MDA: A Formal Approach to Game Design and Game Research

Robin Hunicke, Marc LeBlanc, Robert Zubek

hunicke@cs.northwestern.edu, marc_leblanc@alum.mit.edu, rob@cs.northwestern.edu

Abstract

In this paper we present the MDA framework (standing for Mechanics, Dynamics, and Aesthetics), developed and taught as part of the Game Design and Tuning Workshop at the Game Developers Conference, San Jose 2001-2004.

MDA is a formal approach to understanding games – one which attempts to bridge the gap between game design and development, game criticism, and technical game research. We believe this methodology will clarify and strengthen the iterative processes of developers, scholars and researchers alike, making it easier for all parties to decompose, study and design a broad class of game designs and game artifacts.

Introduction

All artifacts are created within some design methodology. Whether building a physical prototype, architecting a software interface, constructing an argument or implementing a series of controlled experiments – design methodologies guide the creative thought process and help ensure quality work.

Specifically, iterative, qualitative and quantitative analyses support the designer in two important ways. They help her analyze the *end result* to refine implementation, and analyze the *implementation* to refine the result. By approaching the task from both perspectives, she can consider a wide range of possibilities and interdependencies.

This is especially important when working with computer and video games, where the interaction between coded subsystems creates complex, dynamic (and often unpredictable) behavior. Designers and researchers must consider interdependencies carefully before implementing changes, and scholars must recognize them before drawing conclusions about the nature of the experience generated.

In this paper we present the MDA framework (standing for Mechanics, Dynamics, and Aesthetics), developed and taught as part of the Game Design and Tuning Workshop at the Game Developers Conference, San Jose 2001-2004 [LeBlanc, 2004a]. MDA is a formal approach to understanding games – one which attempts to bridge the gap between game design and development, game criticism, and technical game research. We believe this methodology will clarify and strengthen the iterative processes of developers, scholars and researchers alike, making it easier for all parties to decompose, study and design a broad class of game designs and game artifacts.

Towards a Comprehensive Framework

Game design and authorship happen at many levels, and the fields of games research and development involve people from diverse creative and scholarly backgrounds. While it's often necessary to focus on one area, everyone, regardless of discipline, will at some point need to consider issues outside that area: base mechanisms of game systems, the overarching design goals, or the desired experiential results of gameplay.

AI coders and researchers are no exception. Seemingly inconsequential decisions about data, representation, algorithms, tools, vocabulary and methodology will trickle upward, shaping the final gameplay. Similarly, all desired user experience must bottom out, somewhere, in code. As games continue to generate increasingly complex agent, object and system behavior, AI and game design merge.

Systematic coherence comes when conflicting constraints are satisfied, and each of the game's parts can relate to each other as a whole. Decomposing, understanding and creating this coherence requires travel between all levels of abstraction – fluent motion from systems and code, to content and play experience, and back.

We propose the MDA framework as a tool to help designers, researchers and scholars perform this translation.

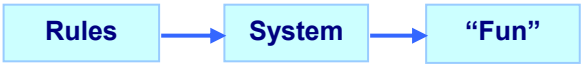
MDA

Games are created by designers/teams of developers, and consumed by players. They are purchased, used and eventually cast away like most other consumable goods.

The production and consumption of game artifacts.

The difference between games and other entertainment products (such as books, music, movies and plays) is that their consumption is relatively *unpredictable*. The string of events that occur during gameplay and the outcome of those events are unknown at the time the product is finished.

The MDA framework formalizes the consumption of games by breaking them into their distinct components:



…and establishing their design counterparts:

Mechanics describes the particular components of the game, at the level of data representation and algorithms.

Dynamics describes the run-time behavior of the mechanics acting on player inputs and each others' outputs over time.

Aesthetics describes the desirable emotional responses evoked in the player, when she interacts with the game system.

Fundamental to this framework is the idea that games are *more like artifacts* than media. By this we mean that the content of a game is its *behavior* – not the media that streams out of it towards the player.

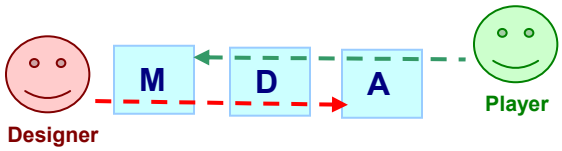
Thinking about games as designed artifacts helps frame them as systems that build behavior via interaction. It supports clearer design choices and analysis at all levels of study and development.

MDA in Detail

MDA as Lens

Each component of the MDA framework can be thought of as a "lens" or a "view" of the game – separate, but causally linked. [LeBlanc, 2004b].

From the designer's perspective, the mechanics give rise to dynamic system behavior, which in turn leads to particular aesthetic experiences. From the player's perspective, aesthetics set the tone, which is born out in observable dynamics and eventually, operable mechanics.



The designer and player each have a different perspective.

When working with games, it is helpful to consider both the designer and player perspectives. It helps us observe how even small changes in one layer can cascade into others. In addition, thinking about the player encourages experience-driven (as opposed to feature-driven) design.

As such, we begin our investigation with a discussion of Aesthetics, and continue on to Dynamics, finishing with the underlying Mechanics.

Aesthetics

What makes a game "fun"? How do we know a specific type of fun when we see it? Talking about games and play is hard because the vocabulary we use is relatively limited.

In describing the aesthetics of a game, we want to move away from words like "fun" and "gameplay" towards a more directed vocabulary. This includes but is not limited to the taxonomy listed here:

1. **Sensation**
   Game as sense-pleasure
2. **Fantasy**
   Game as make-believe
3. **Narrative**
   Game as drama
4. **Challenge**
   Game as obstacle course
5. **Fellowship**
   Game as social framework
6. **Discovery**
   Game as uncharted territory
7. **Expression**
   Game as self-discovery
8. **Submission**
   Game as pastime

For example, consider the games Charades, Quake, The Sims and Final Fantasy. While each are "fun" in their own right, it is much more informative to consider the aesthetic components that create their respective player experiences:

Charades: Fellowship, Expression, Challenge.

Quake: Challenge, Sensation, Competition, Fantasy.

The Sims: Discovery, Fantasy, Expression, Narrative.

Final Fantasy: Fantasy, Narrative, Expression, Discovery, Challenge, Submission.

Here we see that each game pursues multiple aesthetic goals, in varying degrees. Charades emphasizes Fellowship over Challenge; Quake provides Challenge as a main element of gameplay. And while there is no Grand Unified Theory of games or formula that details the combination and proportion of elements that will result in "fun", this

taxonomy helps us describe games, shedding light on how and why different games appeal to different players, or to the same players at different times.

Aesthetic Models

Using out aesthetic vocabulary like a compass, we can define models for gameplay. These models help us describe gameplay dynamics and mechanics.

For example: Charades and Quake are both competitive. They succeed when the various teams or players in these games are *emotionally invested* in defeating each other. This requires that players have adversaries (in Charades, teams compete, in Quake, the player competes against computer opponents) and that all parties want to win.

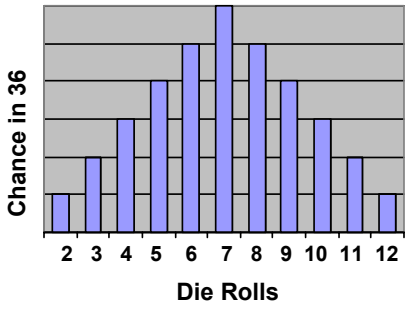
It's easy to see that supporting adversarial play and clear feedback about who is winning are essential to competitive games. If the player doesn't see a clear winning condition, or feels like they can't possibly win, the game is suddenly a lot less interesting.

Dynamic Models

Dynamics work to create aesthetic experiences. For example, *challenge* is created by things like time pressure and opponent play. *Fellowship* can be encouraged by sharing information across certain members of a session (a team) or supplying winning conditions that are more difficult to achieve alone (such as capturing an enemy base).

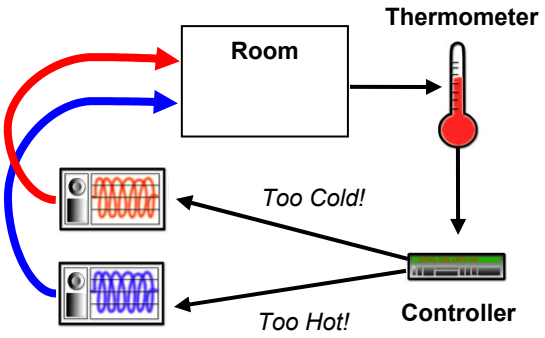
Expression comes from dynamics that encourage individual users to leave their mark: systems for purchasing, building or earning game items, for designing, constructing and changing levels or worlds, and for creating personalized, unique characters. *Dramatic tension* comes from dynamics that encourage a rising tension, a release, and a denouement.

As with aesthetics, we want our discussion of dynamics to remain as concrete as possible. By developing models that predict and describe gameplay dynamics, we can avoid some common design pitfalls.



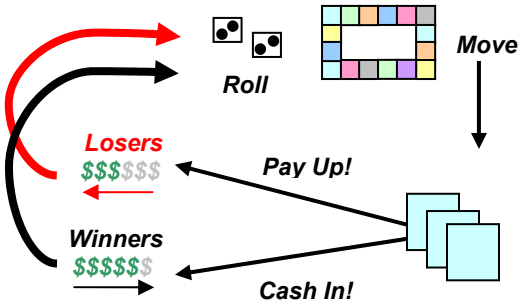
Probabilistic distribution of the random variable 2 D6.

For example, the model of 2 six-sided die will help us determine the average time it will take a player to progress around the board in Monopoly, given the probability of various rolls.



A thermostat, which acts as a feedback system.

Similarly, we can identify feedback systems within gameplay to determine how particular states or changes affect the overall state of gameplay. In Monopoly, as the leader or leaders become increasingly wealthy, they can penalize players with increasing effectiveness. Poorer players become increasingly poor.



The feedback system in Monopoly.

As the gap widens, only a few (and sometimes only one) of the players is really invested. Dramatic tension and agency are lost.

Using our understanding of aesthetics and dynamics, we can imagine ways to fix Monopoly – either rewarding players who are behind to keep them within a reasonable distance of the leaders, or making progress more difficult for rich players. Of course – this might impact the game's ability to recreate the reality of monopoly practices – but reality isn't always "fun".

Mechanics

Mechanics are the various actions, behaviors and control mechanisms afforded to the player within a game context. Together with the game's content (levels, assets and so on) the mechanics support overall gameplay dynamics.

For example, the mechanics of card games include shuffling, trick-taking and betting – from which dynamics like bluffing can emerge. The mechanics of shooters include weapons, ammunition and spawn points – which sometimes produce things like camping and sniping. The mechanics of golf include balls, clubs, sand traps and water hazards – which sometimes produce broken or drowned clubs.

Adjusting the mechanics of a game helps us fine-tune the game's overall dynamics. Consider our Monopoly example. Mechanics that would help lagging players could include bonuses or "subsidies" for poor players, and penalties or "taxes" for rich players – perhaps calculated when crossing the Go square, leaving jail, or exercising monopolies over a certain threshold in value. By applying such changes to the fundamental rules of play, we might be able to keep lagging players competitive and interested for longer periods of time.

Another solution to the lack of tension over long games of Monopoly would be to add mechanics that encourage time pressure and speed up the game. Perhaps by depleting resources over time with a constant rate tax (so people spend quickly), doubling all payouts on monopolies (so that players are quickly differentiated), or randomly distributing all properties under a certain value threshold.

Tuning

Clearly, the last step our Monopoly analysis involves play testing and tuning. By iteratively refining the value of penalties, rate of taxation or thresholds for rewards and punishments, we can refine the Monopoly gameplay until it is balanced.

When tuning, our aesthetic vocabulary and models help us articulate design goals, discuss game flaws, and measure our progress as we tune. If our Monopoly taxes require complex calculations, we may be defeating the player's sense of investment by making it harder for them to track cash values, and therefore, overall progress or competitive standings.

Similarly, our dynamic models help us pinpoint where problems may be coming from. Using the D6 model, we can evaluate proposed changes to the board size or layout, determining how alterations will extend or shorten the length of a game.

MDA at Work

Now, let us consider developing or improving the AI component of a game. It is often tempting to idealize AI components as black-box mechanisms that, in theory, can be injected into a variety of different projects with relative ease. But as the framework suggests, game components cannot be evaluated in vacuo, aside from their effects on a system behavior and player experience.

First Pass

Consider an example Babysitting game [Hunicke, 2004]. Your supervisor has decided that it would be beneficial to prototype a simple game-based AI for tag. Your player will be a babysitter, who must find and put a single baby to sleep. The demo will be designed to show off simple emotive characters (like a baby), for games targeted at 3-7 year-old children.

What are the aesthetic goals for this design? Exploration and discovery are probably more important than challenge. As such the dynamics are optimized here not for "winning" or "competition" but for having the baby express emotions like surprise, fear, and anticipation.

Hiding places could be tagged manually, paths between them hard-coded; the majority of game logic would be devoted to maneuvering the baby into view and creating baby-like reactions. Gameplay mechanics would include talking to the baby ("I see you!" or "boo!"), chasing the baby (with an avatar or with a mouse), sneaking about, tagging and so on.

Second Pass

Now, consider a variant of this same design – built to work with a franchise like Nickelodeon's "Rugrats" and aimed at 7-12 year-old-girls. Aesthetically, the game should feel more challenging – perhaps there is some sort of narrative involved (requiring several "levels", each of which presents a new piece of the story and related tasks).

In terms of dynamics, the player can now track and interact with several characters at once. We can add time pressure mechanics (i.e. get them all to bed before 9 pm), include a "mess factor" or monitor character emotions (dirty diapers cause crying, crying loses you points) and so on.

For this design, static paths will no longer suffice – and it's probably a good idea to have them choose their own hiding places. Will each baby have individual characteristics, abilities or challenges? If so, how will they expose these differences to the player? How will they track internal state, reason about the world, other babies, and the player? What kinds of tasks and actions will the player be asked to perform?

Third Pass

Finally, we can conceive of this same tag game as a full-blown, strategic military simulation – the likes of Splinter Cell or Thief. Our target audience is now 14-35 year old men.

Aesthetic goals now expand to include a fantasy element (role-playing the spy-hunting military elite or a loot-

seeking rogue) and challenge can probably border on submission. In addition to an involved plot full of intrigue and suspense, the player will expect coordinated activity on the part of opponents – but probably a lot less emotional expression. If anything, agents should express fear and loathing at the very hint of his presence.

Dynamics might include the ability to earn or purchase powerful weapons and spy equipment, and to develop tactics and techniques for stealthy movement, deceptive behavior, evasion and escape. Mechanics include expansive tech and skill trees, a variety of enemy unit types, and levels or areas with variable ranges of mobility, visibility and field of view and so on.

Agents in this space, in addition to coordinating movement and attacks must operate over a wide range of sensory data. Reasoning about the player's position and intent should indicate challenge, but promote their overall success. Will enemies be able to pass over obstacles and navigate challenging terrain, or will you "cheat"? Will sound propagation be "realistic" or will simple metrics based on distance suffice?

Wrapping Up

Here we see that simple changes in the aesthetic requirements of a game will introduce mechanical changes for its AI on many levels – sometimes requiring the development of entirely new systems for navigation, reasoning, and strategic problem solving.

Conversely, we see that there are no "AI mechanics" as such – intelligence or coherence comes from the interaction of AI logic with gameplay logic. Using the MDA framework, we can reason explicitly about aesthetic goals, draw out dynamics that support those goals, and then scope the range of our mechanics accordingly.

Conclusions

MDA supports a formal, iterative approach to design and tuning. It allows us to reason explicitly about particular design goals, and to anticipate how changes will impact each aspect of the framework and the resulting designs/implementations.

By moving between MDA's three levels of abstraction, we can conceptualize the dynamic behavior of game systems. Understanding games as dynamic systems helps us develop techniques for iterative design and improvement – allowing us to control for undesired outcomes, and tune for desired behavior.

In addition, by understanding how formal decisions about gameplay impact the end user experience, we are able to better decompose that experience, and use it to fuel new designs, research and criticism respectively.

References

Barwood, H. & Falstein, N. 2002. "More of the 400: Discovering Design Rules". Lecture at *Game Developers Conference*, 2002. Available online at: http://www.gdconf.com/archives/2002/hal_barwood.ppt

Church, D. 1999. "Formal Abstract Design Tools." *Game Developer*, August 1999. San Francisco, CA: CMP Media. Available online at: http://www.gamasutra.com/features/19990716/design_tools_01.htm

Hunicke, R. 2004. "AI Babysitter Elective". Lecture at *Game Developers Conference Game Tuning Workshop*, 2004. In LeBlanc et al., 2004a. Available online at: http://algorithmancy.8kindsoffun.com/GDC2004/AITutorial5.ppt

LeBlanc, M., ed. 2004a. "Game Design and Tuning Workshop Materials", *Game Developers Conference 2004*. Available online at: http://algorithmancy.8kindsoffun.com/GDC2004/

LeBlanc, M. 2004b. "Mechanics, Dynamics, Aesthetics: A Formal Approach to Game Design." Lecture at Northwestern University, April 2004. Available online at: http://algorithmancy.8kindsoffun.com/MDAnwu.ppt