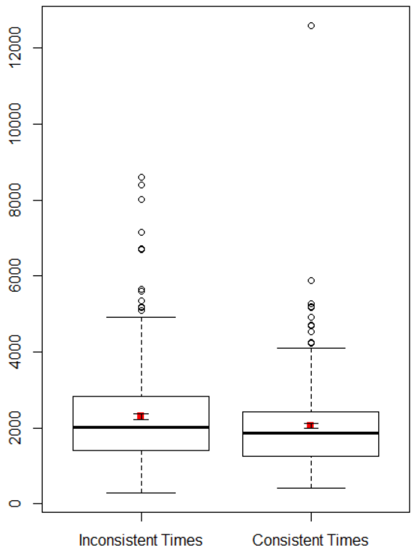



Figure 2. A box plot of the results. The y-axis is reading time in milliseconds and the red dots represent the mean of each group.

approach is driven by the alibi generator's underlying "perceptual simulation" that attempts to guarantee that the cheaper-to-maintain partially-rendered world is perceptually indistinguishable from its (hypothetical) actual simulation. Recently, Diamanti and Thue [19] broadened the LoD approach from characters to general world-state information; they developed a mechanism that abstracts more-detailed world states to logically-consistent but less-detailed world states (and vice-versa), relative to the player's perception of the world. In both cases, (1) the player's perception is mathematically modeled as an intrinsic part of the system and (2) the EM's resulting effect on the player is left unevaluated. In contrast, our approach affords extensibly specifying the conditions when the player's knowledge (in our case, of the game's mechanics) is updated on the basis of their perception (i.e. perception simulation is a module).

Our mechanic adjustments are similar to *interventions* used by the *strong-story* [2] (i.e. narrative-focused) EM, *Mimesis* [20]. Because interventions potentially break the player's suspension of disbelief [6], modern strong-story EMs have dropped support for this feature [21–24]. However, unlike intervention, our model of DMA ensures the world operates consistently from the player's perspective.

# 4. A Formal Model of Invisible DMA

In this section, we describe a formal model of invisible dynamic mechanics adjustment, implemented atop an existing experience management framework we obtained access to [25]. In sum, our system is an EM capable of enacting DMA designed to be invisible to the player.

In the subsequent sections, we describe the model of experience management we build upon, discuss how we expand it to support invisible dynamic mechanics adjustment (iDMA), present an example application of iDMA, and report the results from a pilot evaluation of the system's performance. We conclude the section by detailing how invisibility is supported in general and HMD-displayed VR.

## 4.1. Basis: Plan-based Experience Management

We adopt an automated planning-based experience management framework [9, 21, 23, 24] as the basis for defining iDMA. This framework can be described using the Planning Domain Definition Language [26, PDDL], a logical language used to formally specify state-transition systems and goal conditions. Each state is a set of (typically first-order) logic statements that are true.

A PDDL-based EM strives to solve an *experience management problem*, defined as a tuple $\langle i, \gamma, \omega \rangle$: $i$ models the *initial* state of the game world; $\gamma$ models the *experience goals*, a set of statements that must be true when the experience ends); and $\omega$ represents the game world's *mechanics*.

Formally, $\omega$ is a set of action operators that describe what characters—player-controlled and NPCs alike—can do in the game world. Each operator is a tuple $\langle n, p, e, b \rangle$ of an action name $n$, a set of preconditions $p$ that describe what must be true in the world's current state for the action to be performed, a set of effects $e$ that update said state, and a set of bindings $b$ that map operator variables to objects in the game world. An action is an *instance* of an operator.

## 4.2. Expanding Plan-based EM with iDMA

Invisible DMA requires two changes to the base PDDL EM setup: a set of operator sets $O$ and a microtheory [27] $Mt$ of player knowledge.

The first modification changes the EM problem from $\langle i, \gamma, \omega \rangle$ to $\langle i, \gamma, O \rangle$, where $O = \{\omega_0, \omega_1, \ldots, \omega_n\}$ is a set of individual action operator sets $\omega_i$ that iDMA can use to shift between game worlds at runtime.

The second modification is a set of axioms, called a microtheory $Mt$, used to extensibly determine what the player has observed. Microtheories are extensible because the set of axioms can change depending on the domain but the mechanics used to determine observations remain the same. A simple example axiom is $\mathbf{A1} = \forall x, y\colon \textit{Location}(y) \wedge \textit{At}(x, y) \wedge \textit{At}(P, y) \rightarrow \textit{Observes}(P, x)$, where $P$ is the player, and $x$ and $y$ are variables that can bind to objects within the game world (per PDDL). $\mathbf{A1}$ states that the player observes any object they are co-located with.

Our implementation of iDMA that builds upon our modifications to plan-based EM is called *domain revision*; it shifts a PDDL-based game world between alternate sets of game mechanics. If the player's actions come into conflict with the EM problem's goals $\gamma$ during gameplay, domain revision works by transitioning the player to an alternate world model within $O$ that preserves a path to a goal state and is *perceptually consistent* with the player's observations. An alternate history timeline $h'$ using domain $o'$ is perceptually consistent with an original timeline $h$ using domain $o$ if, according to $Mt$, all player observations of actions and state formulae are identical in $h$ and $h'$. Algorithm 1 gives pseudocode for

**Algorithm 1** The *IsConsistent* method returns true or false depending on whether two given world histories $h$ and $h'$ share consistent player observations according to a knowledge microtheory $Mt$.

```
ISCONSISTENT (World History Trajectory h, Alternate World History Trajectory h', Knowledge Microtheory Mt)
  for each sequential state-action pairs (s_i, a_i) and (s'_i, a'_i) in histories h and h', starting at i = 1
    for each literal l in Observes(s_i, Mt)
      if literal l is not in s'_i  return False
    if Observes(a_i, s_i, Mt)
      if a_i and a'_i do not share the same name, variables, or bindings  return False
  return True
```

**Algorithm 2** The *Observes* method returns the set of literals observed by the player in a given state according to a given knowledge microtheory.

```
OBSERVES (State s, Knowledge Microtheory Mt)
  L ← ∅
  for each literal l in s
    for each axiom x in Mt
      if x is satisfied by s with respect to l
        L ← L ∪ l and Break
  return L
```

**Algorithm 3** This *Observes* overload method returns true or false according to a given action, state, and microtheory.

```
OBSERVES (Action a, State s, Microtheory Mt)
  for each axiom x in Mt
    if x is satisfied by s with respect to a
      return True
  return False
```

comparing two world histories to determine whether they share consistent player observations. Algorithm 2 returns a set of literals the player observes in a given state according to $Mt$ and Algorithm 3 returns true or false depending on whether $Mt$ predicts an action is observed by the player.

### 4.3. Example

Consider the EM problem's PDDL in Figure 3; this EM must guide the evolution of the game toward predefined set of plot-related goal conditions, termed the *outcome* of the story. *Our* EM uses DGA to preserve invisibility while making mechanics changes to achieve the outcome.

In the example, the player is a mercenary who was injured escaping from thieves, now on the hunt in the woods outside their lair. The player is defenseless and one bandit has found their position. The EM's goal is to solidify trust between the player and a potential love interest who works with the thieves. The love interest has a horse and a magic potion. The EM plans for the love interest to attack the guard, give the potion to the player, and then ride with them to the nearest town on the horse. However, the player has a magic potion acquired earlier from a mysterious merchant. The mechanics allow the player to drink their own potion before the love interest has an opportunity to heal the player. Doing this will break the author goal.

One strategy is to intervene: prevent the potion from healing the player and proceed with the original story. After the player drinks the potion and nothing happens, their love interest defeats the bandit and heals the player with an identical magic potion. We expect the player will notice these irregularities and realize the system is manipulating their choices. An alternative to intervention is DGA through domain revision. Domain revision is given two alternate sets of world mechanics: one where magic potions are healing elixirs and another where potions have no medicinal properties. When the player drinks the potion and domain revision finds the player has made no observations that distinguish between the magic potion as medicine and snake oil, it shifts the player from the medicine to the snake oil universe.

After the mechanics shift, the love interest can no longer heal the player with a magic potion. The EM finds an alternate story: the love interest takes the player to a nearby doctor in town. This fulfills all author conditions for the love subplot without violating player observations of mechanics.

### 4.4. Analysis

Our plan-based EM fully expands two interactive narrative trees from the example PDDL; the first was generated with a baseline method and the second by the baseline plus domain revision. Both trees are turn-ordered, the player takes an action and then each NPC has a chance to act according to the current story plan.

Figure 4a illustrates a flattened version of the baseline tree that shows only possible player actions and Figure 4b illustrates the additional branch generated by iDMA. Full versions of both trees would show a linear series of actions for each NPC between user choices that neither add nor remove branches. States are represented by circles with unique numbers that identify expansion order. Player actions are represented by directed edges. Edges are labeled with action names. The player can take no action. These edges are omitted unless they lead to unique state sequences.

In the example, domain revision allows the player to take any action and also ensures the narrative goals are met. With the baseline, there is one branch where player actions and system control cannot be balanced. This is a 25% increase in branches where system objectives are met (from 4 to 5) in the example problem. Thus, our method affords more control over events compared to a non-intervention method.

**Initial State**
Player, Bandit, Love, Horse at Woods
Doctor at Town    Love has Potion2
Player is injured
Player has Potion1

**Goal State**
Player not injured
Love saved Player
Bandit is defeated
Player, Love at Town

(a) A simplified, informal description of the example problem. Words that begin with upper-case letters (e.g. Player) are game world objects

**move(?mvr,?to,?from)**
Precons: ?mvr at ?from
?mvr not injured
?from connected ?to
Effects: ?mvr not at ?from
?mvr at ?to

**dismount(?rdr,?hrs,?loc)**
Precons: ?rdr on ?hrs
?hrs at ?loc
Effects: ?rdr not on ?hrs
?rdr at ?loc

**attack(?atkr,?atkd,?loc)**
Precons: ?atkr at ?loc
?atkd at ?loc
?atkr not injured
Effects: ?atkd is defeated

**mount(?mntr,?hrs,?loc)**
Precons: ?mntr at ?loc
?hrs at ?loc
Effects: ?mntr not at ?loc
?mntr on ?hrs

**heal(?chr,?hld,?hlr,?loc)**
Precons: ?hlr at ?loc
?hlr is a doctor
?hld at ?loc
?chr at ?loc
?chr is not injured
?hld is injured
Effects: ?hld is not injured
?chr saved ?hld

(b) The shared example domain. Preconditions and effects are described in English-like syntax for readability. Tokens beginning with "?" (e.g. ?loc) are variables. These operators are shared by both alternate domains.

**drink(?drnkr,?ptn)**
Precons: ?drnkr has ?ptn
?drnkr is injured
Effects: ?drnkr not has ?ptn
?drnkr not injured

**give(?gvr,?drnkr,?ptn,?loc)**
Precons: ?gvr at ?loc
?drnkr at ?loc
?gvr has ?ptn
?drnkr is injured
Effects: ?gvr not has ?ptn
?drnkr not injured
?gvr saved ?drnkr

(c) The first alternate domain where magic potions heal characters.

**drink(?drnkr,?ptn)**
Precons: ?drnkr has ?ptn
?drnkr is injured
Effects: ?drnkr not has ?ptn

**give(?gvr,?drnkr,?ptn,?loc)**
Precons: ?gvr at ?loc
?drnkr at ?loc
?gvr has ?ptn
?drnkr is injured
Effects: ?gvr not has ?ptn

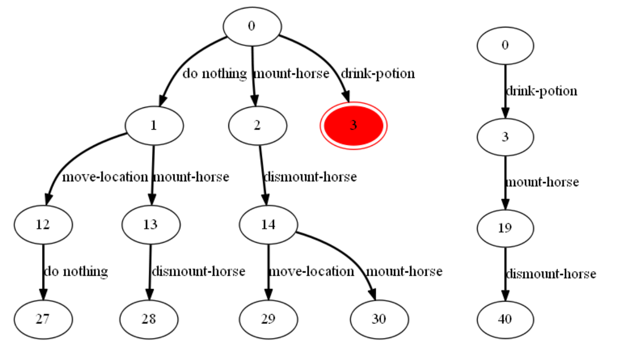
(d) The second alternate domain where magic potions are inert and drinking a magic potion does not heal injuries.

Figure 3. Simplified PDDL model of the example. Figure 3a presents the initial and goal states and 3b is a set of operators shared by both game adjustment alternatives. Figures 3c and 3d are two alternate sets of possible mechanics: one where potions are medicinal and one where they are useless.

## 4.5. Microtheories for Virtual Reality

Algorithms 2 and 3 ultimately determine the effectiveness of domain revision as a mechanism that implements iDMA. They indicate what aspects of the game world are observed by the player via their input microtheory $Mt$, and everything under observation is subject to scrutiny for consistency/inconsistency (§2).

However, these algorithms operate over the *syntactic* specification of the microtheory. Such a specification is a declarative abstraction of the underlying game world, which




(a) (b)

Figure 4. Figure 4a represents the player's choices in a baseline, turn-based interactive narrative tree where the player acts first and NPCs are controlled by the system. The tree edges are labeled with action names and a dead end is marked in red with a double border. Figure 4b is the extra branch created by mechanics adjustment in the dead end's position of the original tree.

must be grounded into the procedural simulation in which players interact [28]; said simulation is what gives a syntactic specification its *semantics*. This *declarative-procedural gap*—when there is no specification of what AI constructs mean in terms of the virtual world they model—must be resolved in order to deploy an EM in VR.

**A Conservative Microtheory.** A modest proposal for a microtheory that solves part of the declarative-procedural gap involves defining a set of distinguished predicates that cut across a variety of game worlds. One candidate set of such abstractions are those involving concepts central to narratively-oriented games, whose worlds typically represent characters, locations, and items [28, 29]: *Character*$(x)$, which denotes that $x$ is a character; *Location*$(x)$, which denotes that $x$ is a location; and *Item*$(x)$, which denotes that $x$ is an item. This would establish a declarative representation that potentially affords defining knowledge-tracking axioms across a wide variety of games. However, it is still conservative because it leaves unspecified what the procedural counterpart of these axioms is.

For instance, although **A1** indicates the player observes any object they are co-located with, we have not yet specified what the declarative formulae *At*$(x, y)$ and *Location*$(y)$ mean in terms of the game world. Naively, we might ground the *Location* predicate as a *bounding box* in the virtual world, and the *At* predicate as a function that checks whether $x$ is contained within said bounding box.

**An HMD VR-centric Microtheory.** In virtual reality, we may be empowered to give an accurate semantic meaning to the *Observes* predicate.

Prior work has demonstrated [30] that people attend to *personal space*—within reach, typically 2 meters—differently from *action space*—outside personal space, typically within 30 meters. Further, objects within the HMD's FoV are more likely to be observed (Figure 5).

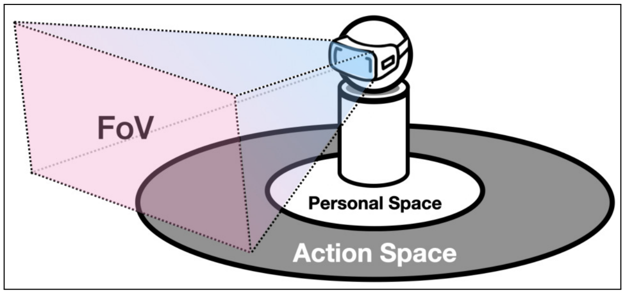



Figure 5. People attend to information in *personal space* differently from that in *action space* (30m), and will likely perceive objects within the HMD's FoV. These attributes can combine to give semantic meaning to the *Observes* predicate for a HMD-specific microtheory.

Thus, we might assess the truth value of *Observes*$(P, x)$ based on a calculation involving the HMD's position, orientation, FoV, and raycast-based occlusion to $x$.

## 5. Conclusion

We defined the problem of dynamic mechanics adjustment (DMA), a generalization of dynamic difficulty adjustment DDA. Like DDA, DMA shares the goal of the adjustment being "inscrutable" (8, p. 429); unlike it, DMA admits changing mechanics *other* than those related to the game's challenge level. We presented an initial technical framework for a PDDL-based experience manager that tracks player knowledge to perform *invisible* adjustments, unnoticed by the player. This was motivated by an empirical evaluation that indicates players notice mechanic adjustments that contradict their knowledge in simple CYOA stories; our experiment evidences the need for our framework.

Our work here is a baseline and (fortunately) there are many open questions around player cognition, dynamic mechanic adjustment, and gameplay experience to explore in future work. We present 4 broader implications to explore as a consequence of our efforts: the potential impact of *internal* v. *external* game knowledge on DMA, the possibilities that *change blindness* afford for DMA, the potential effects of *visible* DGA on player experiences, and the role *consistency* could play during DMA.

Our evaluation only uses pre-existing real world rules for game mechanics. For example, it is common knowledge that snake oil elixirs are not medicine. This leaves an open question of whether participants can identify inconsistencies in rules learned solely through communicated gameplay [31], divorced from external real-world knowledge. In an inversion of our snake oil example, would *Legend of Zelda* [32] players notice if drinking a health potion resulted in no status change instead of restoring their health? The rule that potions restore health in *The Legend of Zelda* runs contrary to real world knowledge and is learned through experience with the fantasy genre and the *Zelda* game series.

Another open question is whether all observable mechanic changes will be noticed by the player. Our experiment provides evidence that players reliably notice mechanic changes in short CYOAs. However, *change blindness*—when a person fails to notice a change in visual stimulus [33]—-has been observed in games with dynamic difficulty adjustment [8] and virtual reality environments [11]. Change blindness leads to logical inconsistencies: changes are presented to but not attended by the player. As a result, changes are not incorporated into the player's mental model of the game world. If accurately predicted, unnoticed inconsistencies could allow DMA to make more modifications to control gameplay experience without affecting invisibility; our VR-centric microtheory could potentially afford this.

A third open question is how mechanic inconsistencies affect gameplay experience. Not all noticed inconsistencies are detrimental to player experience. *Antichamber* [34] intentionally breaks a player's Euclidian expectations of physical space to establish its own set of non-Euclidian principles of how space is navigated. While noticeable, this departure from real-world expectations is the game's central conceit and what makes it interesting.

Finally, some games subvert expectations, continuously and inconsistently. *I Wanna Be the Guy: The Movie: The Game (IWBTG)* [35] is a hyper-difficult game that is intentionally and visibly unfair, in part due to visibly shifting between several different mechanic models. One repeated inconsistent mechanic are apples that fall downward and upwards out of trees to kill the player. This mechanic both contradicts external knowledge of gravity and is applied inconsistently to different apples throughout the game.

Our end goal is an experience manager aware of user psychology, able to generate and shift between game world rules, and dynamically generate virtual reality worlds in order to provide targeted experiences to the player. This integrated experience management agent capable of using dynamic mechanics adjustment will function as an online game master, dynamically revising world mechanics and assembling the virtual environment based on the gameplay experience it wants to provide for the user, and its models of what the user has done and observed.

## References

[1] D. L. Roberts, "Seven design challenges for fully-realized experience management," in *Proc. of the 4th Workshop on Intelligent Narrative Technologies at the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2011.

[2] M. Riedl and V. Bulitko, "Interactive Narrative: An Intelligent Systems Approach," *AI Magazine*, vol. 34, no. 1, pp. 67–77, 2013.

[3] H. Desurvire, M. Caplan, and J. A. Toth, "Using heuristics to evaluate the playability of games," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems–Extended Abstracts*, 2004, pp. 1509–1512.

[4] Bethesda Game Studios, "*The Elder Scrolls V: Skyrim*," 2011.

[5] M. Riedl, C. J. Saretto, and R. M. Young, "Managing Interaction Between Users and Agents in a

Multi-Agent Storytelling Environment," in *International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2003, pp. 741–748.

[6] D. L. Roberts and C. L. Isbell, "A Survey and Qualitative Analysis of Recent Advances in Drama Management," *International Transactions on Systems Science and Applications*, vol. 4, no. 2, pp. 61–75, 2008.

[7] A. Rollings and E. Adams, *Andrew Rollings and Ernest Adams on Game Design*. New Riders, 2003.

[8] R. Hunicke, "The Case for Dynamic Difficulty Adjustment in Games," in *Proc. of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACM, 2005, pp. 429–433.

[9] A. Zook and M. O. Riedl, "Generating and adapting game mechanics," in *Proc. of the 5th Workshop on Procedural Content Generation in Games at the 9th International Conference on the Foundations of Digital Games*, 2014.

[10] J. Robertson and R. M. Young, "Perceptual experience management," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 15–24, 2018.

[11] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Wartell, "Leveraging Change Blindness for Redirection in Virtual Environments," in *Proc. of the IEEE Virtual Reality Conference*, 2011, pp. 159–166.

[12] E. Packard, *The Cave of Time*, ser. Choose Your Own Adventure. Bantam Books, 1979.

[13] E. J. O'Brien and J. E. Albrecht, "Comprehension Strategies in the Development of a Mental Model," *J. of Exp. Psychology: Learning, Memory, and Cognition*, vol. 18, no. 4, pp. 777–784, 1992.

[14] J. E. Albrecht and E. J. O'Brien, "Updating a Mental Model: Maintaining Both Local and Global Coherence," *J. of Exp. Psychology: Learning, Memory, and Cognition*, vol. 19, no. 5, pp. 1061–1070, 1993.

[15] S. J. Gerathewohl, "Fidelity of simulation and transfer of training: A review of the problem," US Department of Transportation, Federal Aviation Administration, Office of Aviation Medicine, Technical Reports Center, K. F. Wendt Engr. Library, University of Wisconsin—Madison, Tech. Rep. AM 69–24, 12 1969.

[16] M. W. McNerney, K. A. Goodwin, and G. A. Radvansky, "A Novel Study: A Situation Model Analysis of Reading Times," *Discourse Processes*, vol. 48, no. 7, pp. 453–474, 2011.

[17] D. Thue and V. Bulitko, "Procedural game adaptation: Framing experience management as changing an mdp," in *Proc. of the Workshop on Intelligent Narrative Technologies at the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2012, pp. 44–50.

[18] B. Sunshine-Hill and N. I. Badler, "Perceptually Realistic Behavior through Alibi Generation," in *Proc. of the 6th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2010, pp. 83–88.

[19] M. Diamanti and D. Thue, "Automatic abstraction and refinement for simulations with adaptive level of detail," in *Proc. of the 15th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2019, pp. 17–23.

[20] R. M. Young, M. O. Riedl, M. Branly, A. Jhala, R. J. Martin, and C. J. Saretto, "An Architecture for Integrating Plan-Based Behavior Generation with Interactive Game Environments," *J. of Game Development*, vol. 1, no. 1, pp. 51–70, 2004.

[21] M. O. Riedl, A. Stern, D. M. Dini, and J. M. Alderman, "Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training," *International Transactions on Systems Science and Applications*, vol. 4, no. 2, pp. 23–42, 2008.

[22] J. Porteous, M. Cavazza, and F. Charles, "Applying Planning to Interactive Storytelling: Narrative Control Using State Constraints," *Transactions on Intelligent Systems and Technology*, vol. 1, no. 2, p. 10, 2010.

[23] A. Ramirez and V. Bulitko, "Automated Planning and Player Modelling for Interactive Storytelling," *Transactions on Computational Intelligence and AI in Games*, pp. 375–386, 2014.

[24] J. Robertson and R. M. Young, "Gameplay as On-Line Mediation Search," in *Experimental AI in Games Workshop*. AAAI, 2014.

[25] ——, "Finding Schrödinger's Gun," in *Proc. of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2014, pp. 153–159.

[26] D. McDermott et al., "PDDL—The Planning Domain Definition Language," Yale Center for Computational Vision and Control, Tech. Rep. CVC TR98003/DCSTR1165, 1998.

[27] R. V. Guha and D. B. Lenat, "Cyc: A Midterm Report," *Readings in Knowledge Acquisition and Learning*, pp. 839–866, 1993.

[28] R. E. Cardona-Rivera, "Foundations of a computational science of game design: Abstractions and tradeoffs," in *Proc. of the 16th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020.

[29] G. Prince, *A Dictionary of Narratology: Revised Edition*. University of Nebraska Press, 2003.

[30] J. E. Cutting and P. M. Vishton, "Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth," in *Perception of space and motion*. Elsevier, 1995, pp. 69–117.

[31] R. E. Cardona-Rivera and R. M. Young, "Games as Conversation," in *Proc. of the 3rd Games and NLP Workshop at the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2014, pp. 2–8.

[32] Nintendo, "The Legend of Zelda," 1986.

[33] D. J. Simons and D. T. Levin, "Change blindness," *Trends in cognitive sciences*, vol. 1, no. 7, pp. 261–267, 1997.

[34] A. D. Bruce, "*Antichamber*," 2013.

[35] M. K. O'Reilly, "*I Wanna Be the Guy: The Movie: The Game (IWBTG)*," 2007.